บรรณานุกรม

สุพงศ์ เภกะนันทน์. "การพัฒนาวงจรเข้ารหัสดิจิตอลจากสัญญาณเสียง."
วิทยานิพนธ์ปริญญามหาบัณฑิต ภาควิชาวิศวกรรมคอมพิวเตอร์ บัณฑิตวิทยาลัย
จุฬาลงกรณ์มหาวิทยาลัย, 2526.

Melmeistein Paul. "Automatic Segmentation of Speech into Syllable Units." The Acoustical Society of America. October 1975:880-882

อนันต์ อ่อมศาสตร์, เนาวรัตน์ อ่อมศาสตร์. ลักษณะภาษาไทย.
กรุงเทพฯ:สำนักพิมพ์ ไทยวัฒนาพานิช, 2519.

Dixon N.Rex and Martin B.Thomas. Automatic Speech & Speak Recognition. IEEE Press.

Cater P.John. Electronically Speaking:Computer Speech Generation Howard W.Sams&Co., Inc., 1983.

Luksaneeyanawin Sudaporn. "Intonation in Thai." Thesis Ph.D. University of Edinburgh, 1983.

ประยุทร กุยสาคร. ภาษาไทยเชิงภาษาศาสตร์. โรงพิมพ์การศาสนา, 2527.

Ongrabieb Preoupayom. "ETUDE COMPARATIVE DES SYSTEMES PHONOLOGIQUE ET PHONETIQUE DU FRANCAIS ET DU THAI EN VUE DE L ENSEIGNEMENT DU FRANCAIS AU THAILANDAIS." ETUDE DU CONSONNANTISME THESE 3 CY PARIS 3., 1975.

Abramson A.S. The Vowels and Tones of Standard Thai:Acoustical Measurements and Experiment. Bloomington:Indiana University Research Center in Anthropology,Folklore and Linguistics., 1962

สุคาพร ลักษณียะนาวิน. เอกสารประกอบคำบรรยายวิชา สัทศาสตร์ เรื่อง พยางค์ คณะอักษรศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย, 2529.

สุคาพร ลักษณียะนาวิน. เอกสารประกอบคำบรรยายวิชา สัทศาสตร์ เรื่อง สัทวิทยา คณะอักษรศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย, 2529.

Veltri J.Steven. How to Make Your Computer Talk. Mc.Graw-Hill Book Company., 1985.

Sakoe H., Chiba S. "Dynamic Programming Algorithm Optimization for Spoken Word Recognition." IEEE.,ASSP. Vol.26 No.1.Feb., 1978.

วีระ ริ้วพิทักษ์. "การรับรู้เสียงพูดของเครื่อง (Speech Recognition)." วารสารพีซีแม็กกาซีน ปีที่ 1 ฉบับที่ 4., 2529.

Gauvain J.L. "TECHNIQUES DE RECONNAISSANCE GLOBALE" LMISI 1985.

Markel J.D.,Gray,Jr. Linear Prediction of Speech Berlin Heidelberg,New York:Springer-verlag,1976.

Rabiner L.R.,Schafer W.R. Digital Processing of Speech Signals Englewood Cliff,New Jersey:Prentichall,1978.

Riewpituk Weera."RECONNAISSANCE DE LA PAROLE PAR MOTS ISOLES, APPLIQUEE A LA LANGUE THAIE,PRENANT EN COMPTE LES TONS LEXICAUX." THESE DE DOCTEUR INGENIER,UNIVERSITE DE PARIS-SUD,CENTRE D'ORSAY., 1985.

Gill P.S. "Voice Technology:An Integrated Approach.",Hayward, California.IEEE,1982.

พจนานุกรมฉบับราชบัณฑิตยสถาน พ.ศ.2493 ฉบับพิมพ์ครั้งที่ 16., 2522.

Weinstein J.,McCandless S.,Mondshein F.,Zue W. "A System for Acoustic-Phonetic Analysis of Continuous Speech." IEEE,ASSP Feb., 1975.

Abercrombie D. Elements of General Phonetics. Edinburgh University Press., 1967.

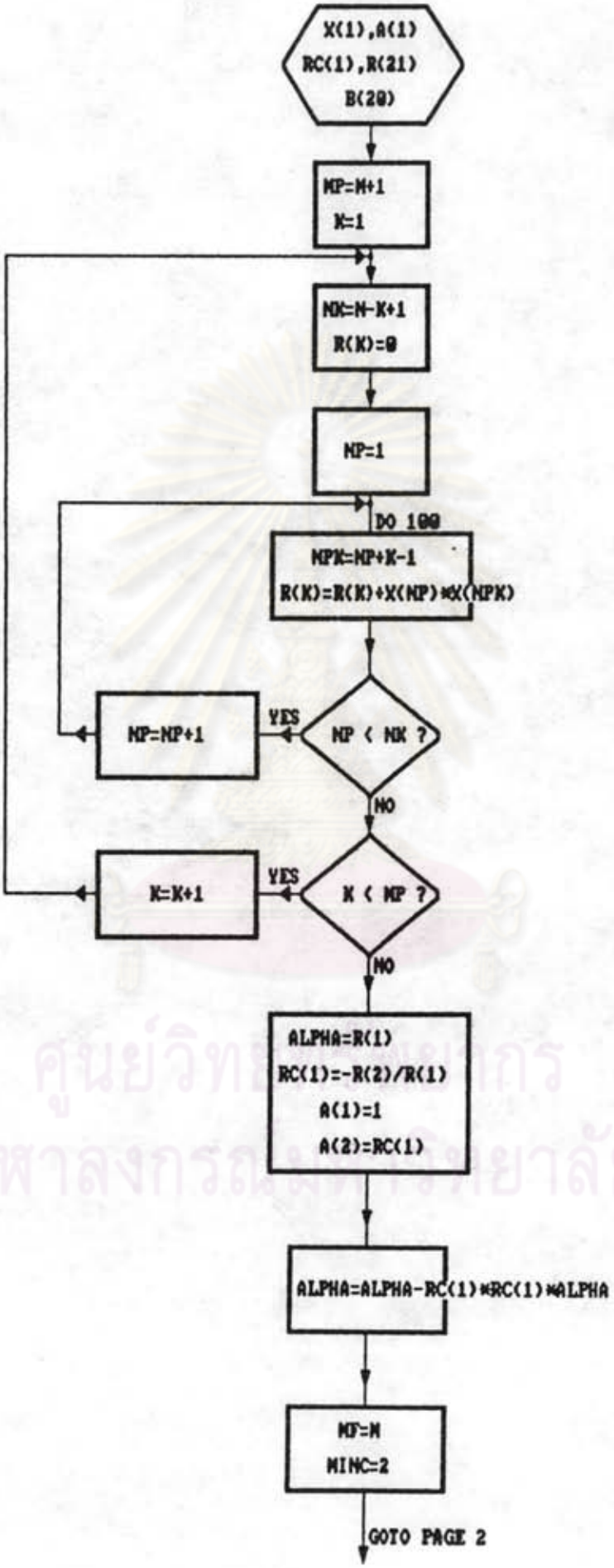
Makhoul J. "Linear Prediction:A Tutorial Review." IEEE. Vol.63 April,1975.

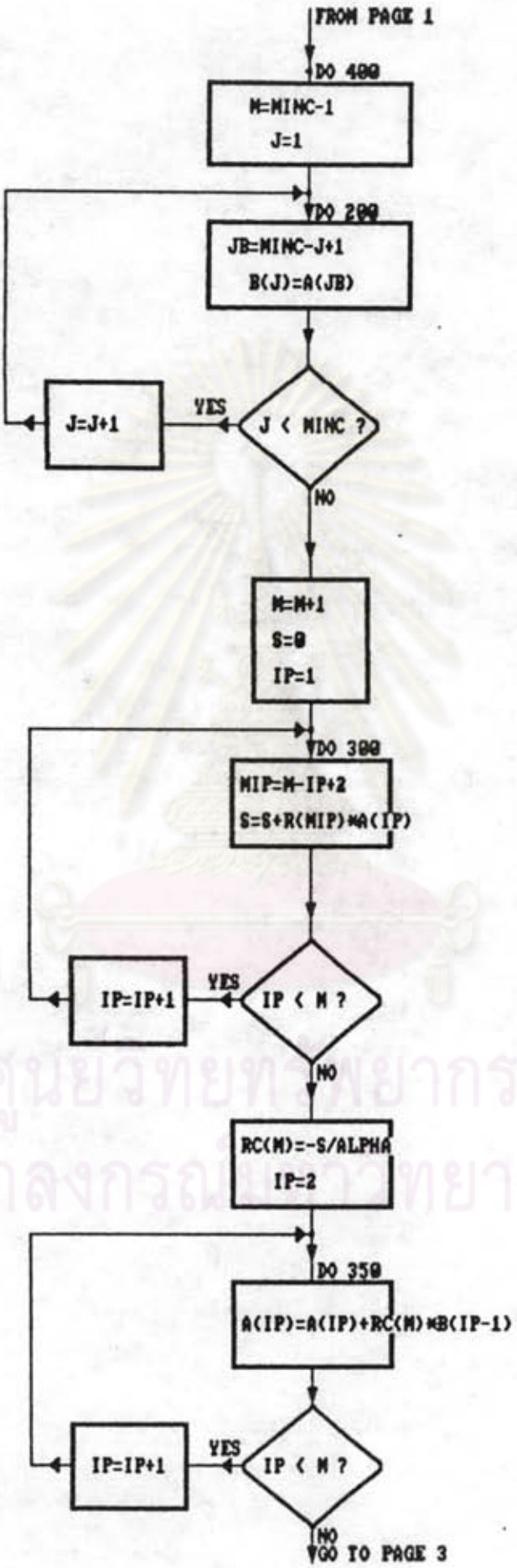
Schroeder M.R."Vocoders:Analysis and Synthesis of Speech." IEEE. May,1966.

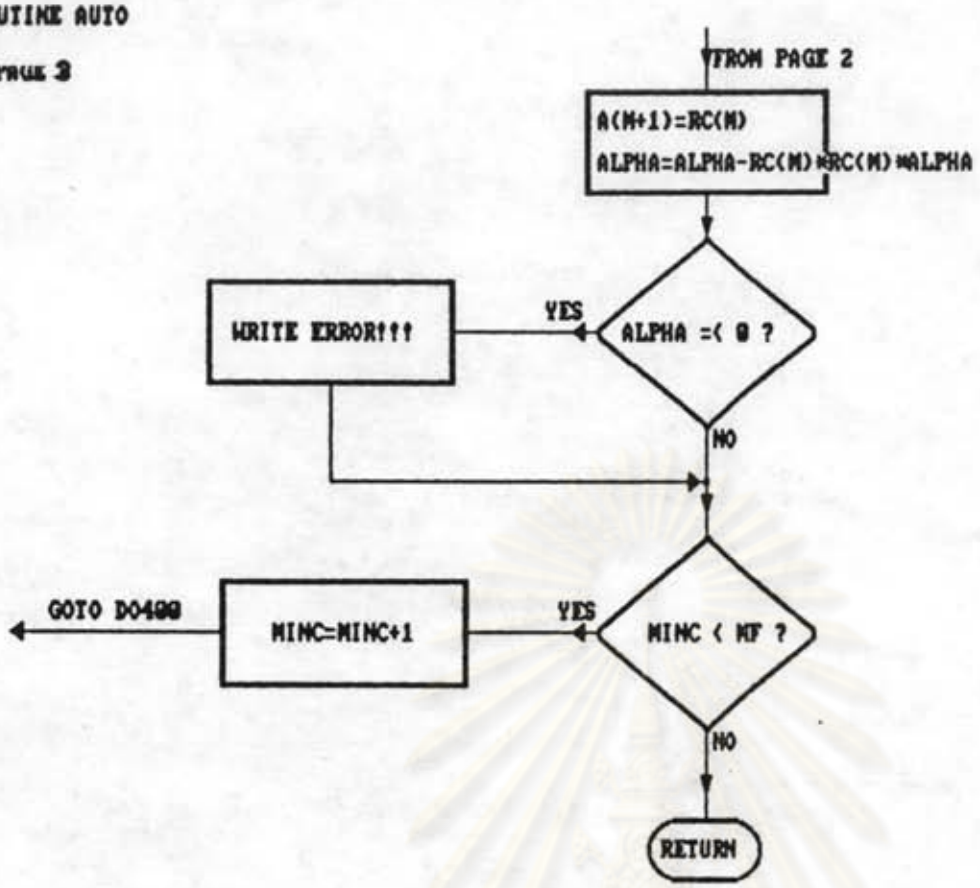
Rigsby M. Verbal Control with Microcomputer.: Tab Book Inc., 1982.

อุดม วโรดมสิกขดิตถ์. ภาษาศาสตร์ เบื้องต้น. มหาวิทยาลัยรามคำแหง, 2527.

ภาคผนวก ก. สังงานการหาค่าพารามิเตอร์ δ , สปส. การสะท้อนกลับ



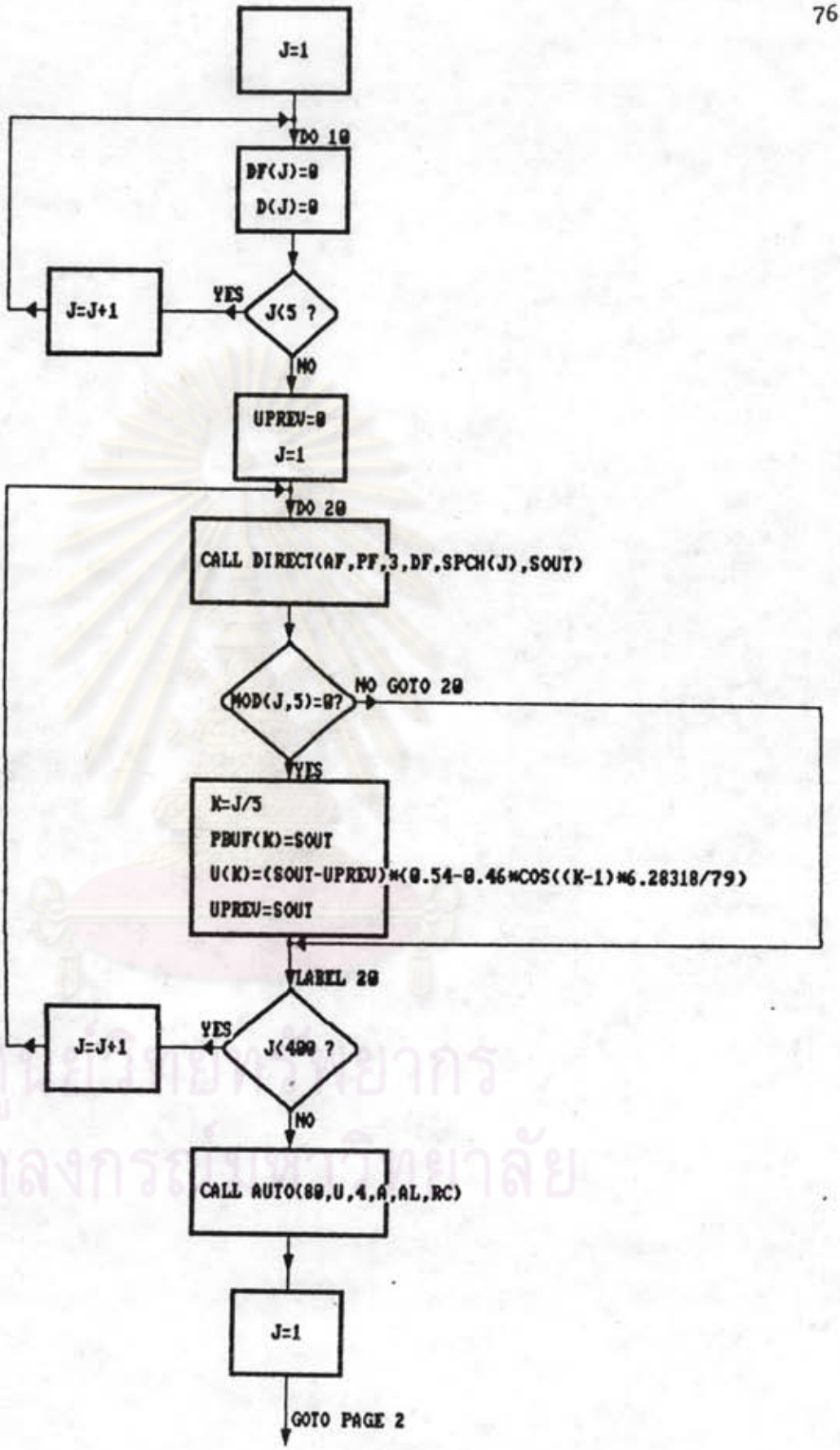




ภาคผนวก ข. ผังงานการหาค่า Pitch Period

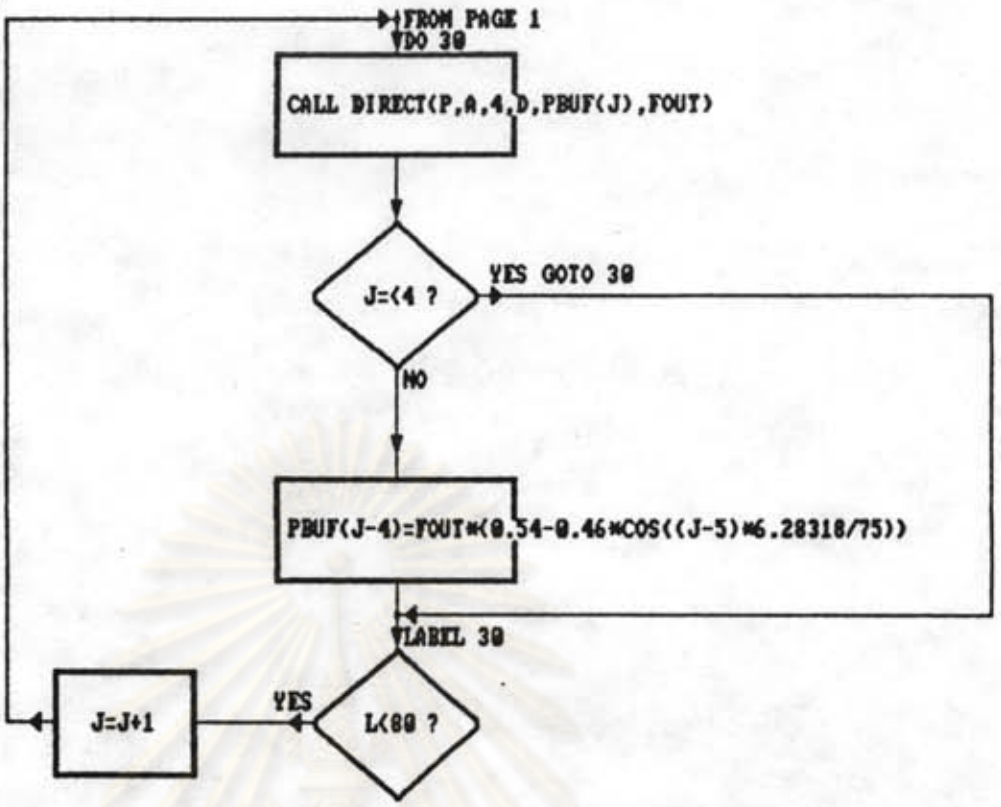
SUBROUTINE SIFTONE(SPCH,PBUF)

PAGE 1

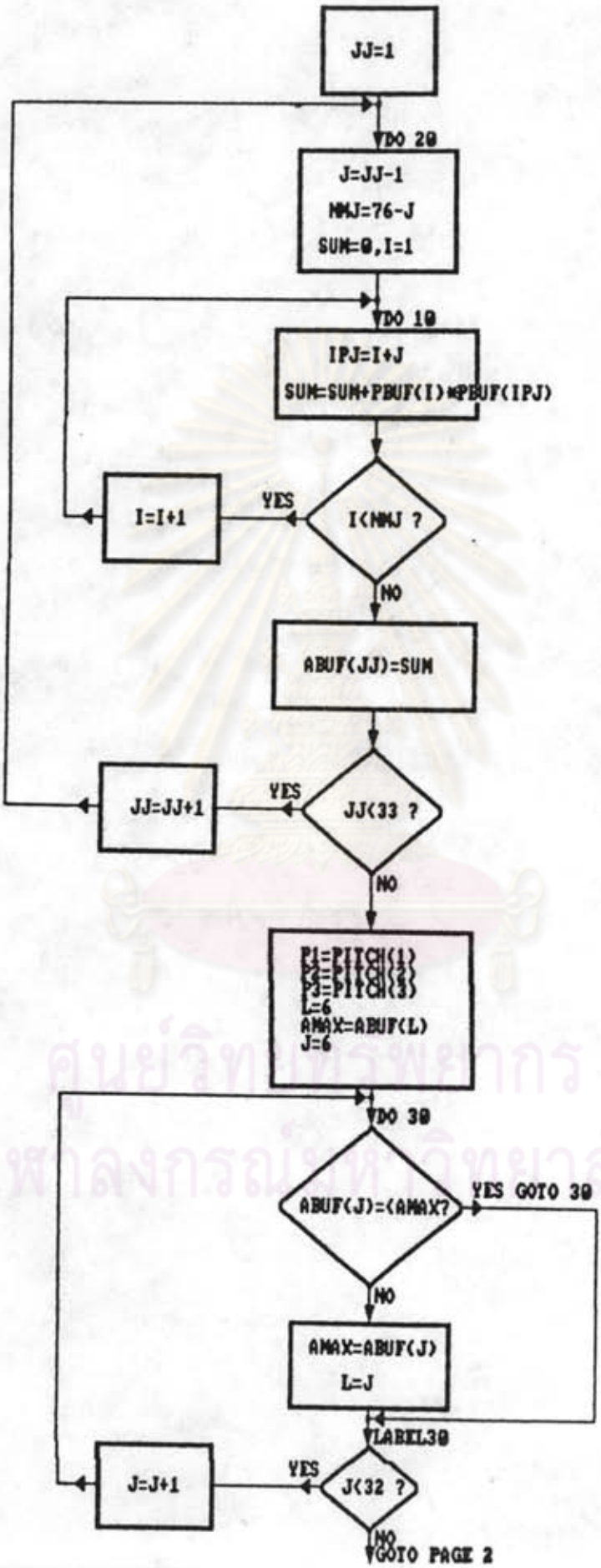


SUBROUTINE SIFTONE(SPCH,PBUF)

PAGE 2

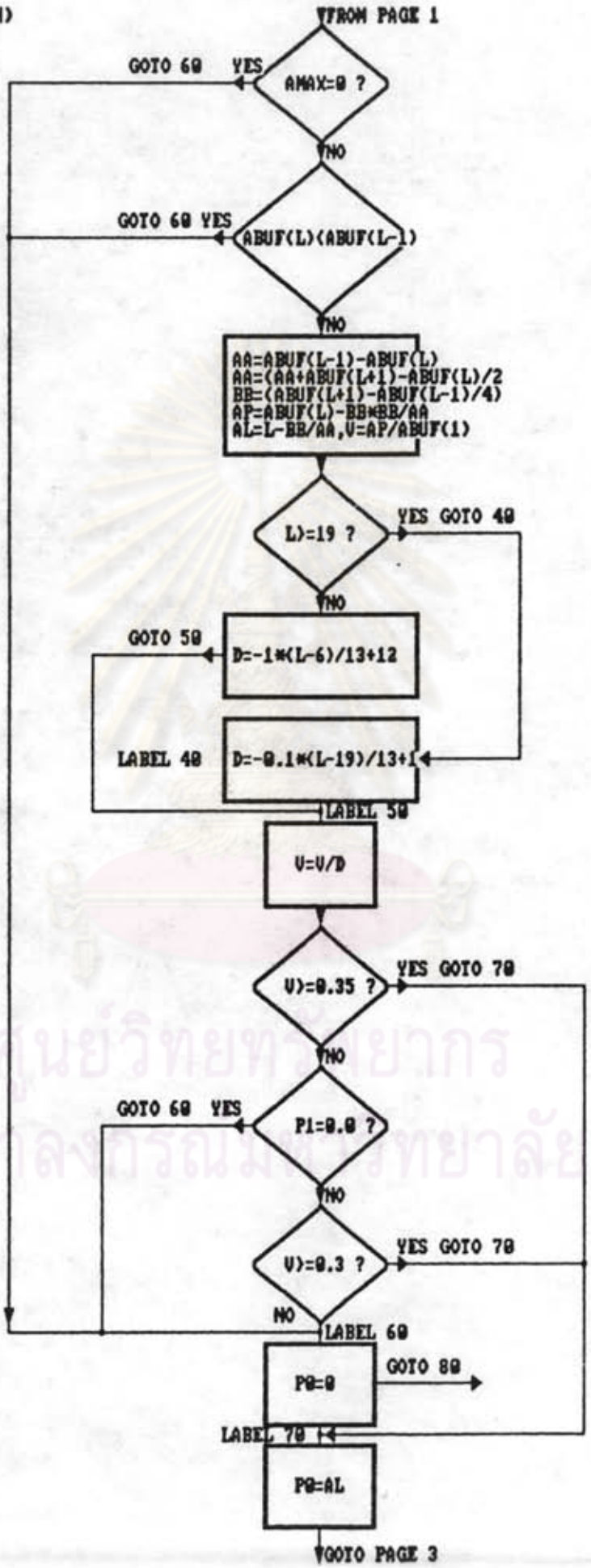


SUBROUTINE SIFTTWO(PBUF,PITCH)



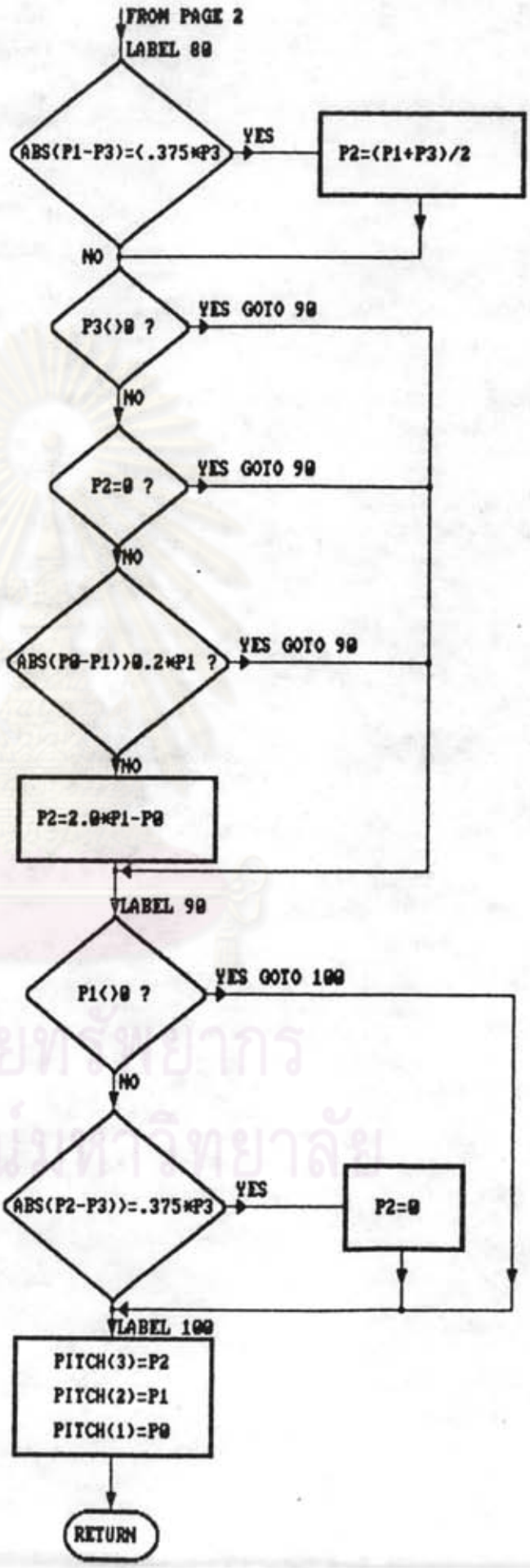
SUBROUTINE SIFTTWO(PBUF,PITCH)

PAGE 2



SUBROUTINE SIFTTWO(PBUF,PITCH)

PAGE 3



ภาคผนวก ค. ประโยคตัวอย่างที่ใช้ในการคัดพยางค์

ประโยคที่ใช้ในการทดลอง เพื่อคัดพยางค์ เป็นประโยคเกี่ยวกับชื่อและหัว ๆ ไปดังต่อไปนี้

ประโยคที่ 1 "นาย ชาลี พงษ์สมบูรณ์"
ประโยคที่ 2 "นาย ประกิต จิตมั่นคง"
ประโยคที่ 3 "นาย แก้วพันธุ์ ธีระโยธิน"
ประโยคที่ 4 "นาย ทวี ประทุมทาน"
ประโยคที่ 5 "นาย เชิดชัย เสริมคารงค์สักดิ์"
ประโยคที่ 6 "นาย อาจิน จิรชีพพัฒนา"
ประโยคที่ 7 "นาย ประยูร จินดาประดิษฐ์"
ประโยคที่ 8 "นาย เติมพันธุ์ บุนนาค"
ประโยคที่ 9 "นาง ประไพ ใจดี"
ประโยคที่ 10 "นาง สุชาดา ไชยประวัติ"
ประโยคที่ 11 "นางสาว บุศริน เจริญวัฒนกุลย์"
ประโยคที่ 12 "นางสาว พรทิพย์ ประทุมทาน"
ประโยคที่ 13 "นางสาว สุกัญญา เสริมคารงค์สักดิ์"
ประโยคที่ 14 "นางสาว มณี ชยานนท์"
ประโยคที่ 15 "นางสาว นันทิดา แก้วบัวสาย"

ประโยคที่ 16 "ลิงมักรนต้นไม้"
ประโยคที่ 17 "การควบคุมดูแล"
ประโยคที่ 18 "ก็บอกคุณแล้ว คุณไม่เชื่อ"
ประโยคที่ 19 "กลายเป็นเมืองชั้นมานานแล้ว"
ประโยคที่ 20 "สัตว์ทั้งหลายรอดชีวิตได้"
ประโยคที่ 21 "ทาดีได้ดี ทาชั่วได้ชั่ว"
ประโยคที่ 22 "คณะผู้เชี่ยวชาญด้านคอมพิวเตอร์"

ประโยคที่ 23 "คาดคะเนในสิ่งที่ยังหวังไม่ได้"

ประโยคที่ 24 "การเรียงลำดับคำยังไม่ดี"

ประโยคที่ 25 "นโยบายดังกล่าวมิได้มาจากความจริง"

ประโยคที่ 26 "ในอดีตเคยดำรงค์ตำแหน่งผู้จัดการ"

ประโยคที่ 27 "น้อยคนที่จะไม่เกี่ยวข้องกับคอมพิวเตอร์"

ประโยคที่ 28 "เป็นภาษาที่นิยมกันมาก"

ประโยคที่ 29 "จุฬาลงกรณ์มหาวิทยาลัย"

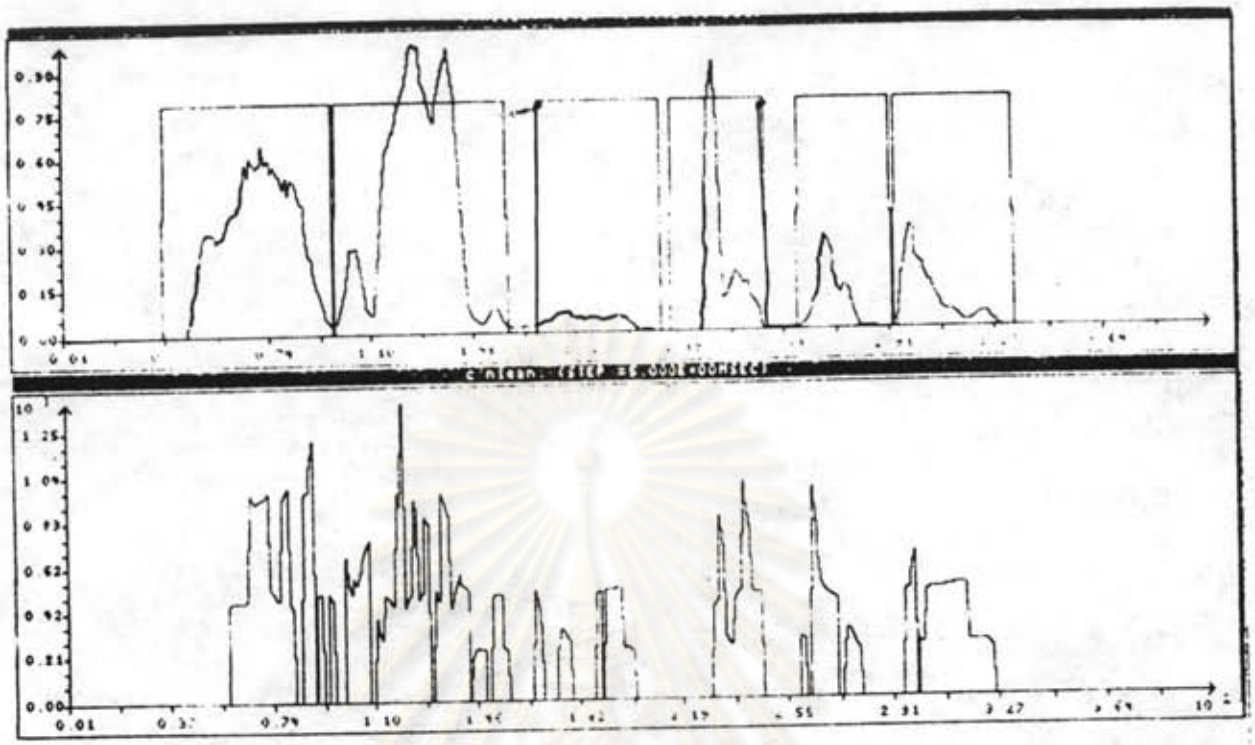
ประโยคที่ 30 "สถาบันเทคโนโลยีพระจอมเกล้า"

ภาคผนวก ง. ผลของการคัดพยางค์

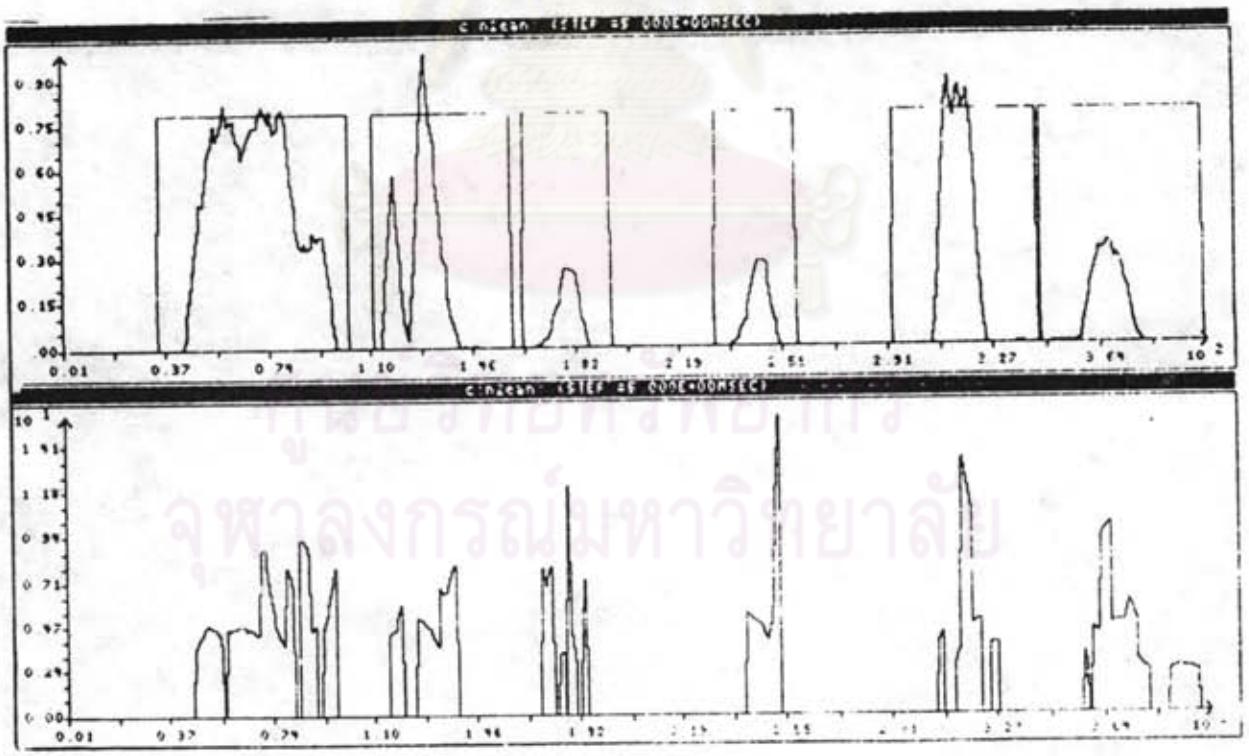
ในแต่ละประโยค รูปบนคือ ค่าพลังงาน

รูปล่างคือ Pitch Period

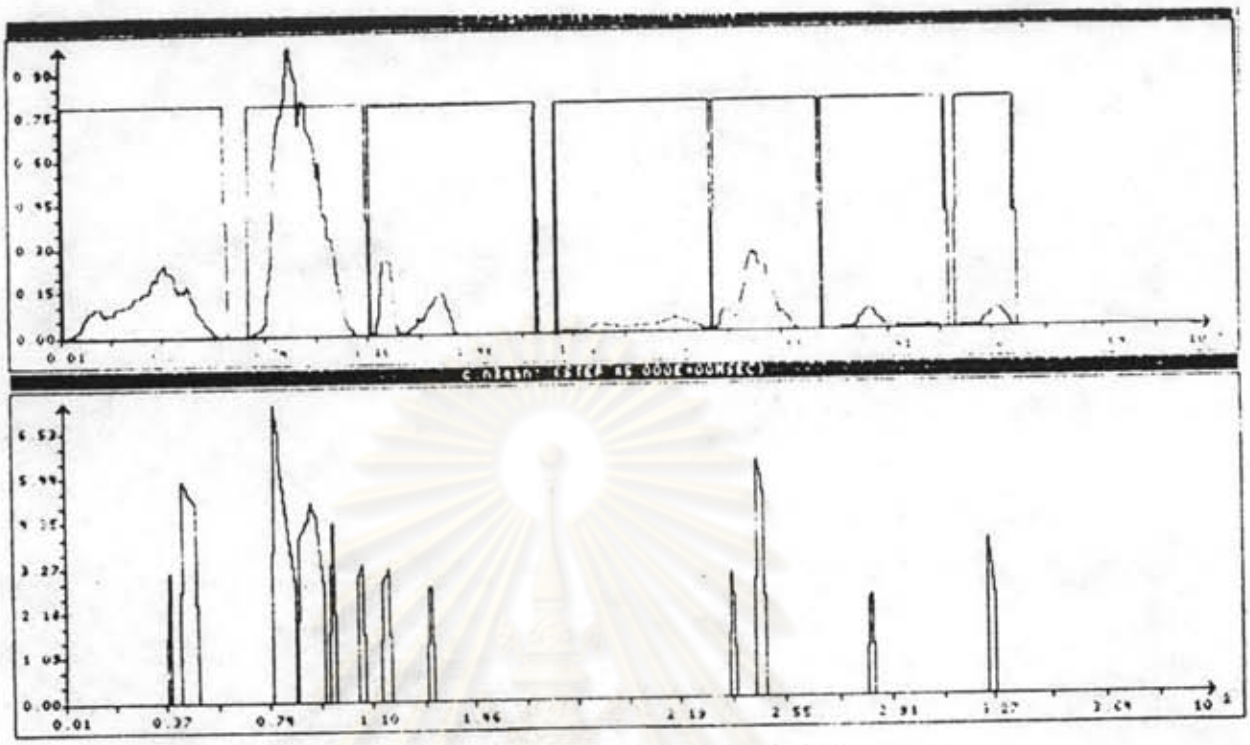
ตัวอย่างเสียงของ ช.



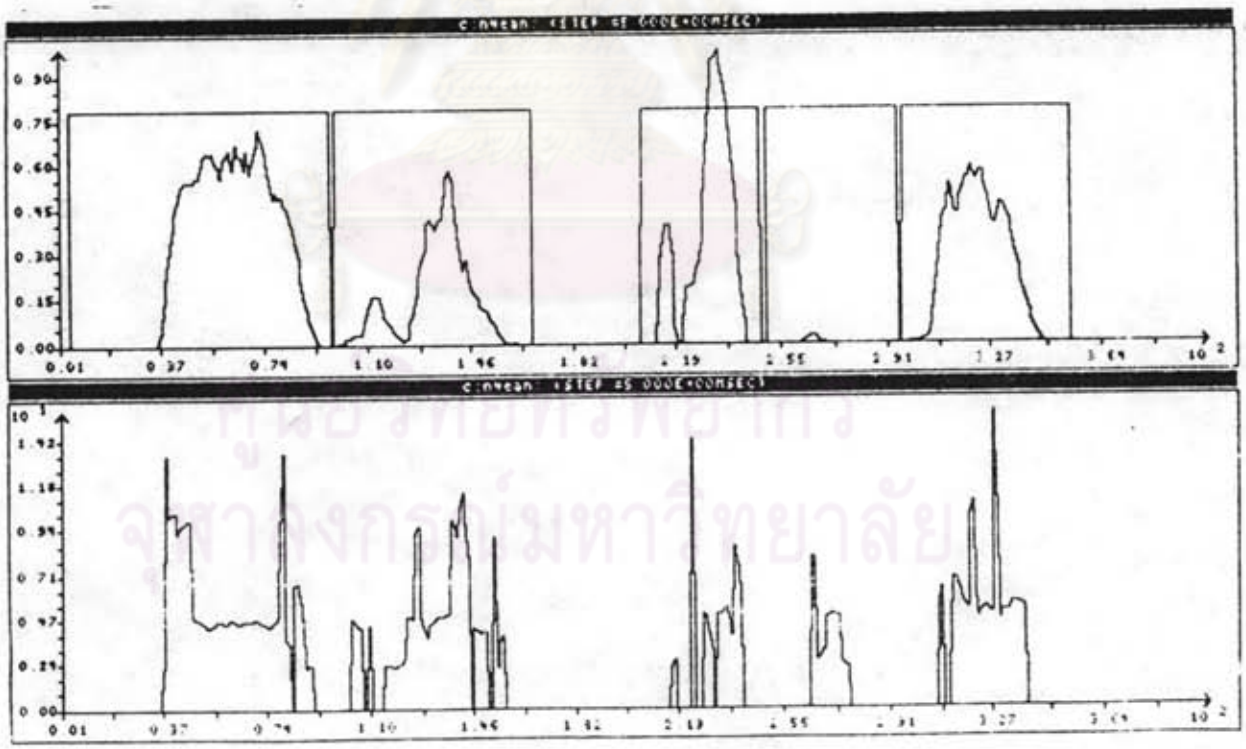
ประวัติคนที่ 1 "นาย ชาลี หงษ์สมบูรณ์"



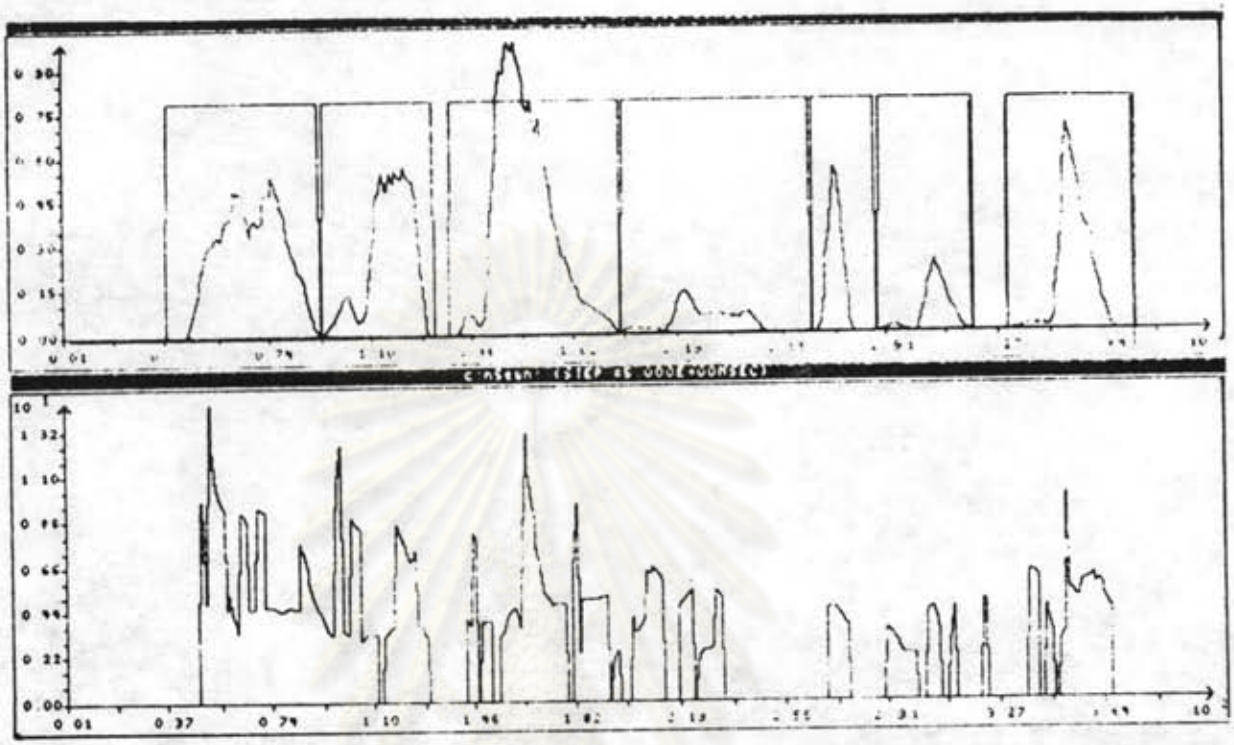
ประวัติคนที่ 2 "นาย ประดิษ จิตมั่นคง"



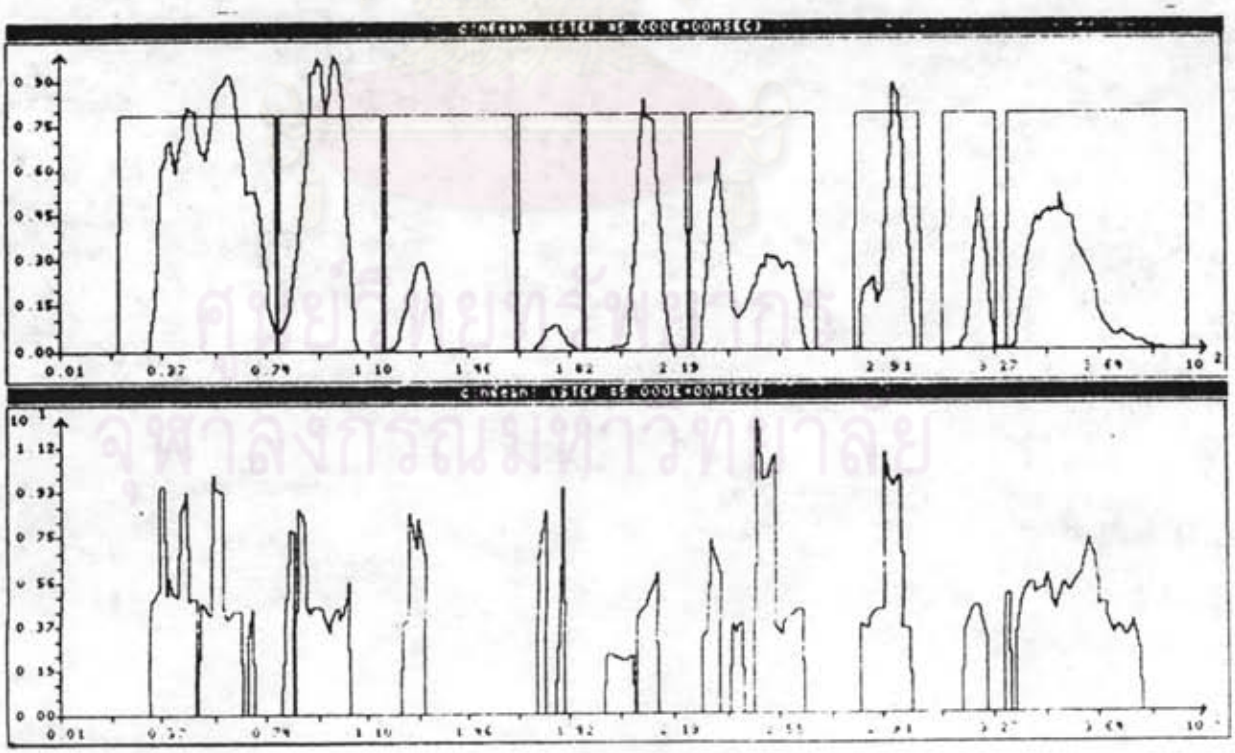
ประวัติที่ 3 "นาย แก้วพินธุ์ ชีระวานิน"



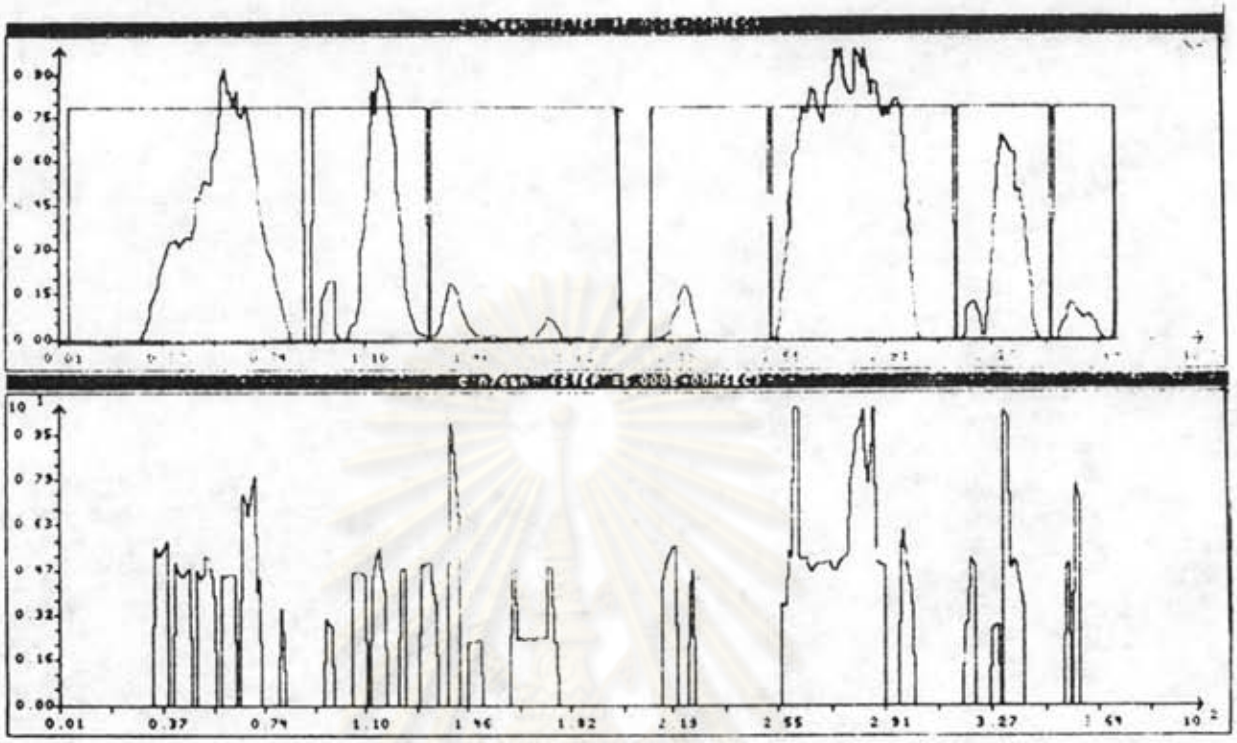
ประวัติที่ 4 "นาย ทวี ประทุมทาน"



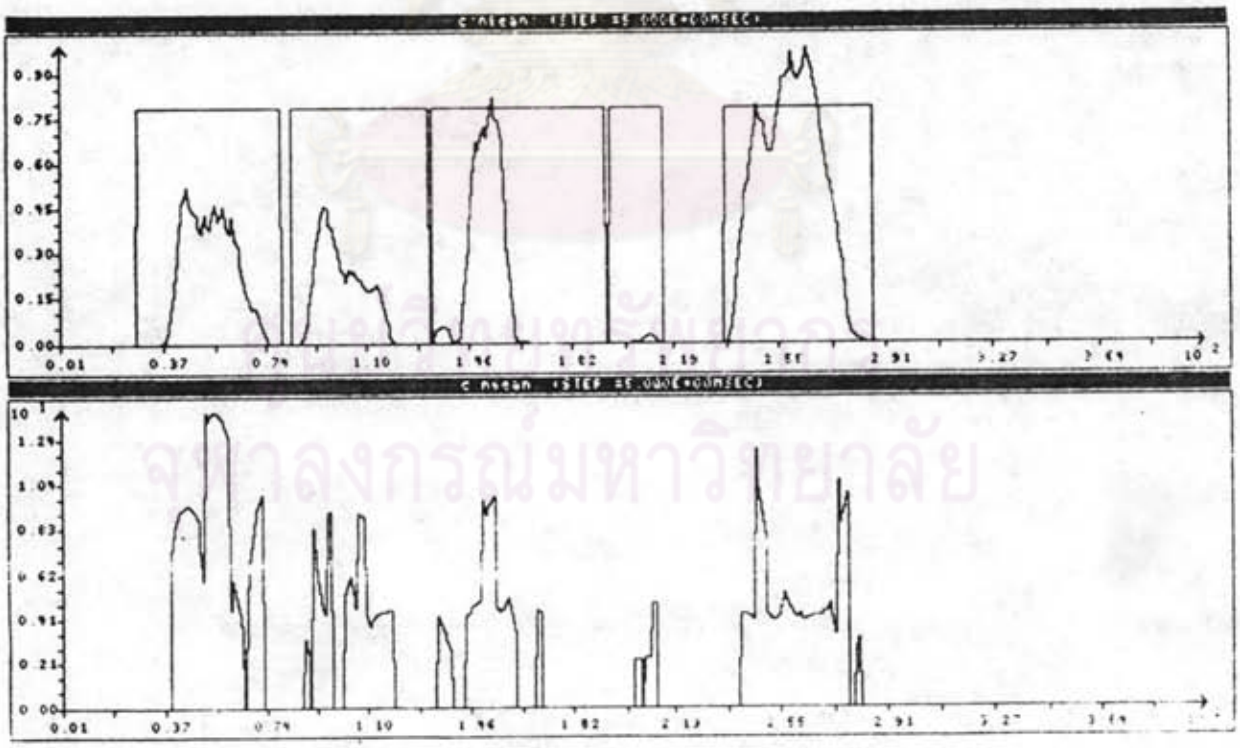
ประวัติที่ 5 "นาย เชิดชัย เสริมการงศักดิ์"



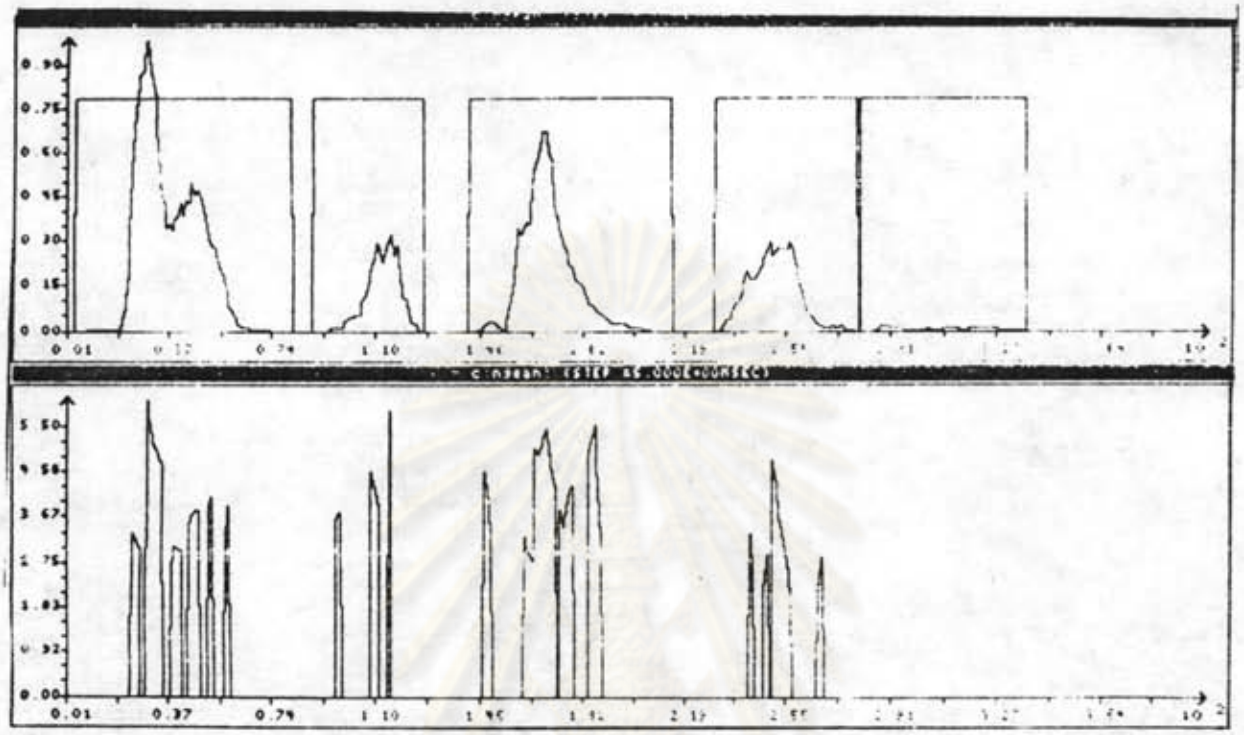
ประวัติที่ 6 "นาย อาจิน จิรชีพพัฒนา"



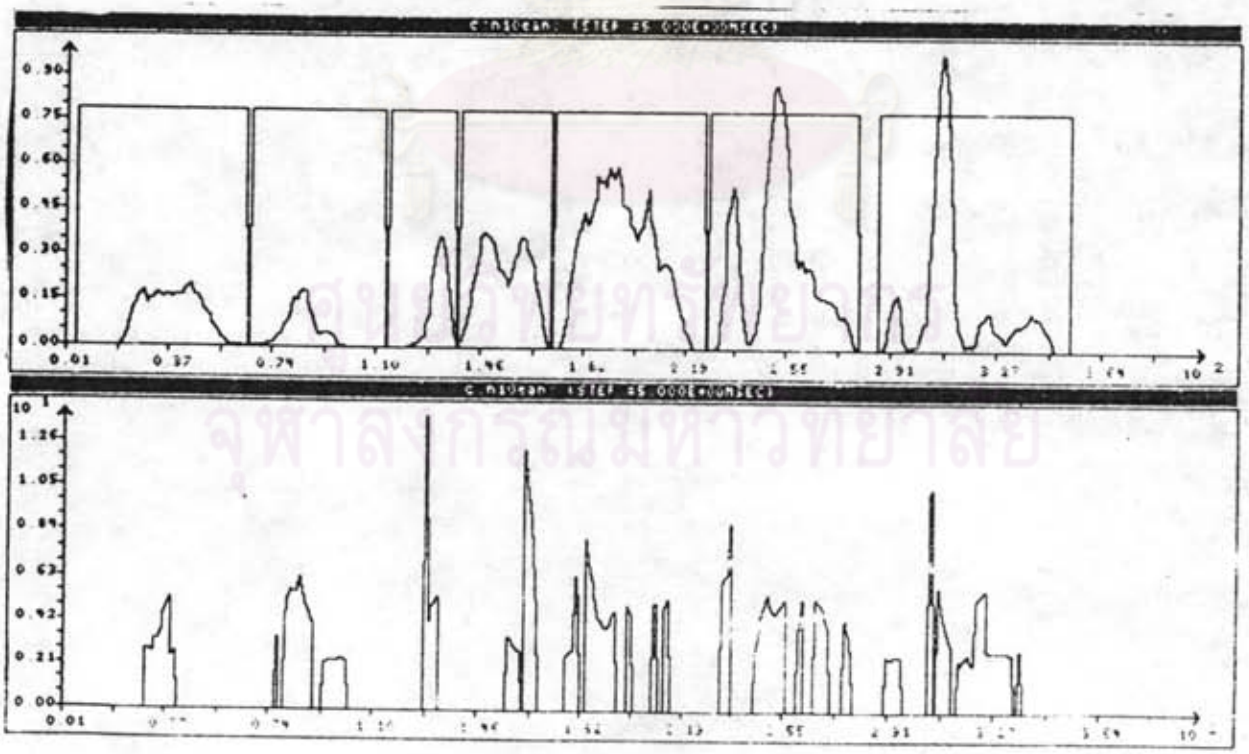
ประโยคที่ 7 "นาย ประยูร จินดาประดิษฐ์"



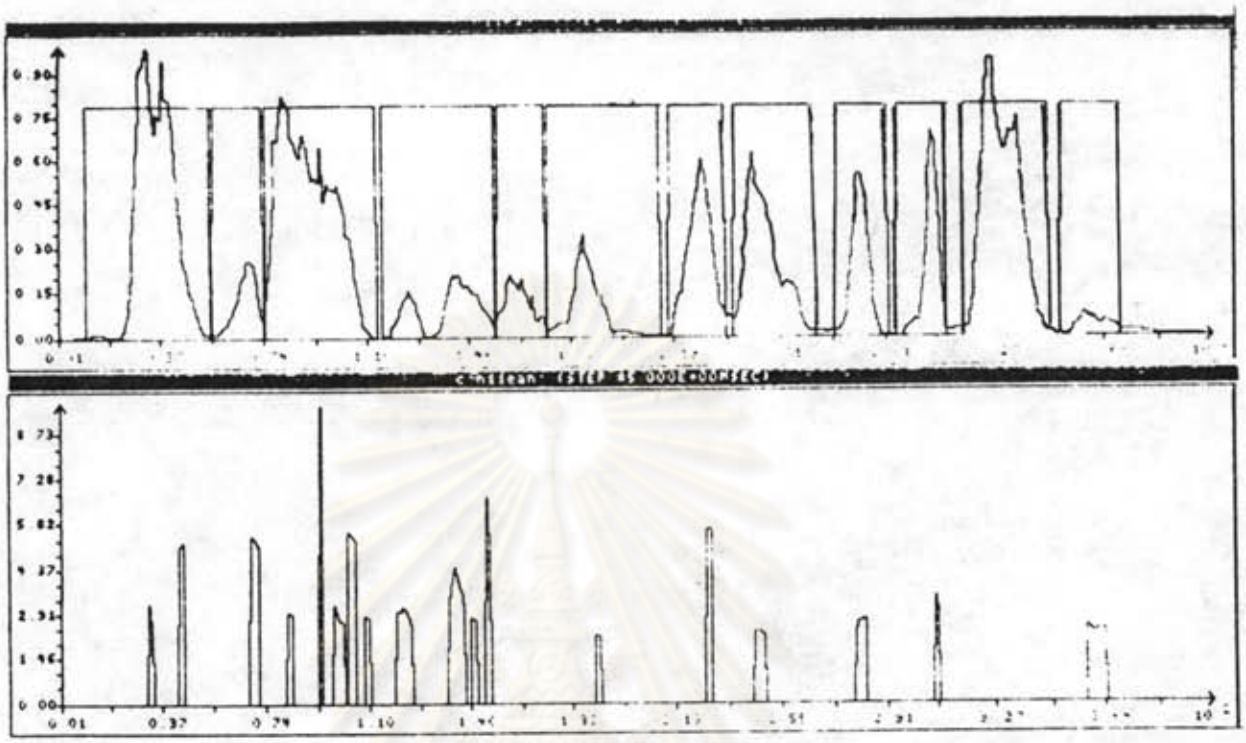
ประโยคที่ 8 "นาย เดิมพันธุ์ บุนนาค"



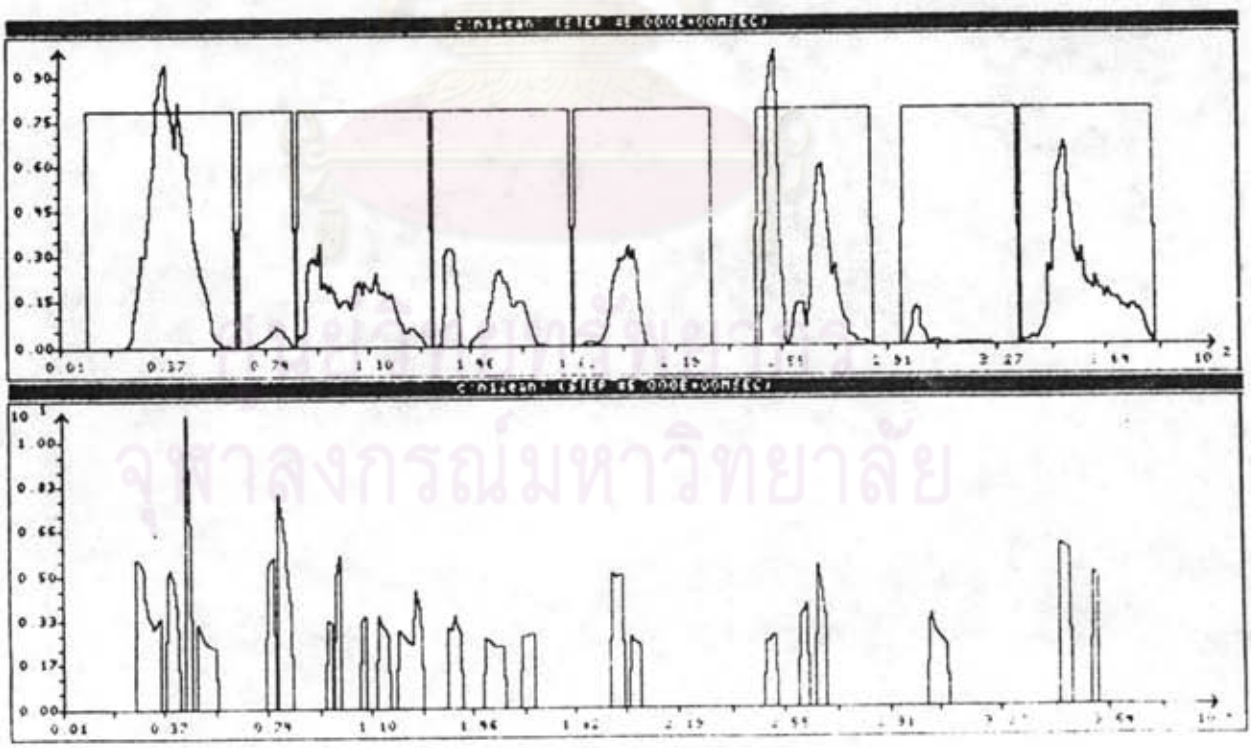
ประจักษ์ที่ 9 "นาง ประไพ วงศ์"



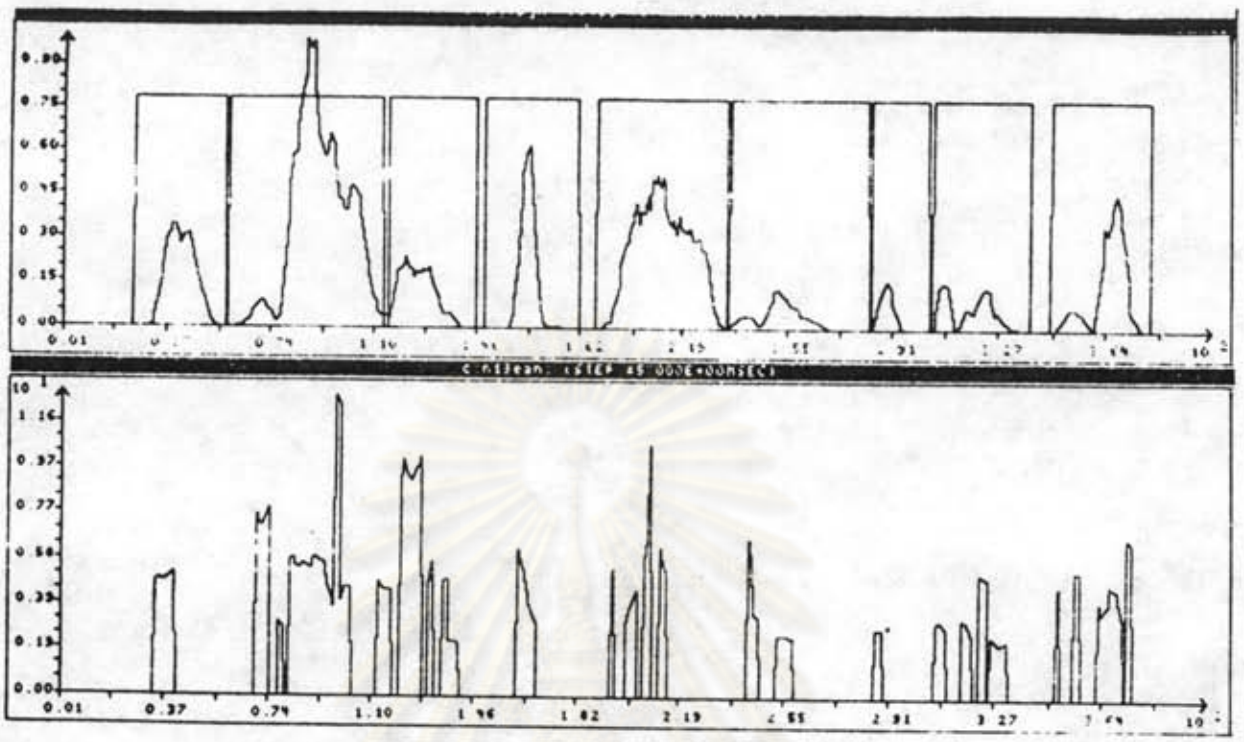
ประจักษ์ที่ 10 "นาง สุชาดา ชัยประวัติ"



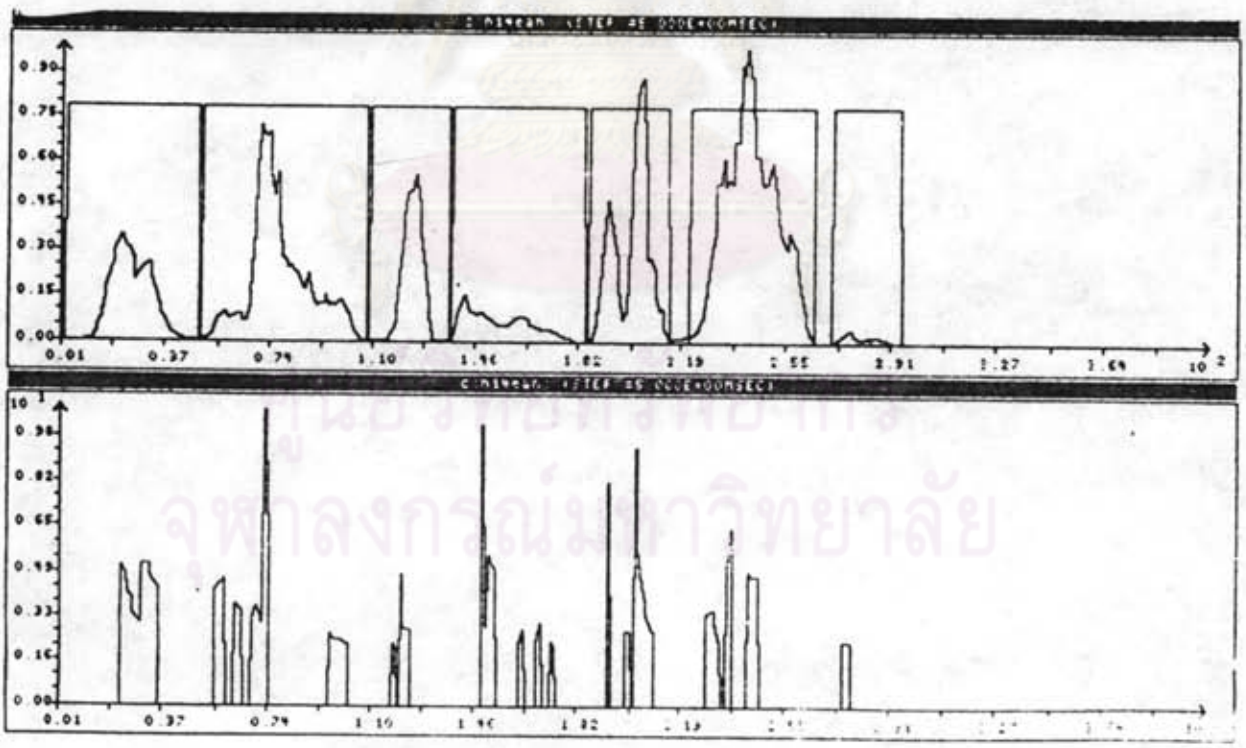
ประวัติที่ 11 "นางสาว บุศริน เจริญวัฒนกุลย์"



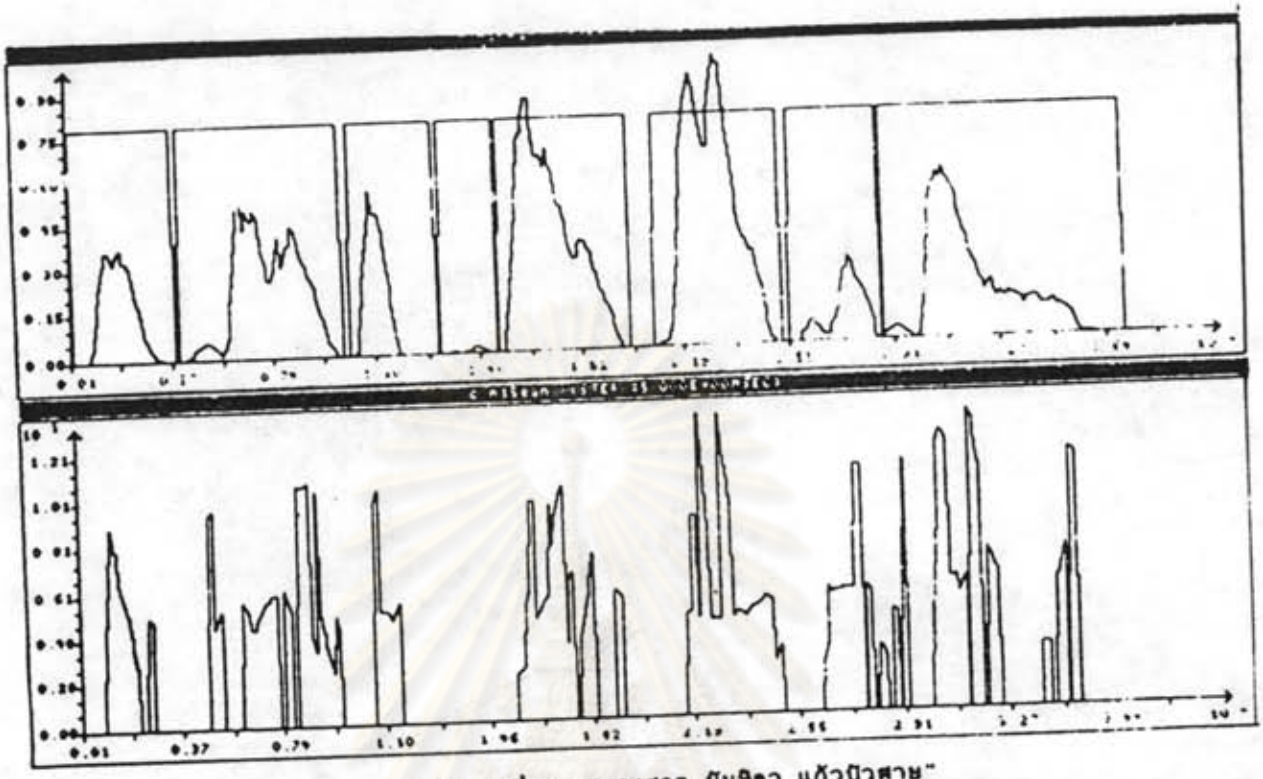
ประวัติที่ 12 "นางสาว พรทิพย์ ประทุมทาน"



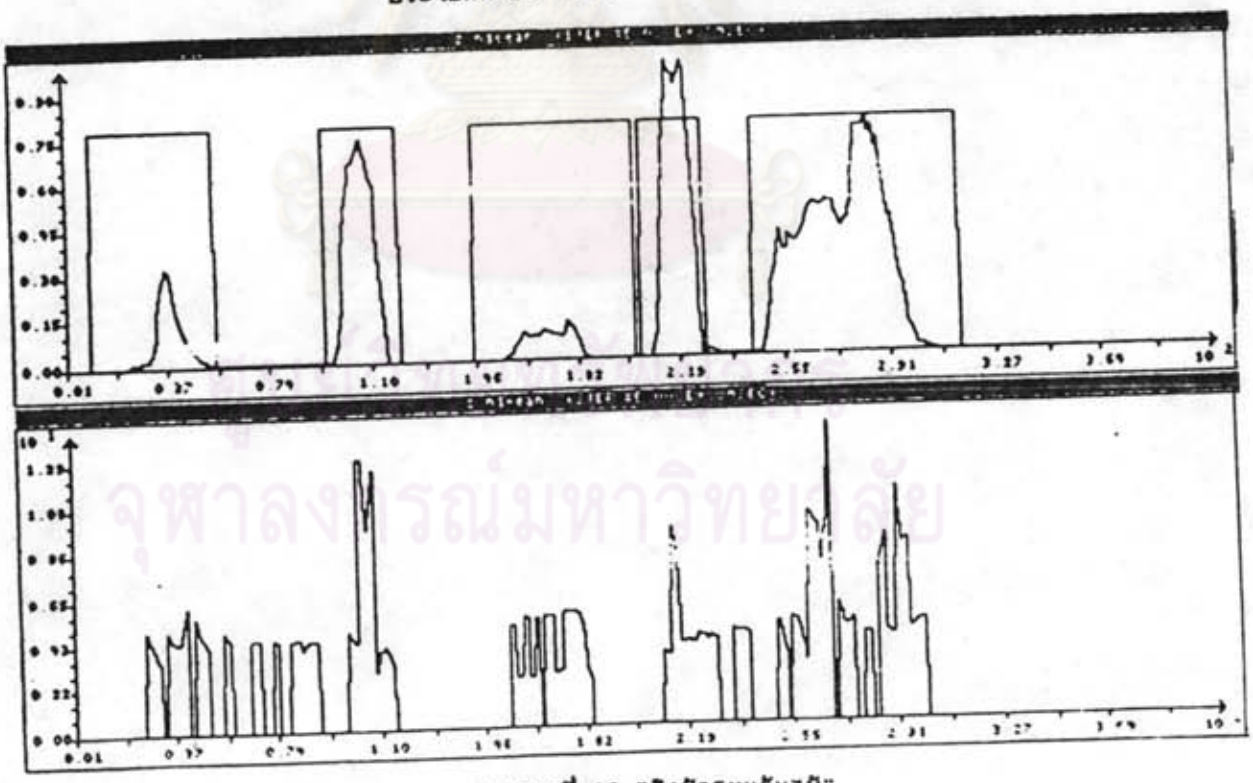
ประวัติคนที่ 13 "นางสาว สุกัญญา เสริมศารงศ์สักดิ์"



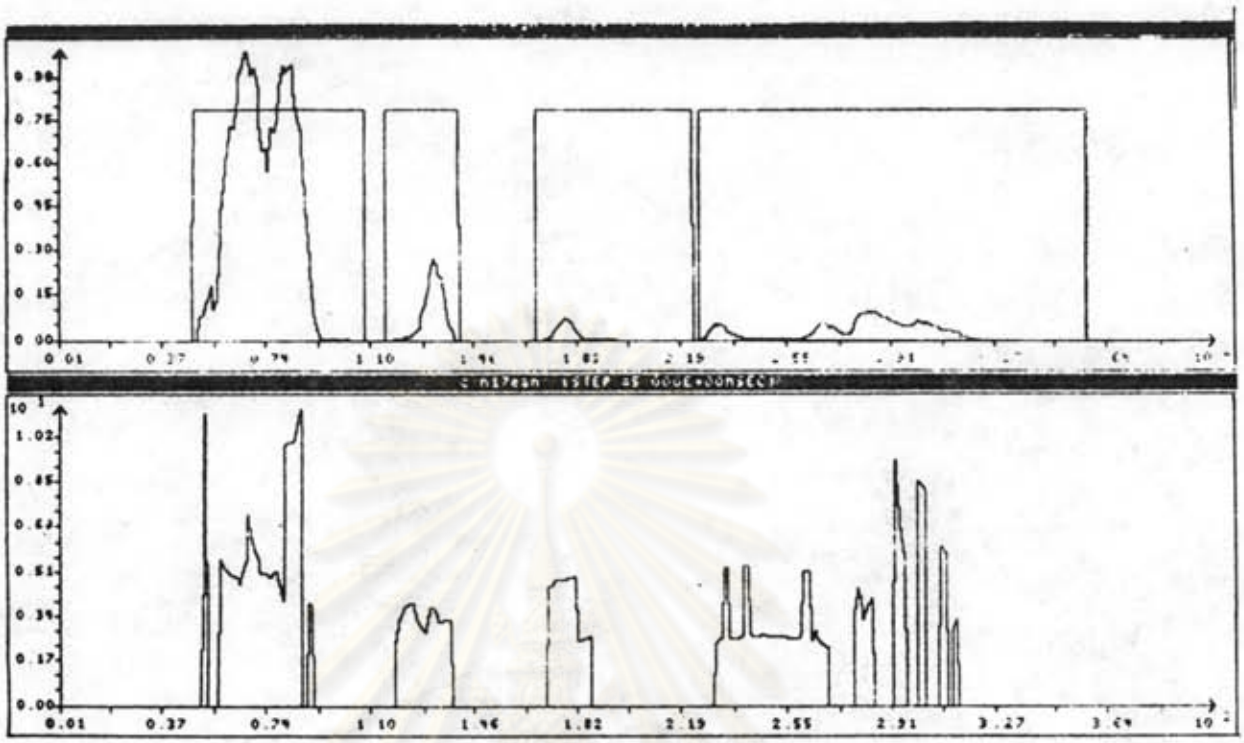
ประวัติคนที่ 14 "นางสาว มณี ชยานนท์"



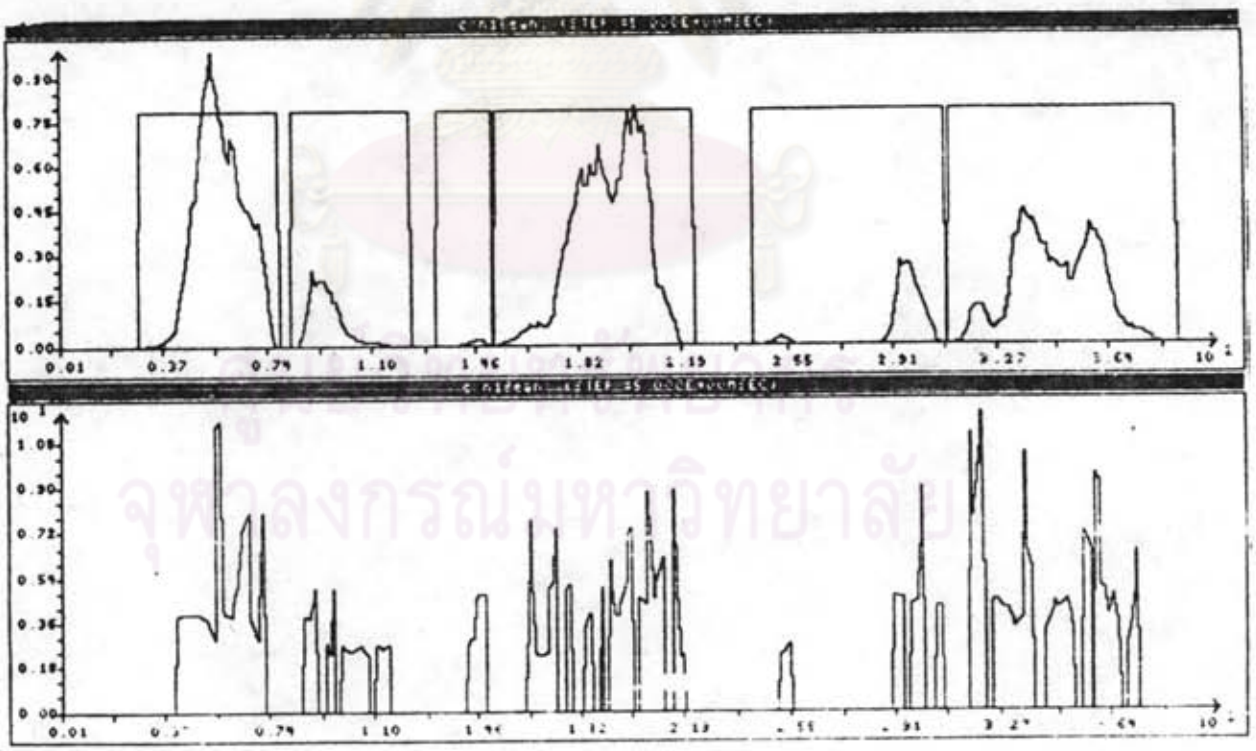
ประยุทธที่ 15 "นางสาว ปิยศิลา แก้วบัวสาย"



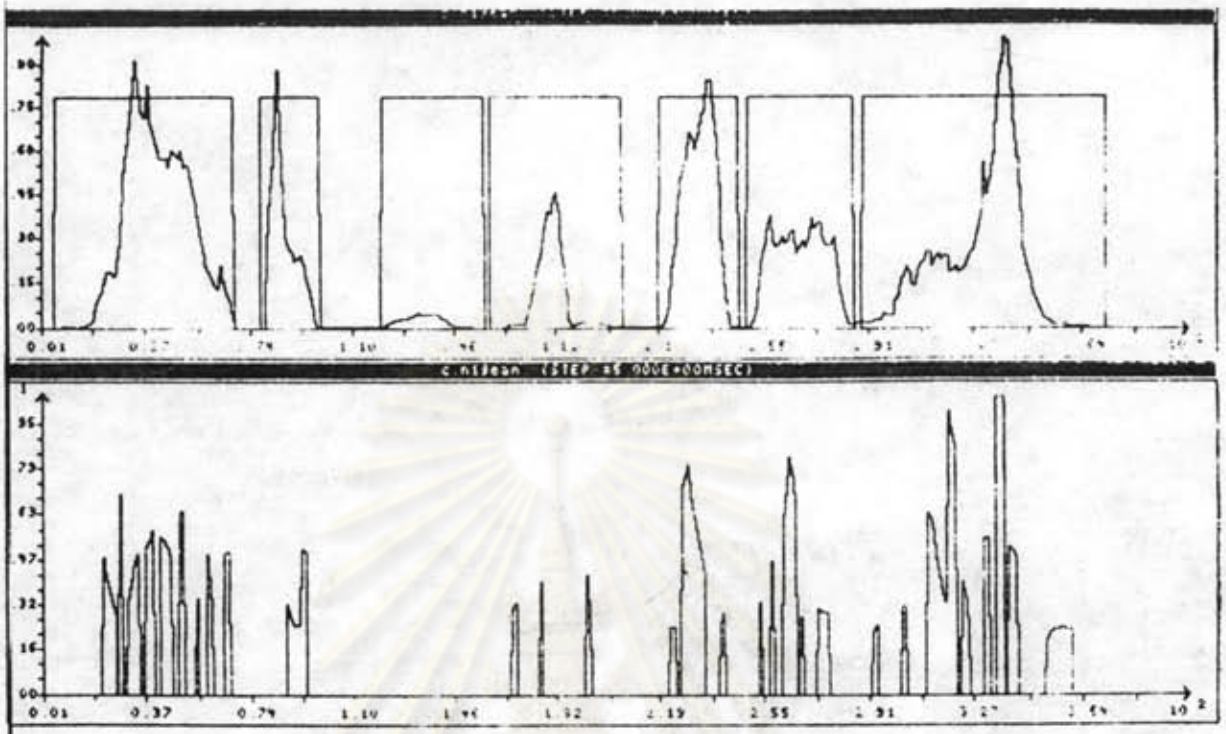
ประยุทธที่ 16 "สิงฆิกาหนดันาม"



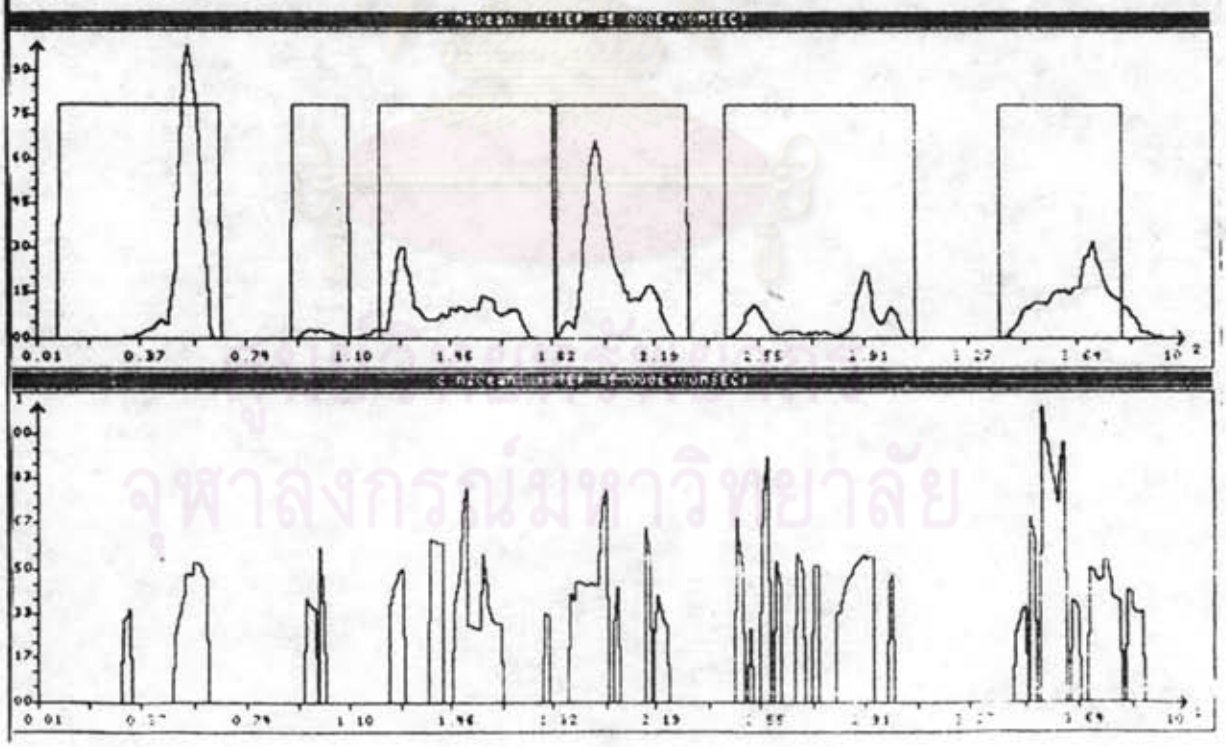
ประจำบทที่ 17 "การควบคุมดูแล"



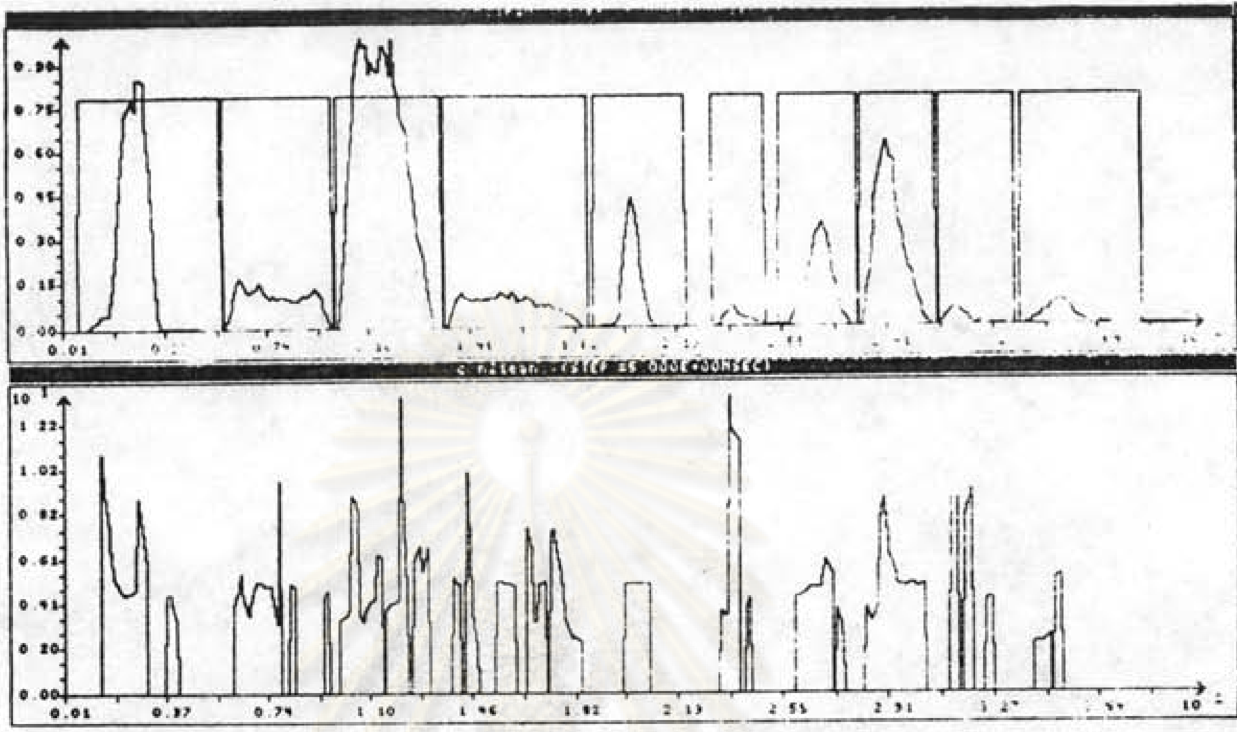
ประจำบทที่ 18 "ก็บอกคุณแล้ว คุณไม่เชื่อ"



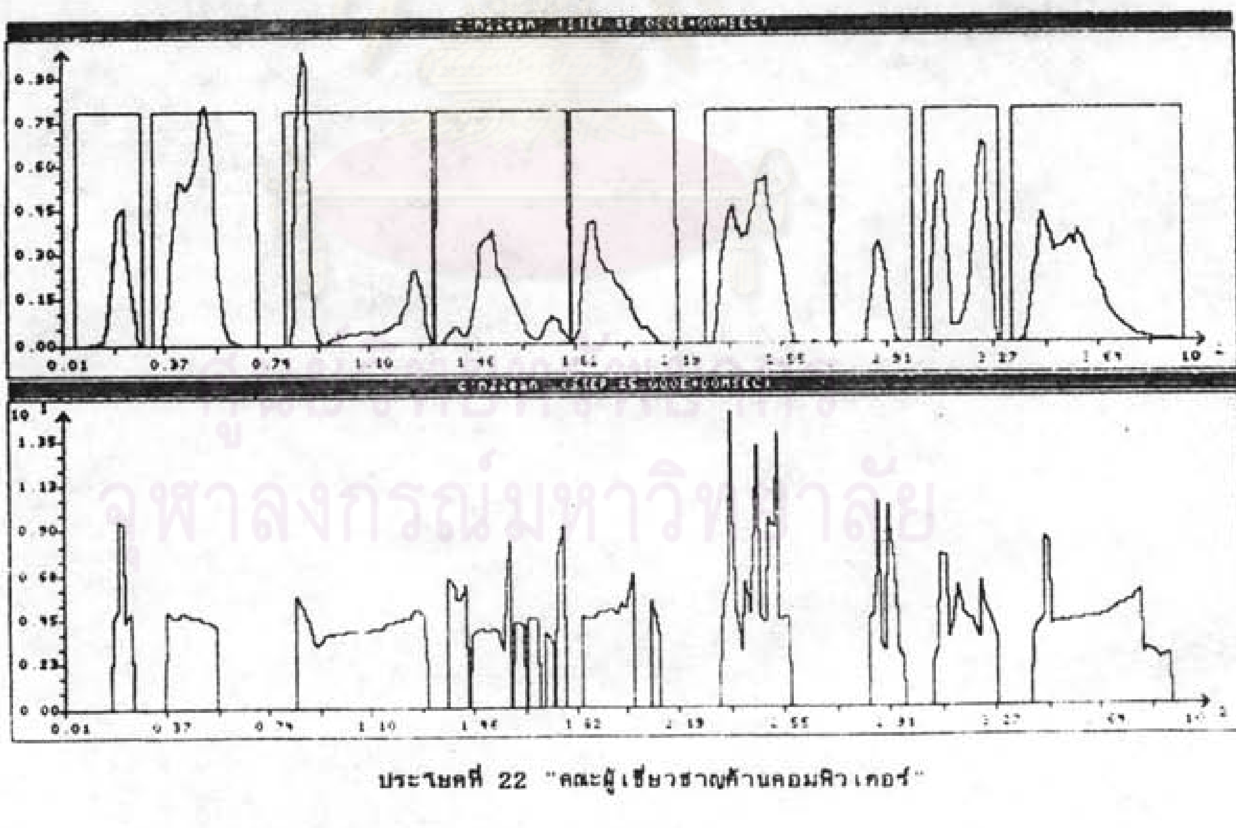
ประจายคที่ 19 "กลายเป็นเมืองขึ้นมานานแล้ว"



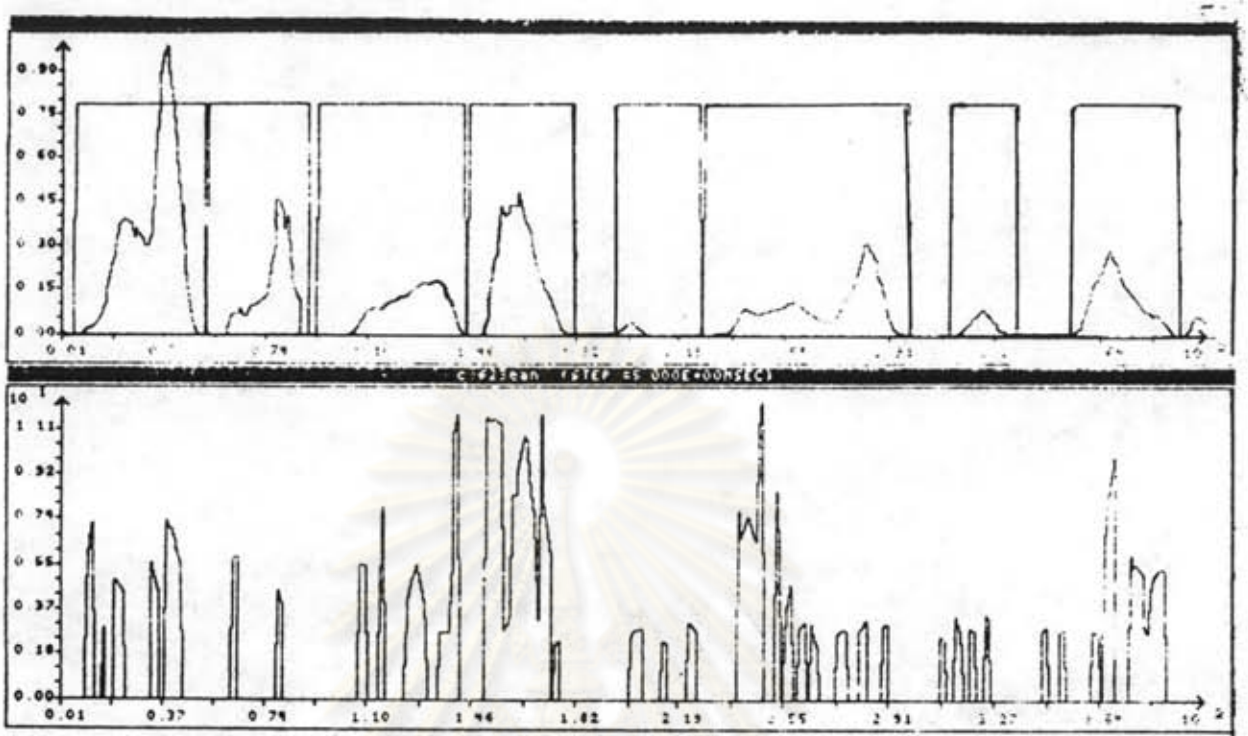
ประจายคที่ 20 "สิทธิ์ทั้งหลายรอดชีวิตได้"



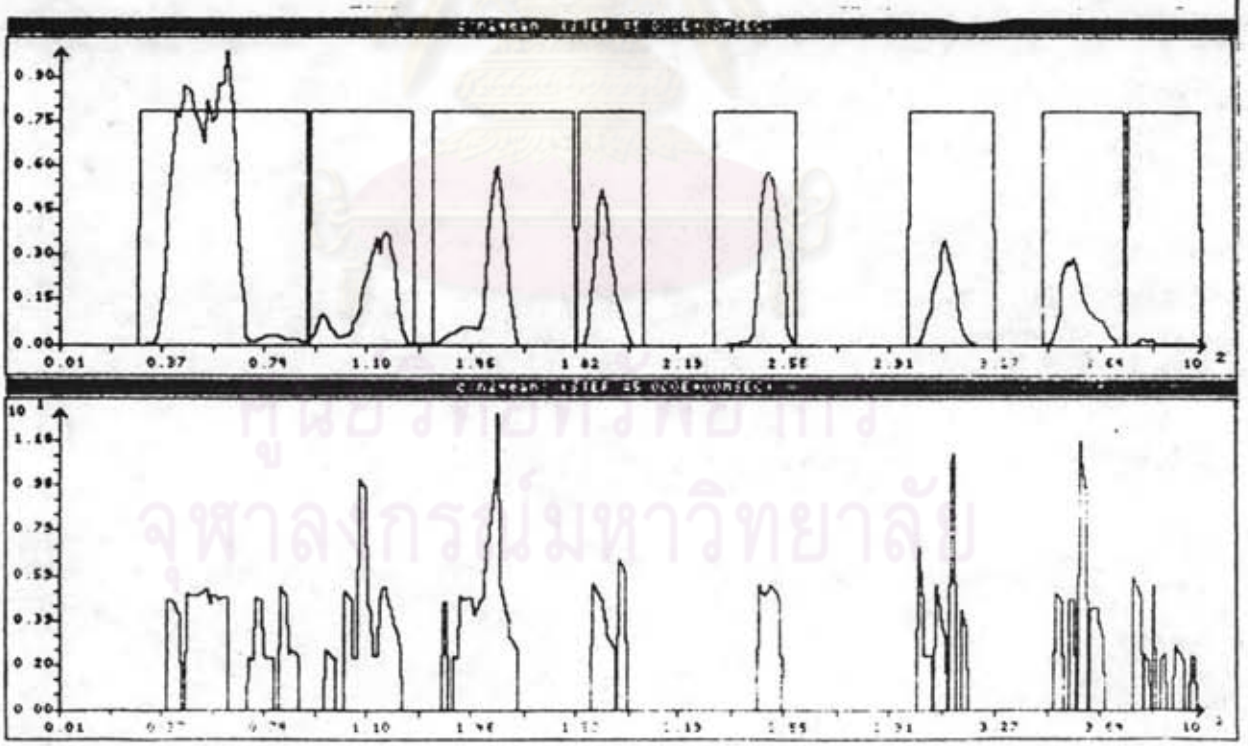
ประโยคที่ 21 "ทำดีได้ดี ทำชั่วได้ชั่ว"



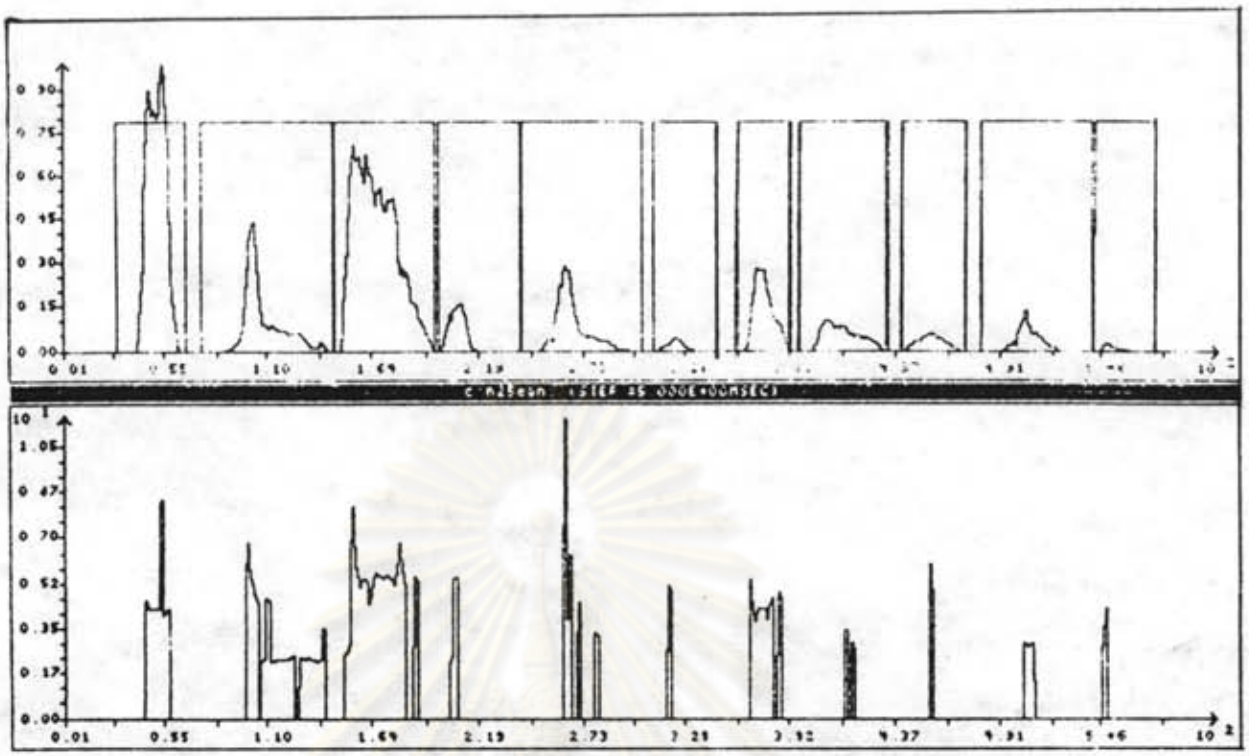
ประโยคที่ 22 "คณะผู้เชี่ยวชาญด้านคอมพิวเตอร์"



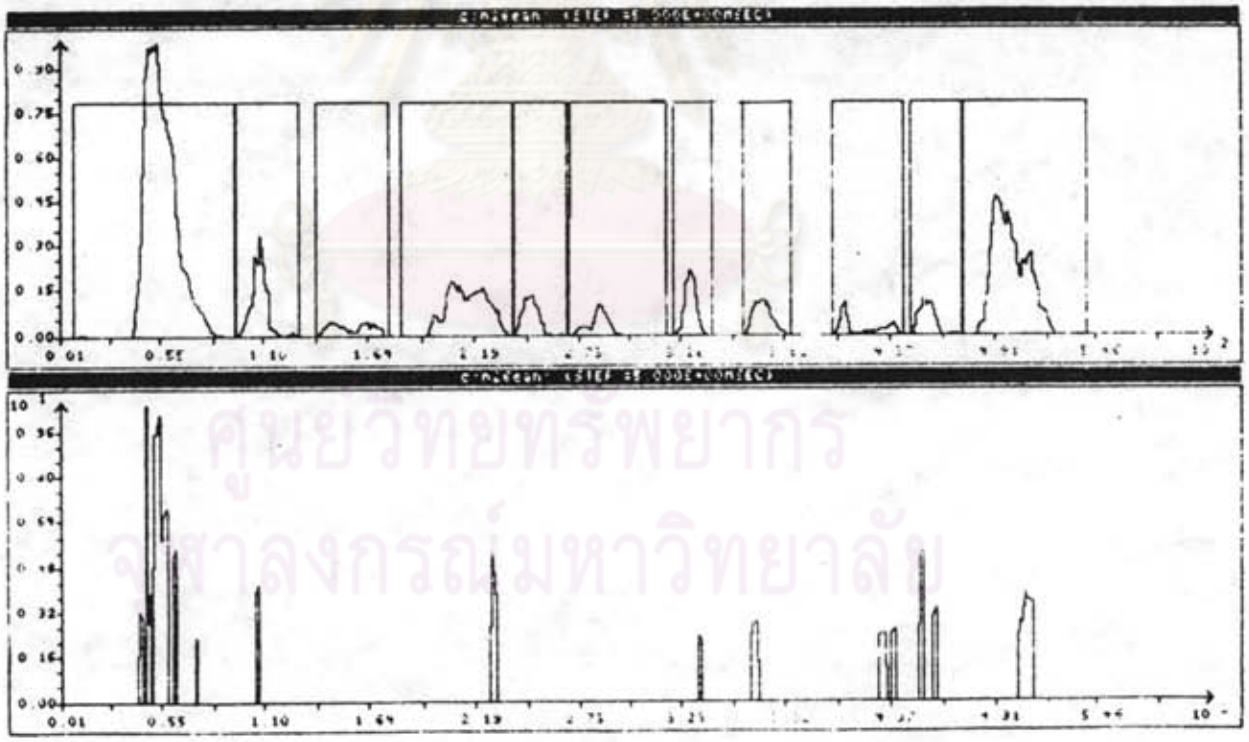
ประจวบที่ 23 "ตาคดะเนานสิ่งที่ยังหวังไม่ได้"



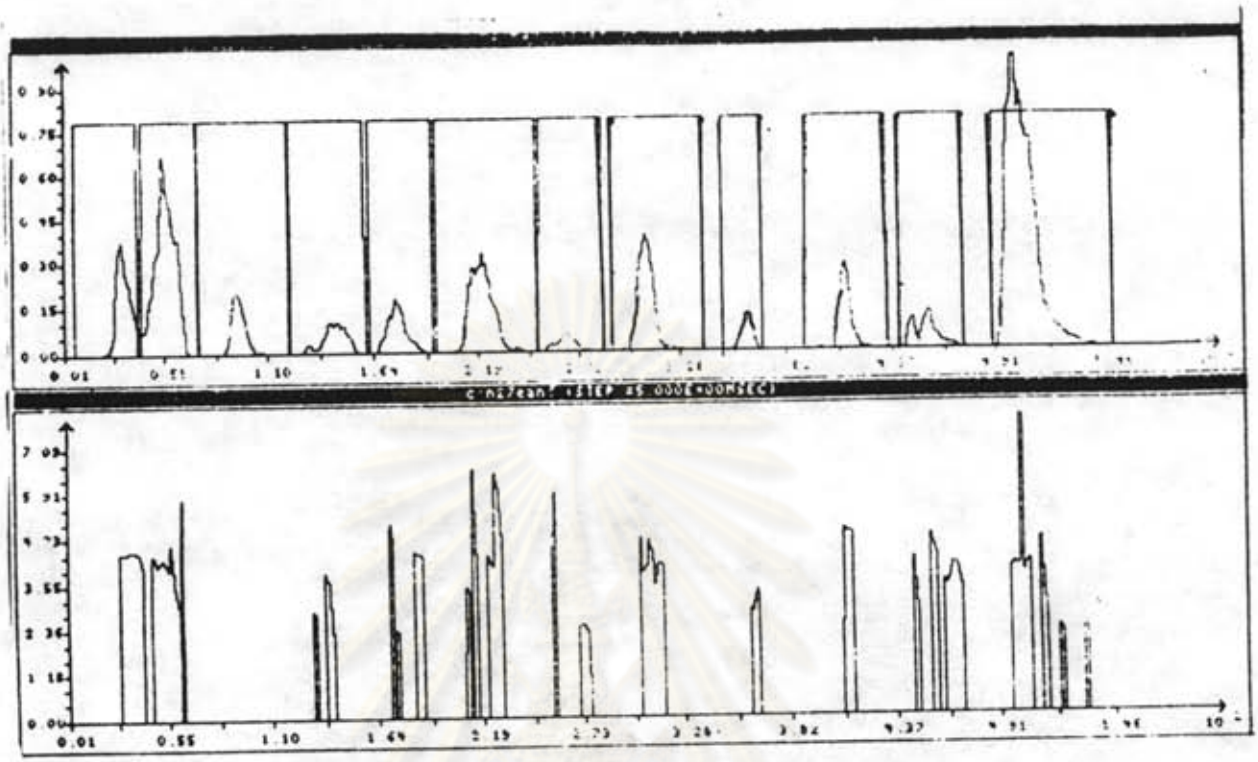
ประจวบที่ 24 "การเรียงตาคับตายังไม่ดี"



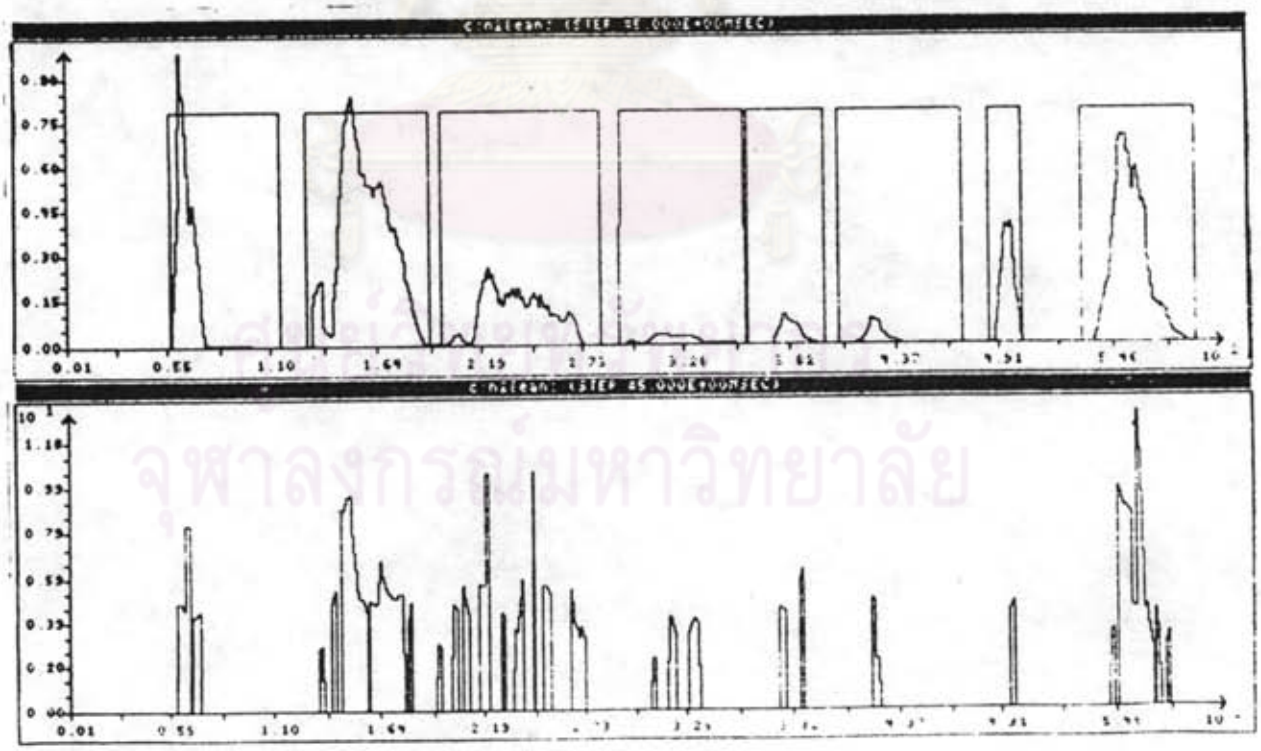
ประโยคที่ 25 "นายบายดังกล่าวมิได้มาจากความจริง"



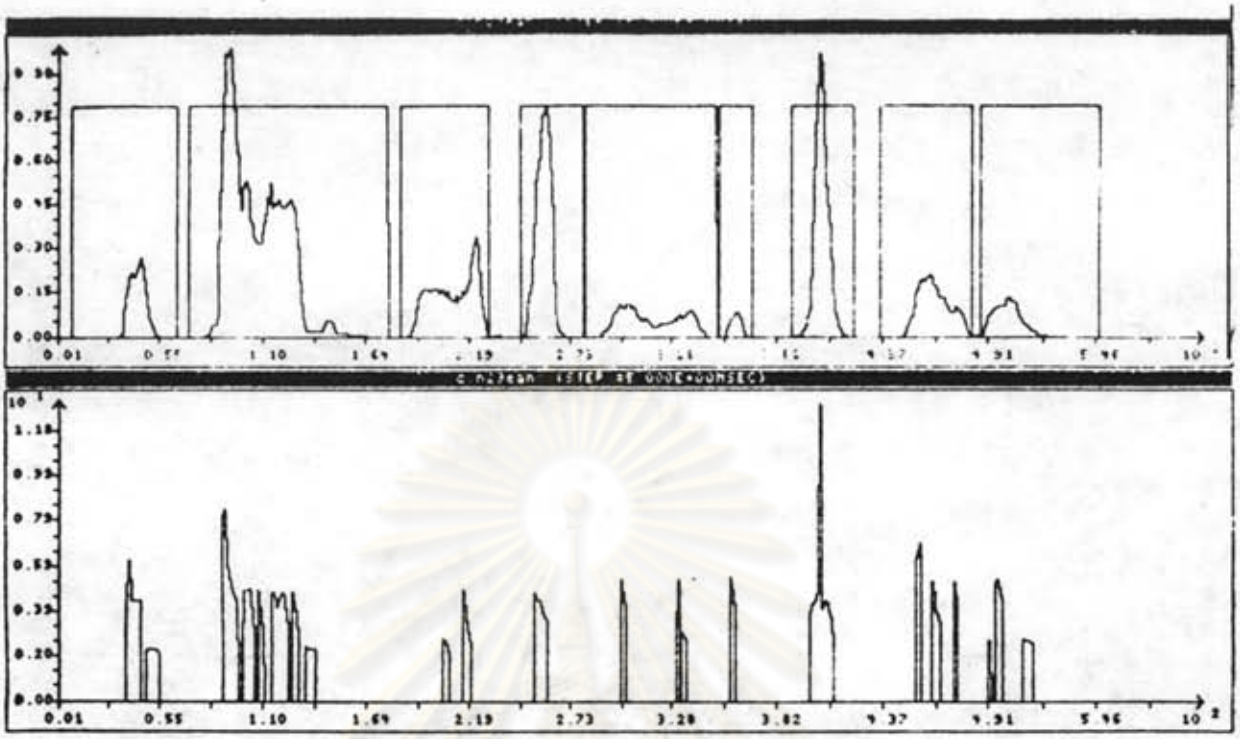
ประโยคที่ 26 "ในอดีตเคยดำรงตำแหน่งผู้จัดการ"



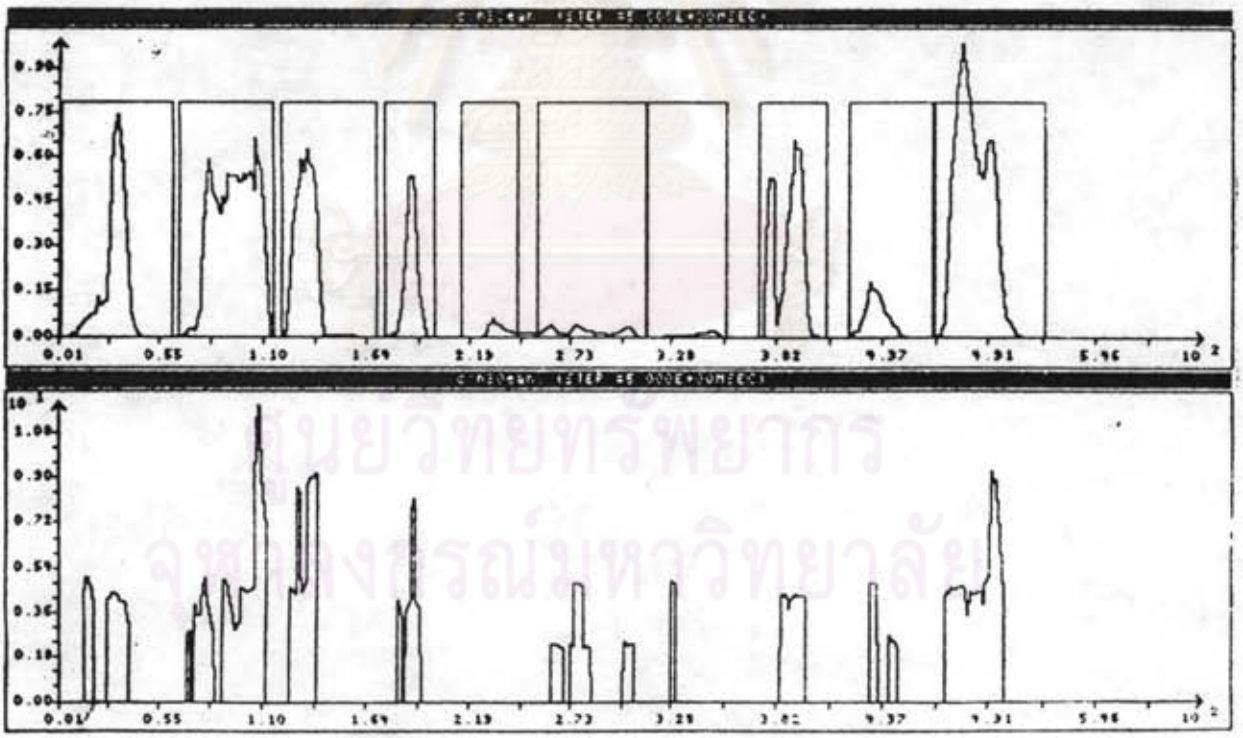
ประจักษ์ที่ 27 "น้อยคนที่จะไม่เกี่ยวข้องกับคอมพิวเตอร์"



ประจักษ์ที่ 28 "เป็นภาษาที่นิยมกันมาก"

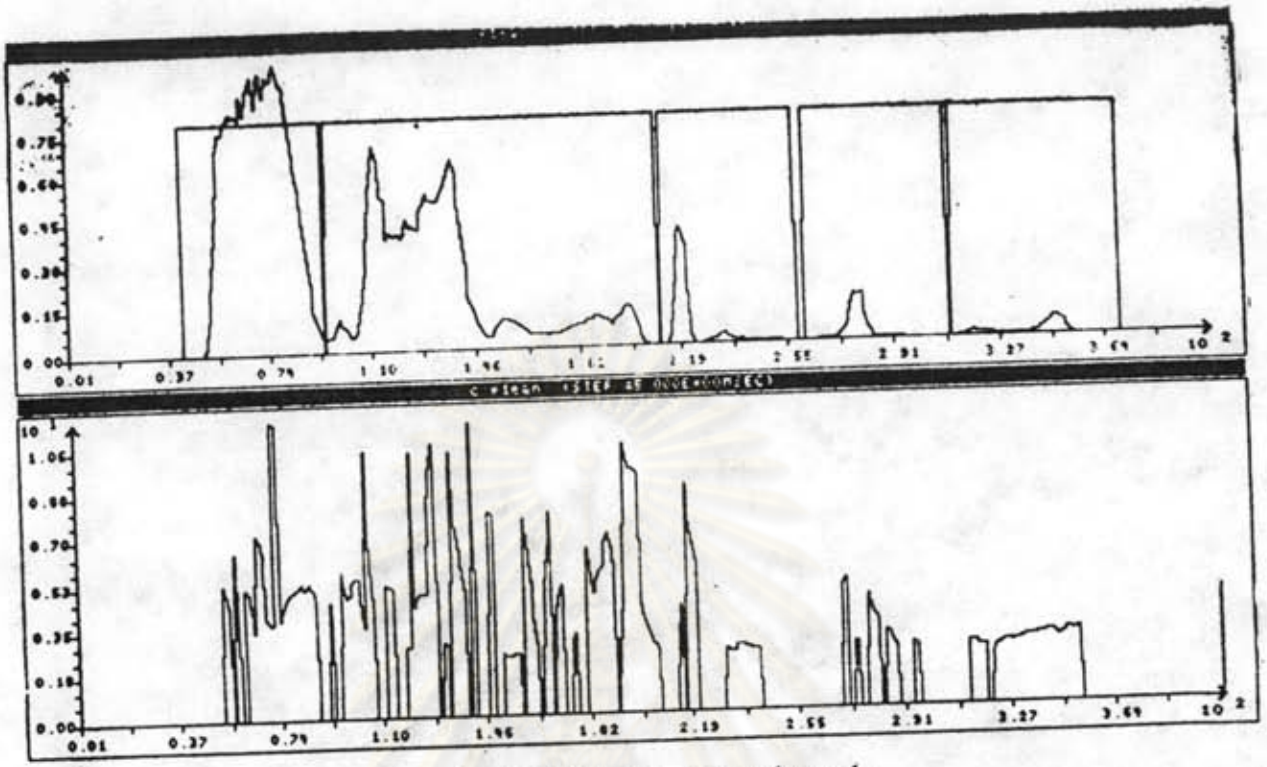


ประยุคที่ 29 "จุฬาลงกรณ์มหาวิทยาลัย"

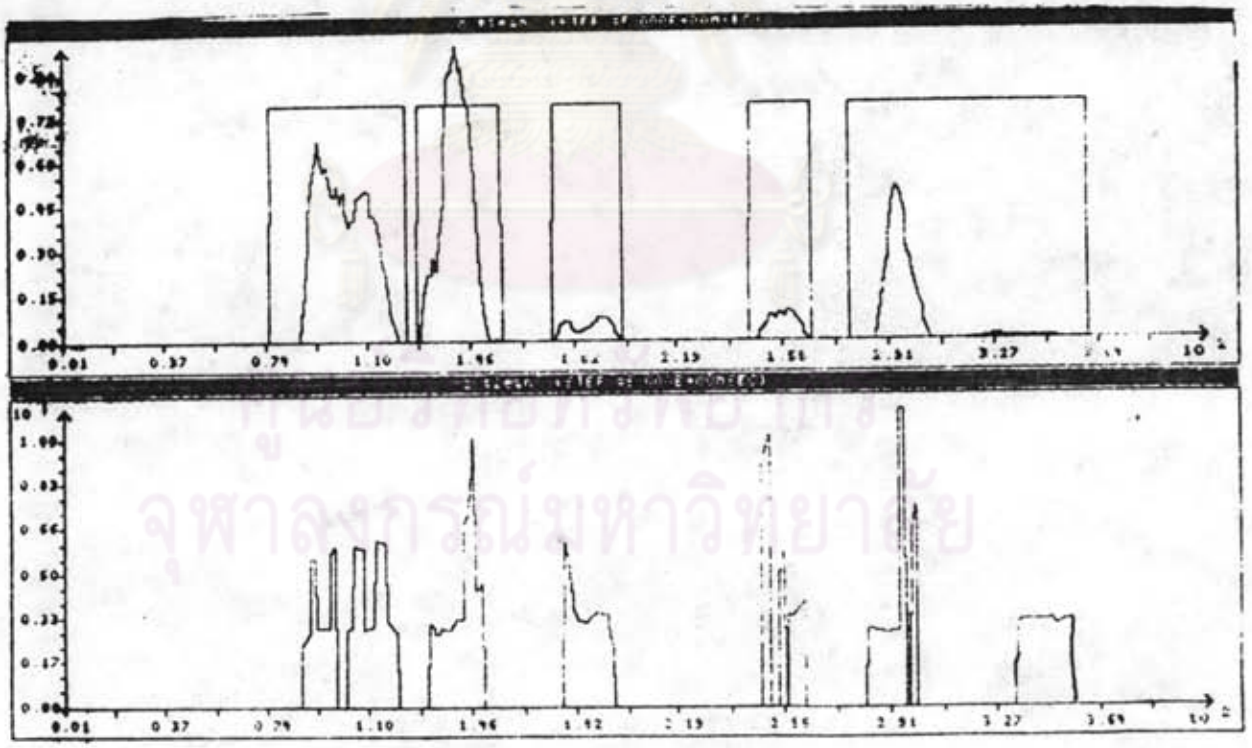


ประยุคที่ 30 "สถาบันเทคโนโลยีพระจอมเกล้า"

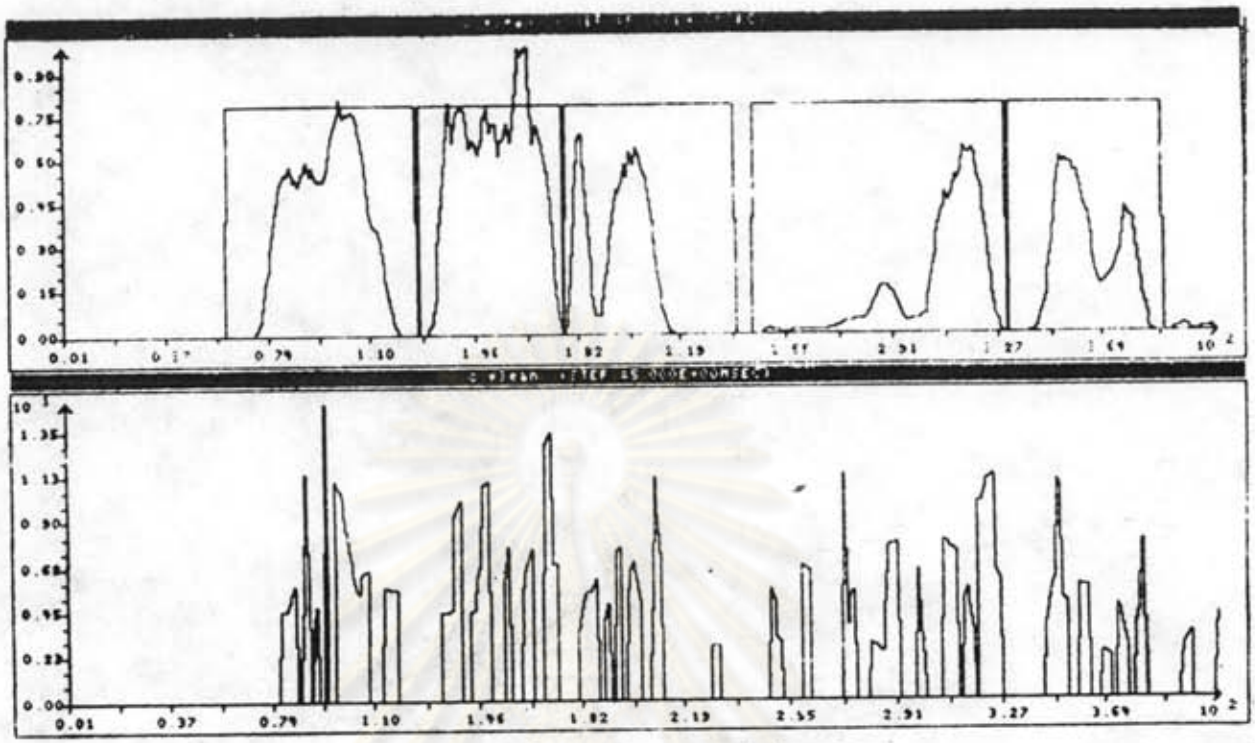
ตัวอย่างเสียงของ ฌ1. และ ฌ2.



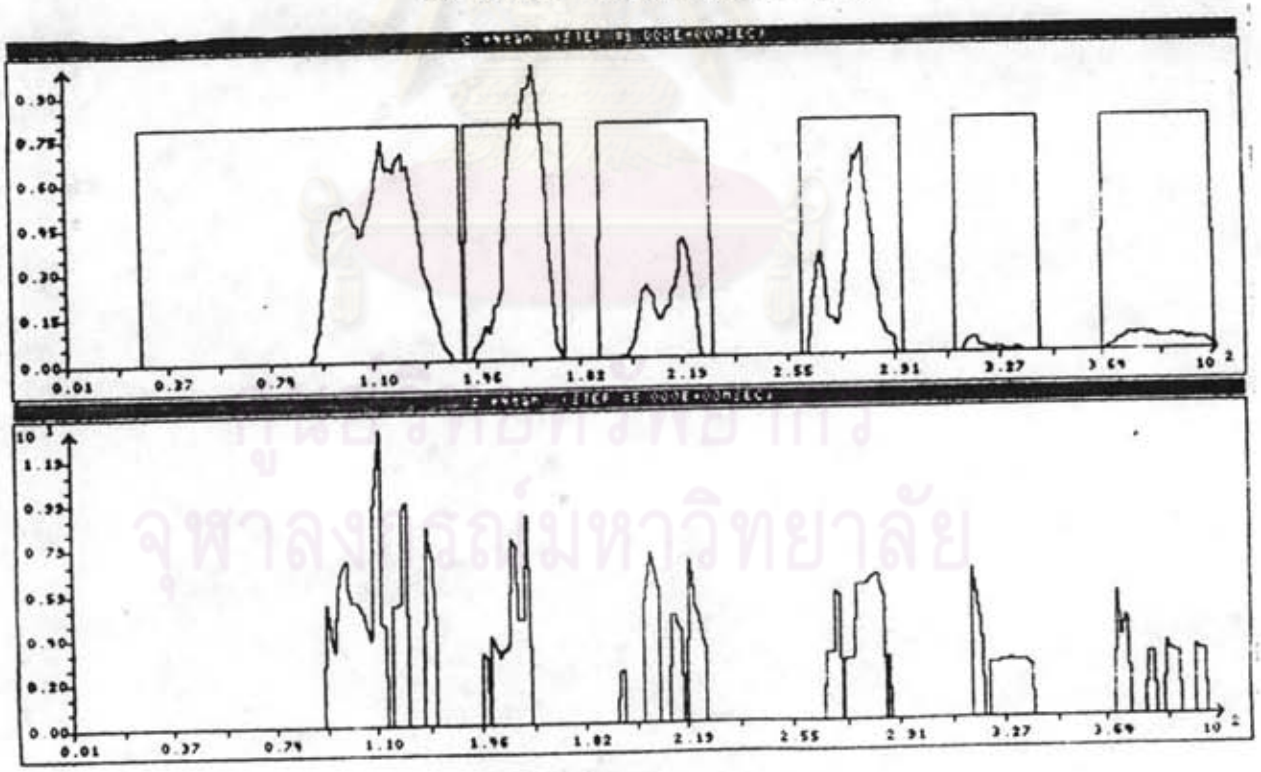
ประวัติที่ 1 "นาย ชาลี พงษ์สมบูรณ์"



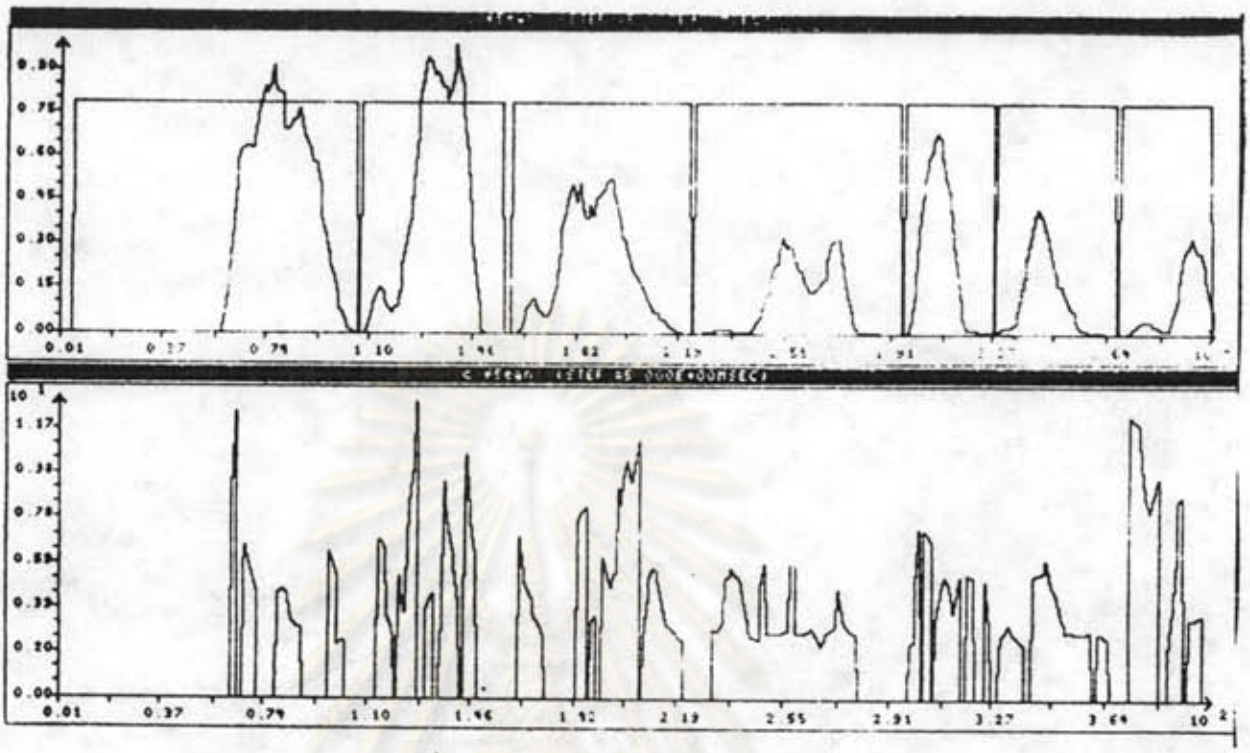
ประวัติที่ 2 "นาย ประดิษ จิตมั่นคง"



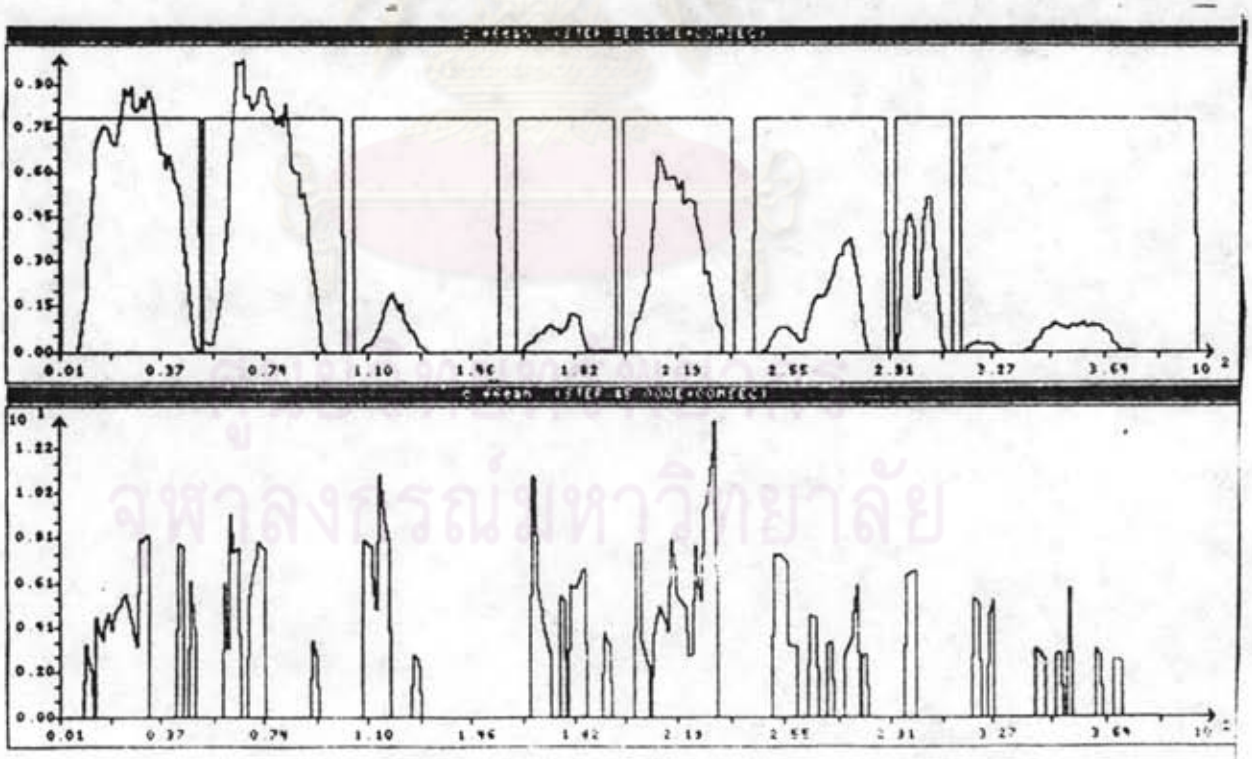
ประวัติที่ 3 "นาย แก้วพันธุ์ ธีระวาณิน"



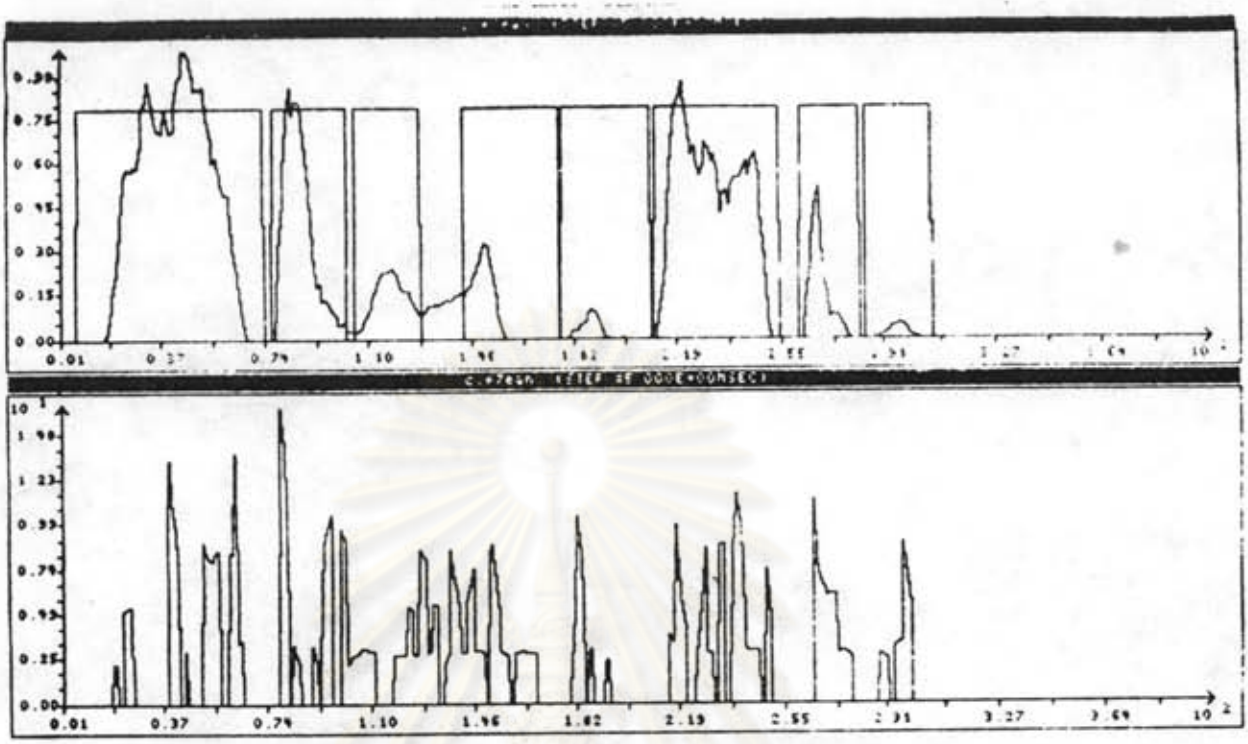
ประวัติที่ [illegible] "นาย [illegible] ทรี ประทุมทาน"



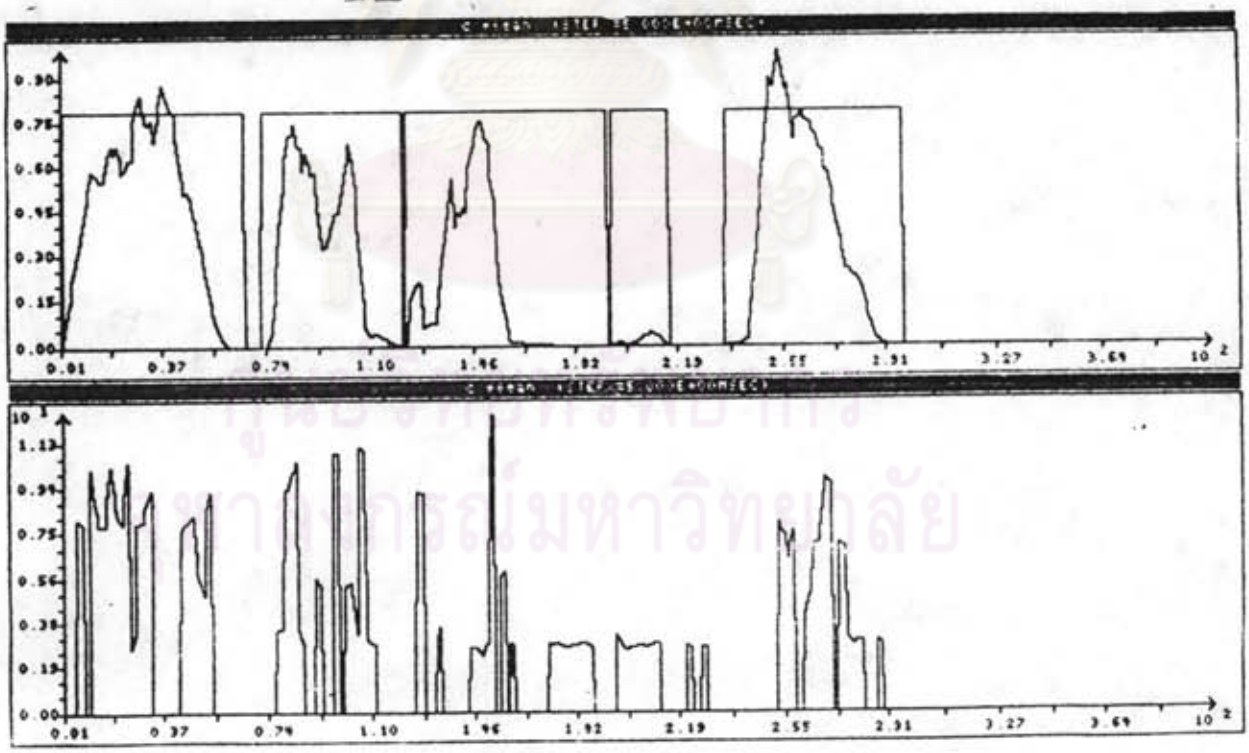
ประวัติที่ 5 "นาย เชิดชัย เสริมคารวงศ์สักดิ์"



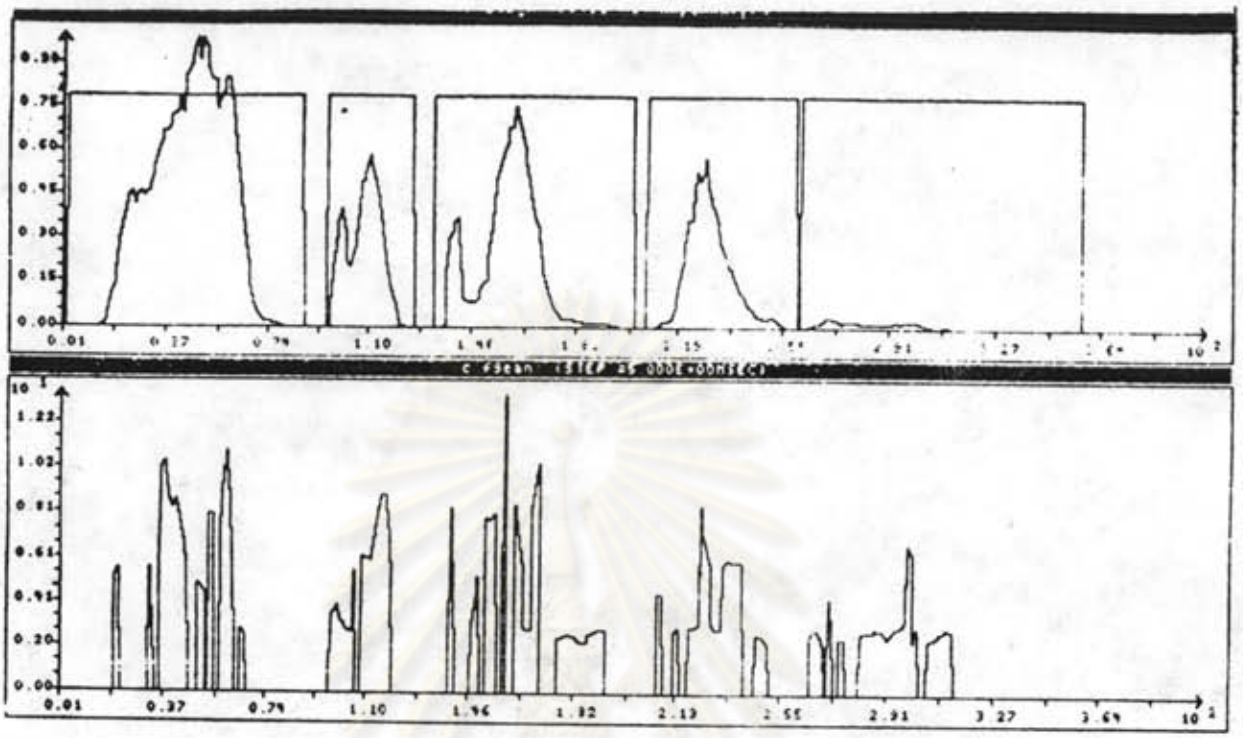
ประวัติที่ 6 "นาย อาจิน จิรชีพพัฒนา"



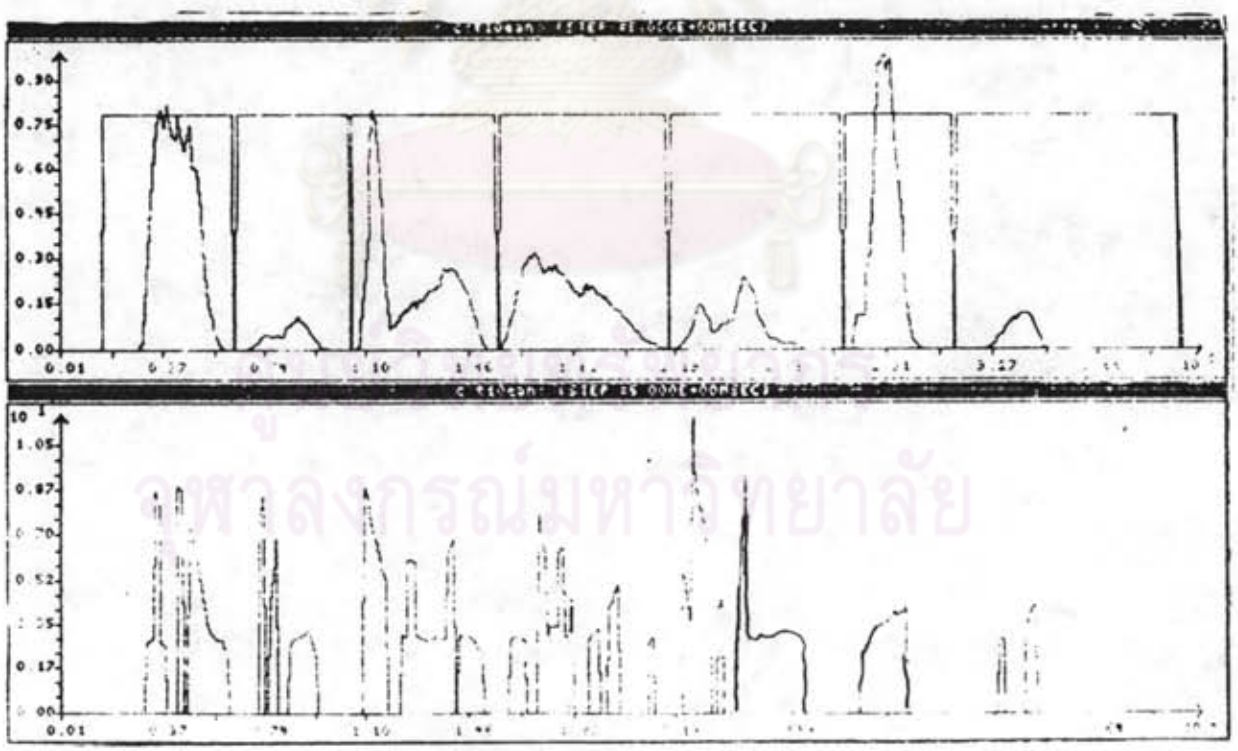
ประวัติที่ 7 "นาย ประยูร จินดาประดิษฐ์"



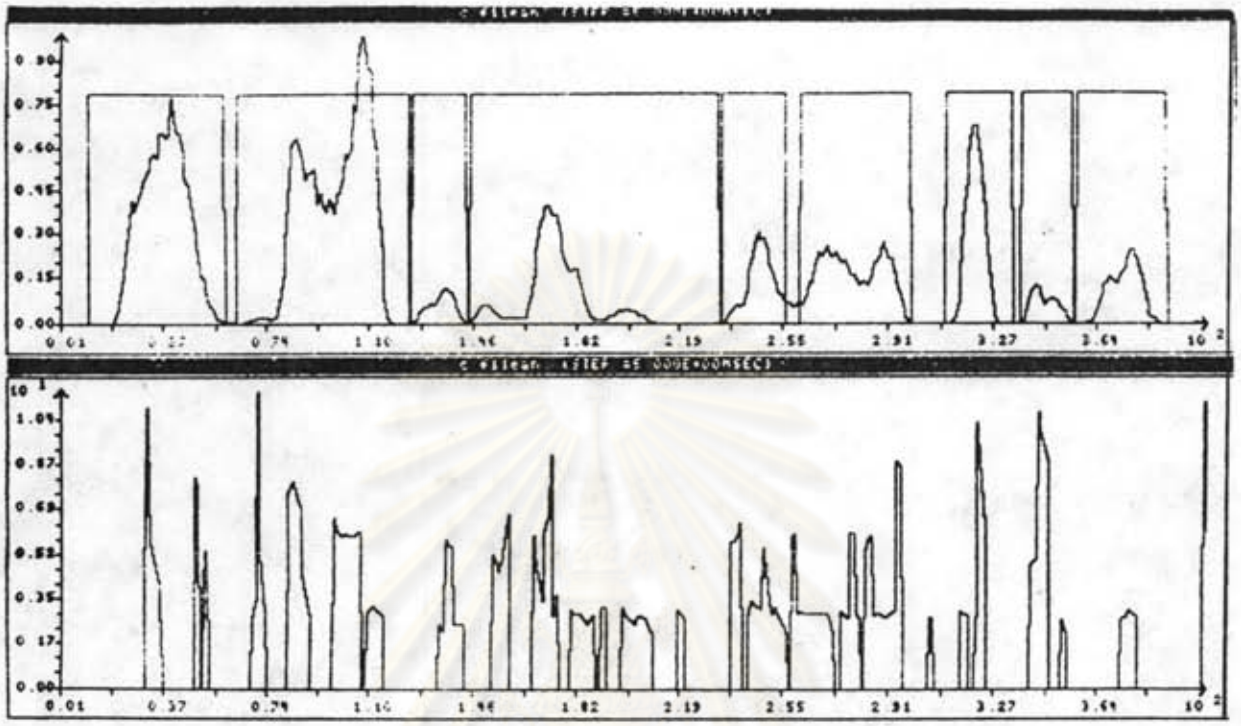
ประวัติที่ 8 "นาย เสริมศักดิ์ บุนนาค"



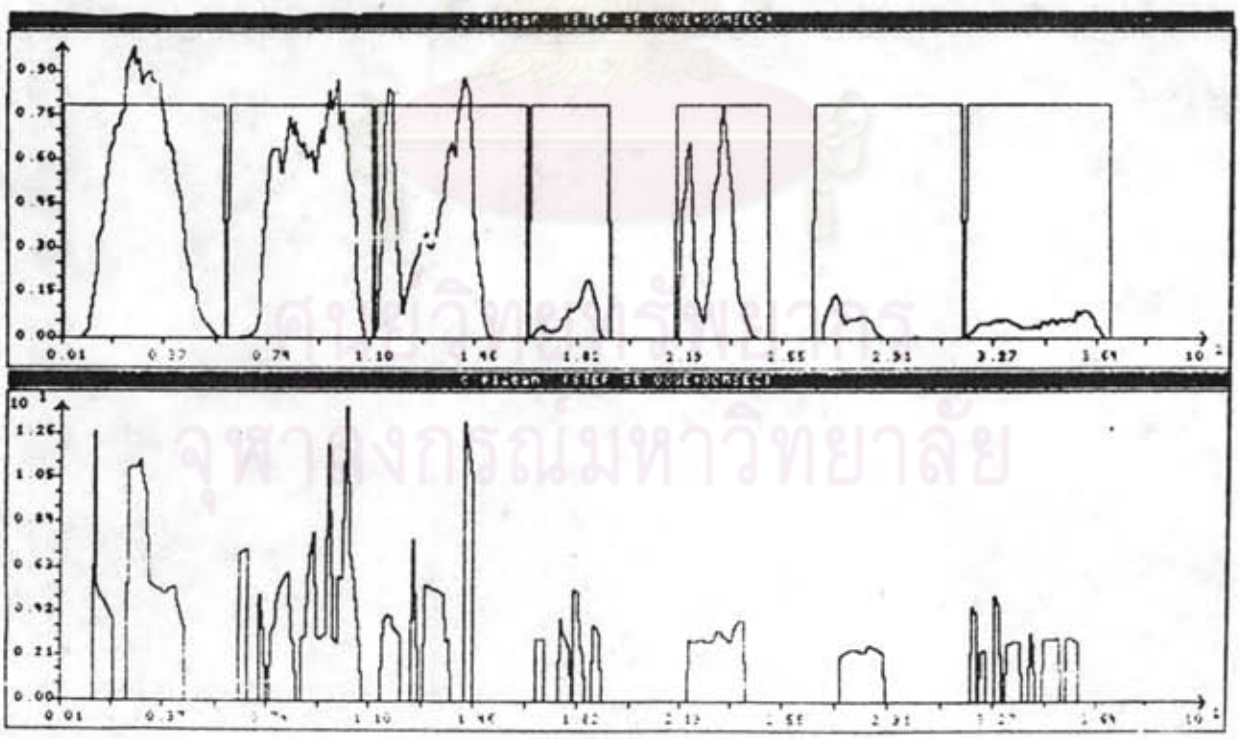
ประวัติที่ 9 "นาง ประไพ วงศ์"



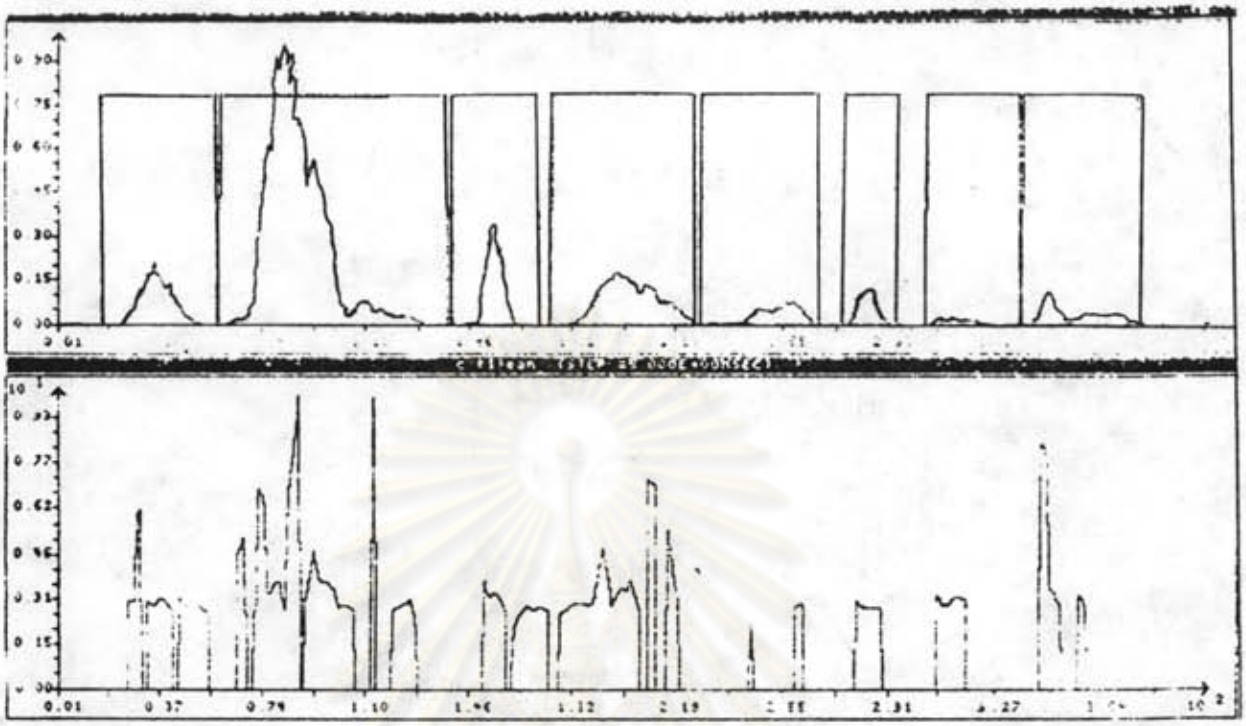
ประวัติที่ 10 "นาง สุชาดา วัชยประวัติ"



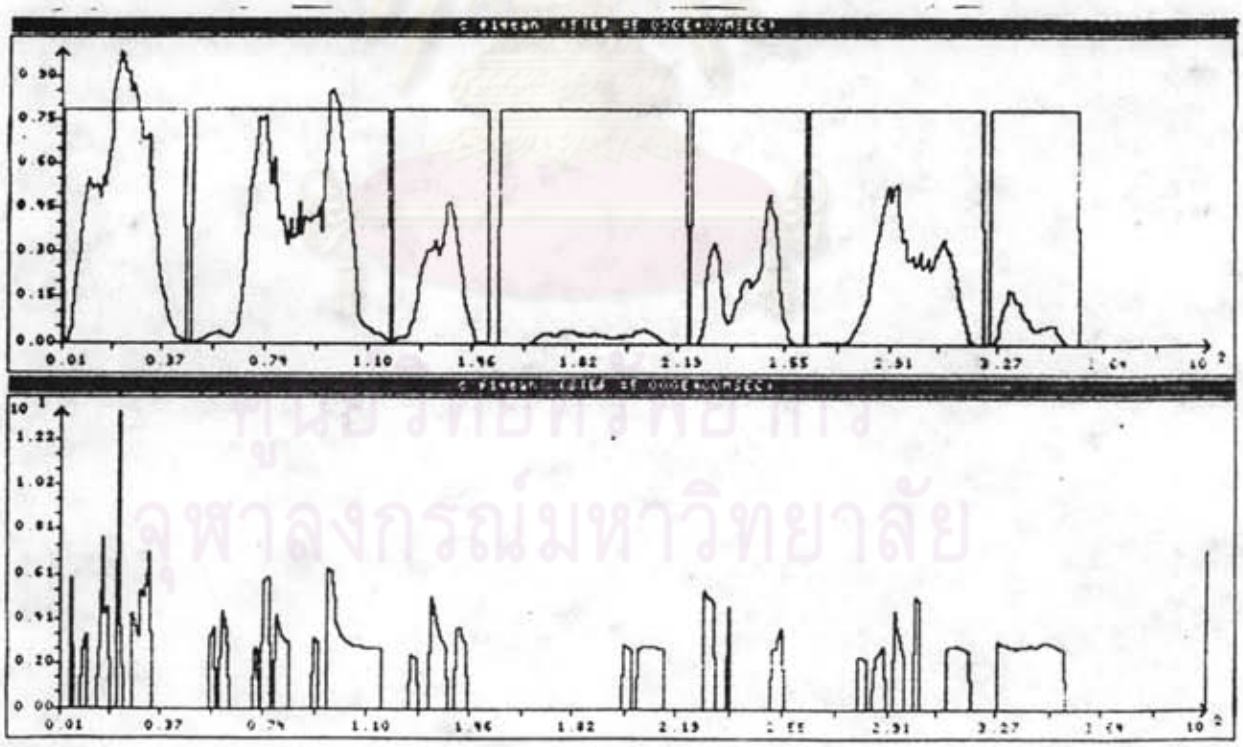
ประโยคที่ 11 "นางสาว บุศริน เจริญวัฒนกุลย์"



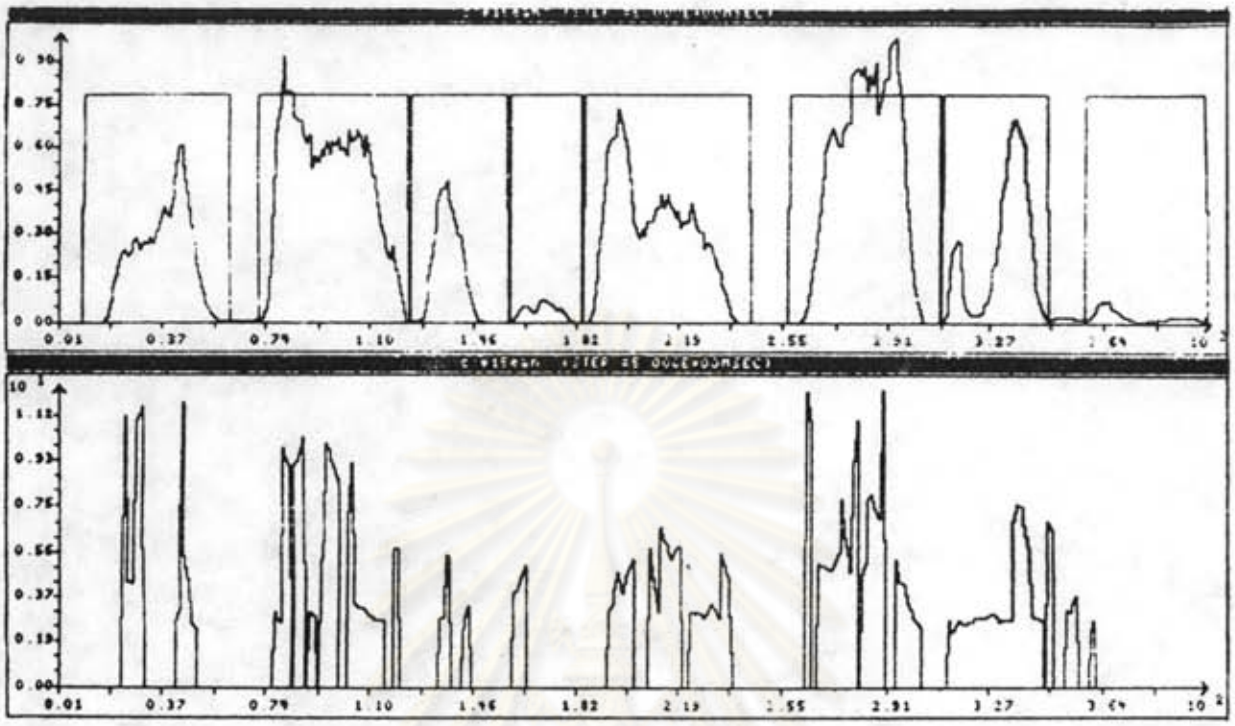
ประโยคที่ 12 "นางสาว พรทิพย์ ประทุมทาน"



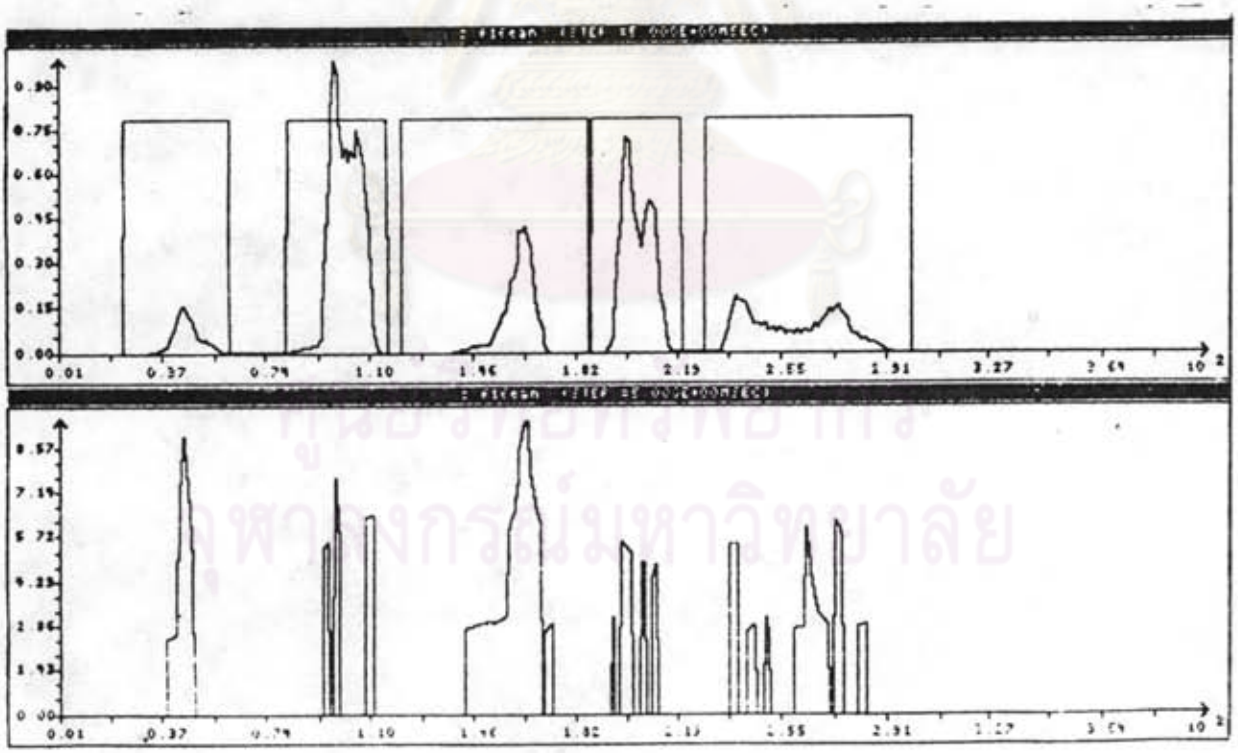
ประวัติที่ 13 "นางสาว สุกัญญา เสริมศารงค์ศักดิ์"



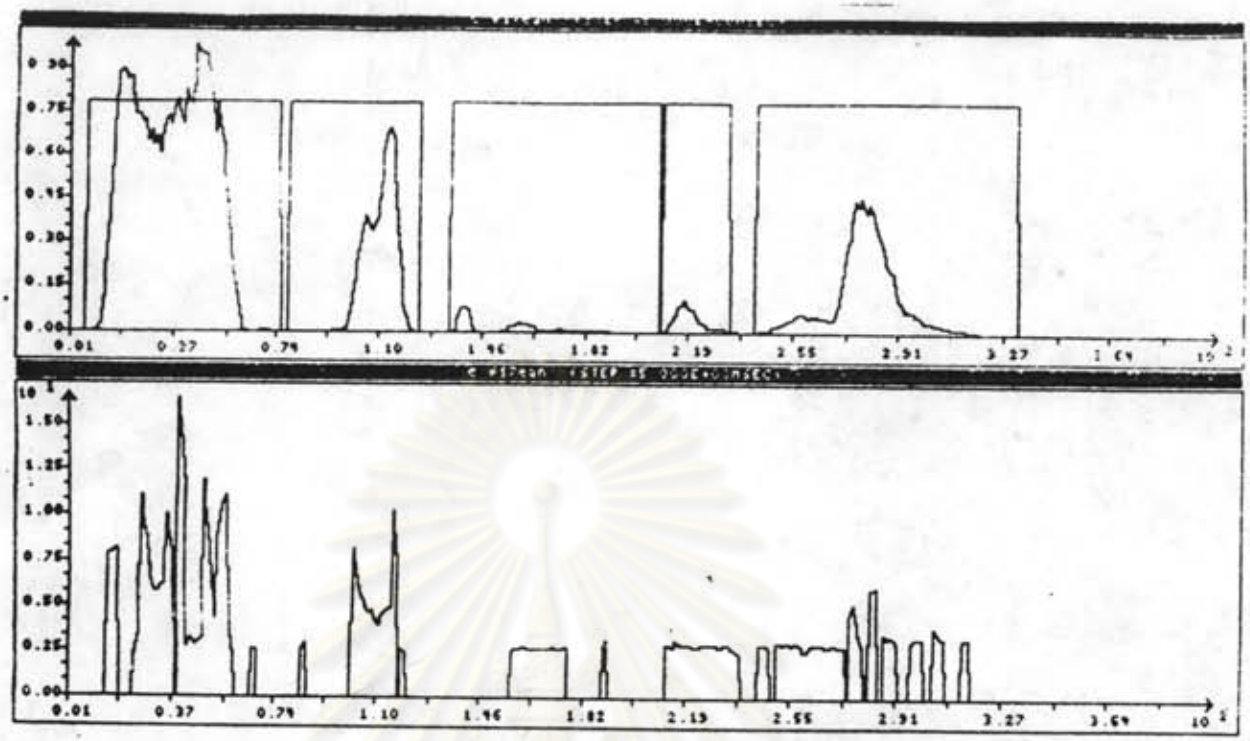
ประวัติที่ 14 "นางสาว มณี ชยานนท์"



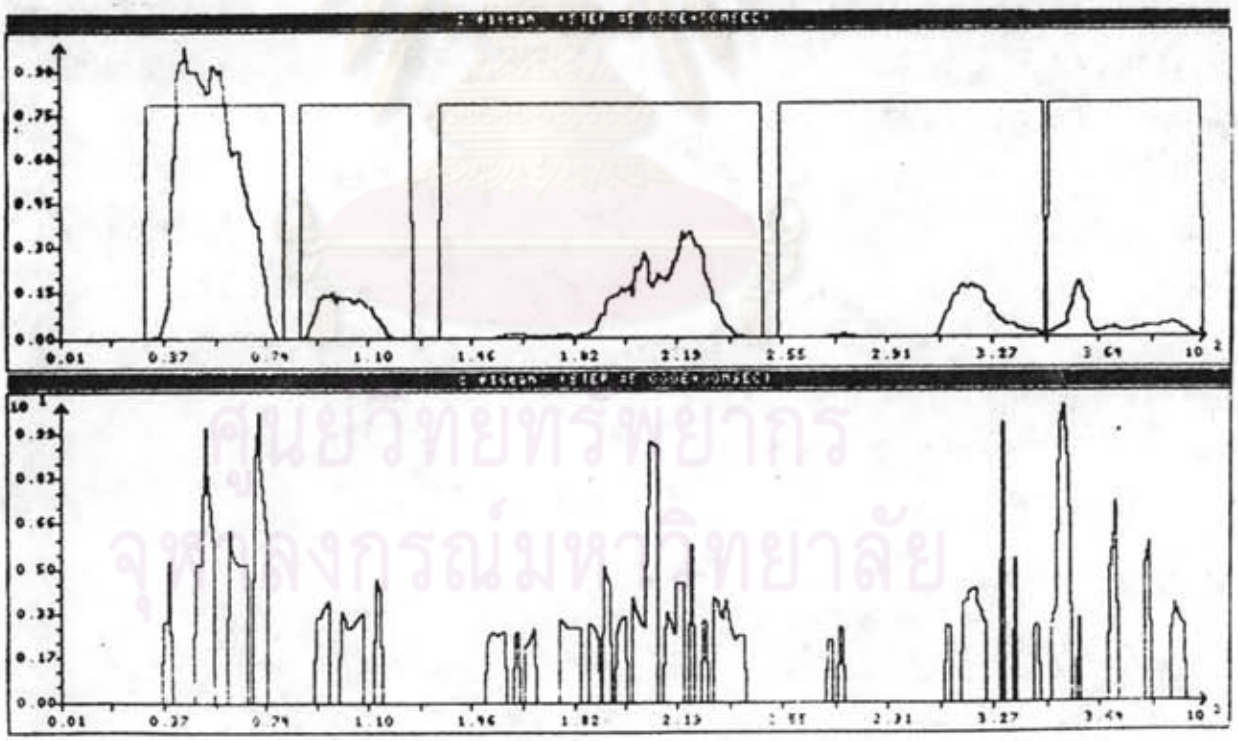
ประโยคที่ 15 "นางสาว นันทิดา แก้วบัวสาย"



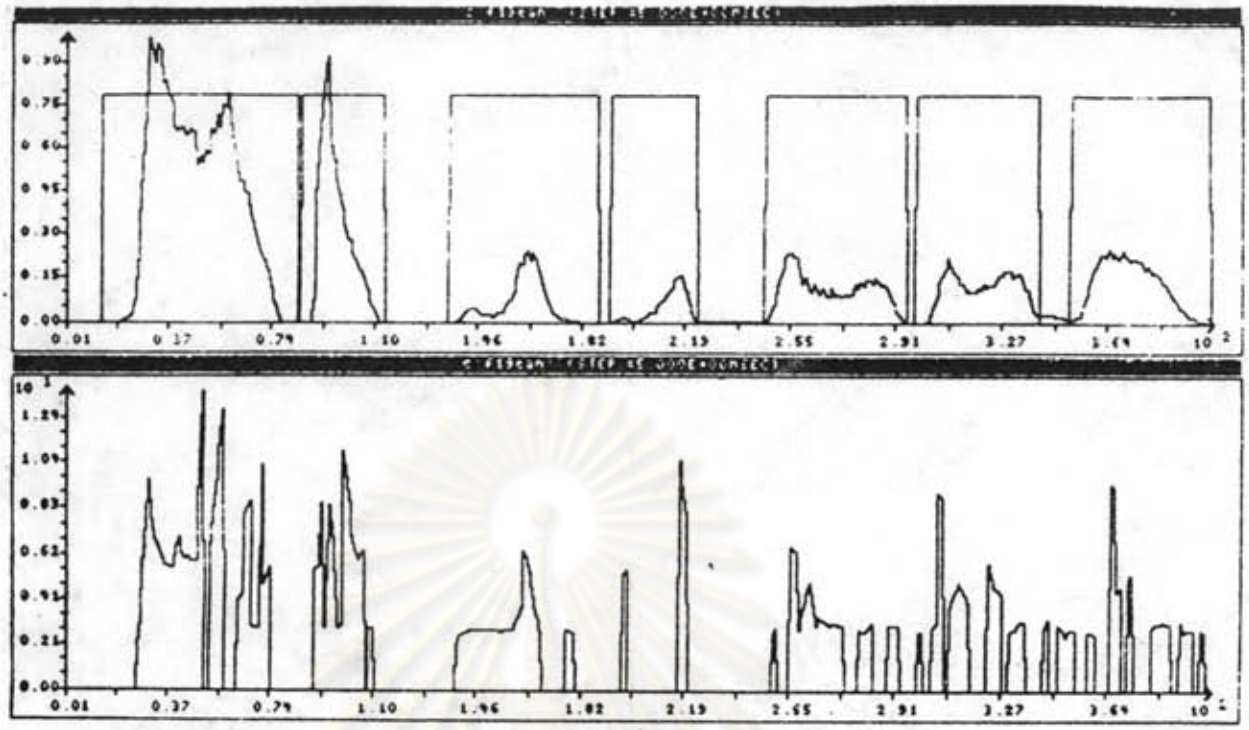
ประโยคที่ 16 "ลิงมักโหนต้นไม้"



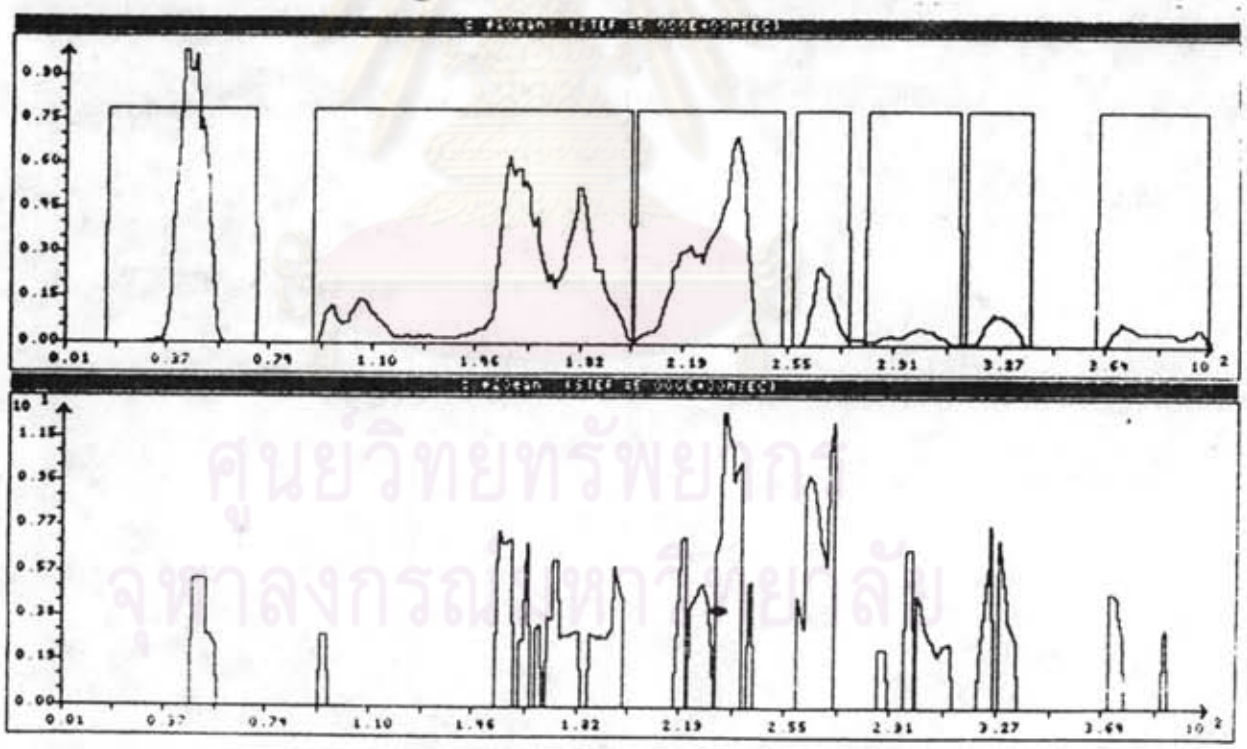
ประโยคที่ 17 "การควบคุมดูแล"



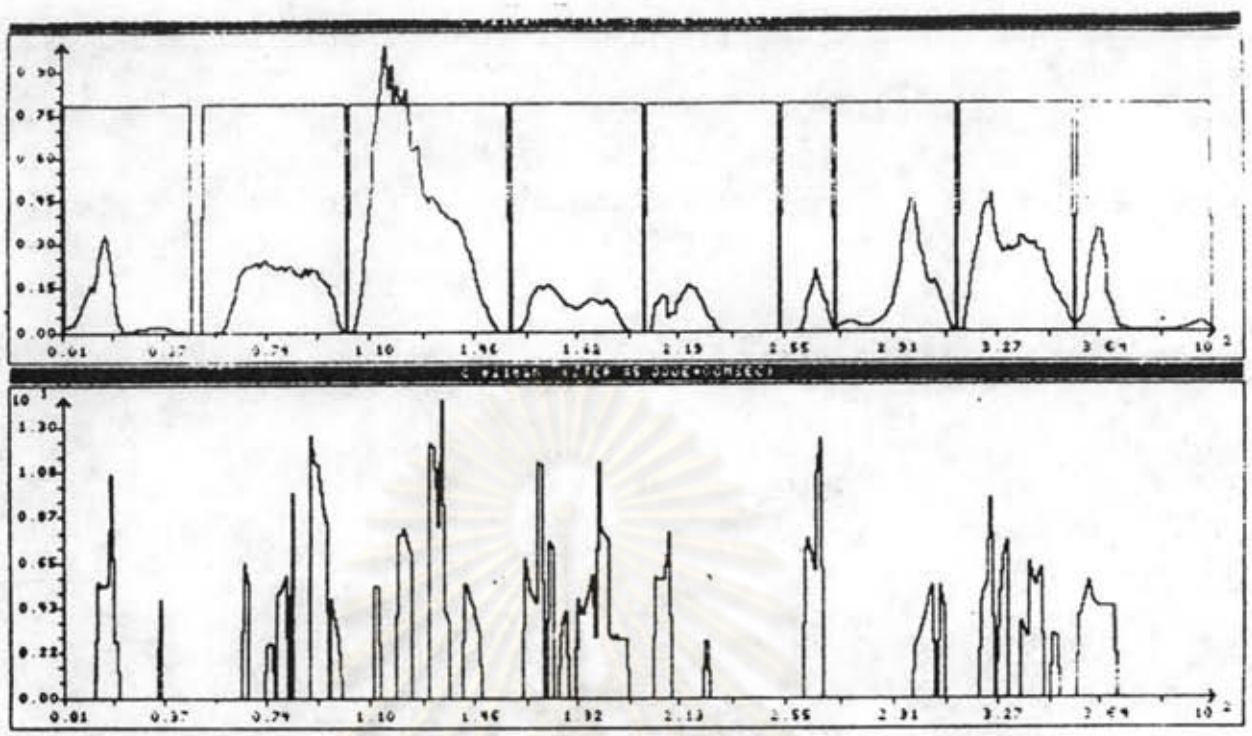
ประโยคที่ 18 "ถึงบอกคุณแล้ว คุณไม่เชื่อ"



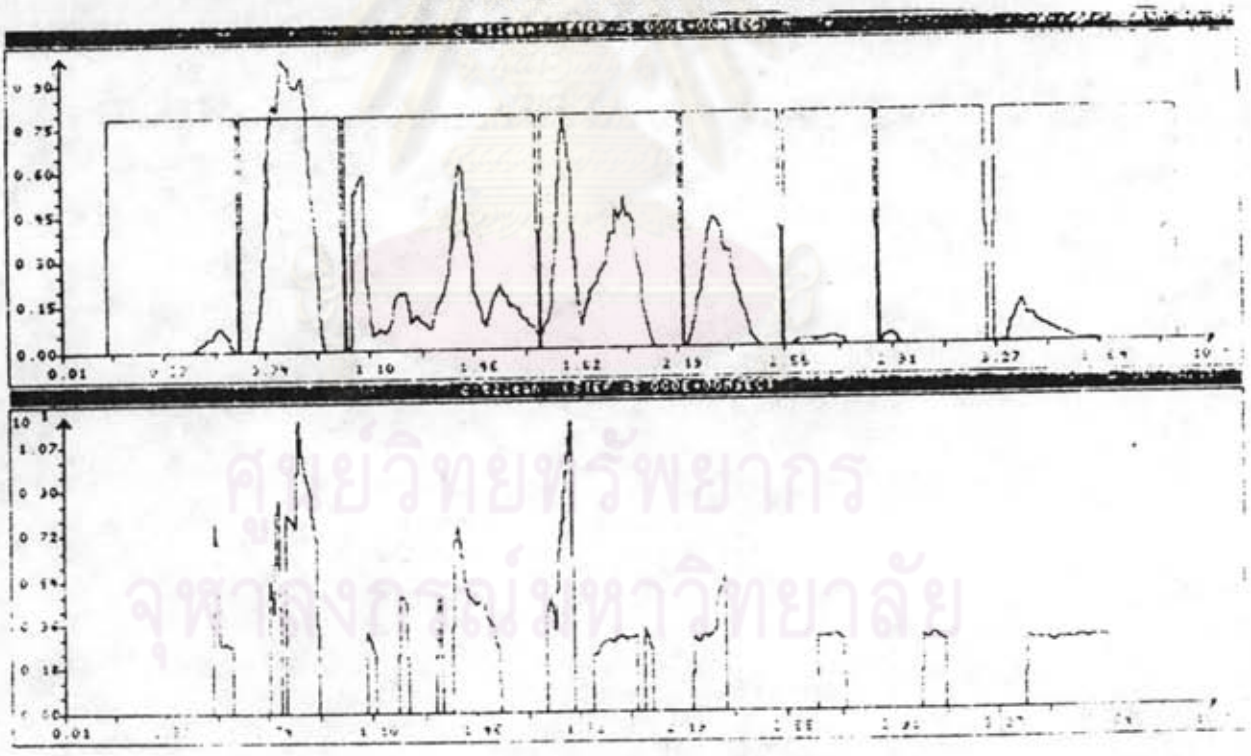
ประโยคที่ 19 "กลายเป็นเมืองขึ้นมานานแล้ว"



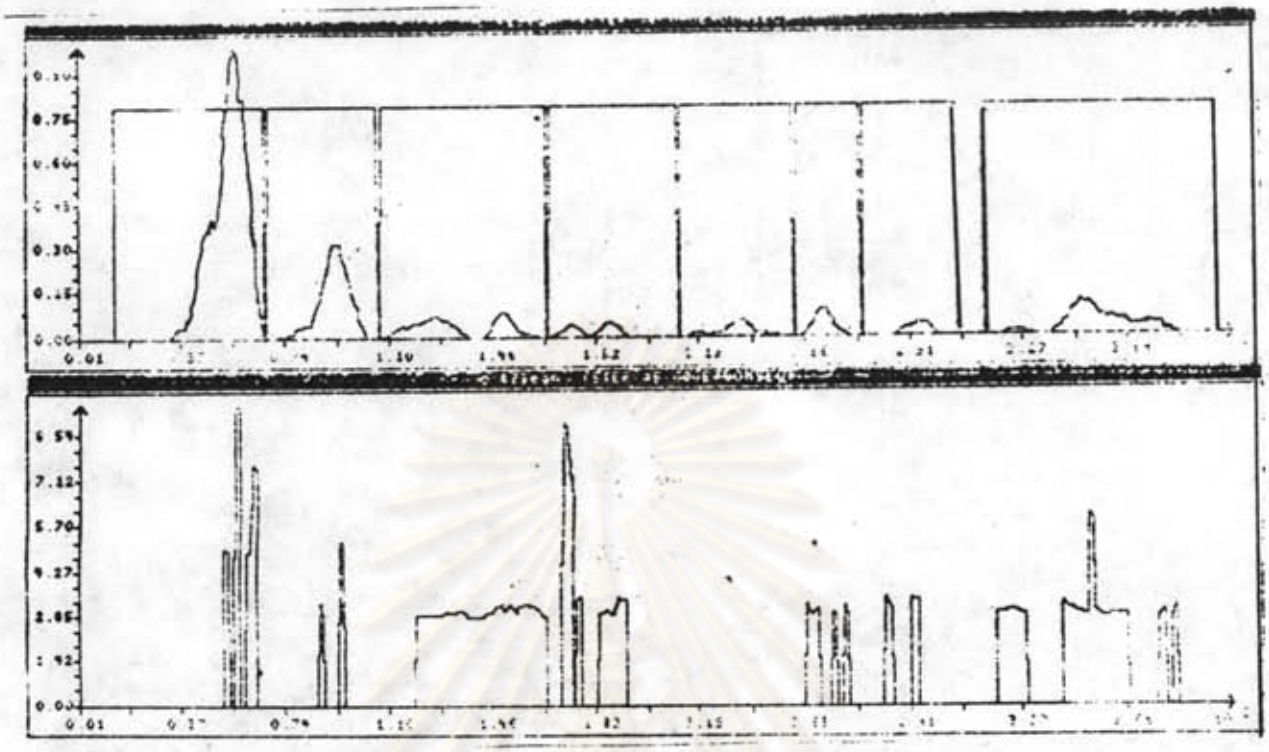
ประโยคที่ 20 "สัตว์ทั้งหลายรอดชีวิตได้"



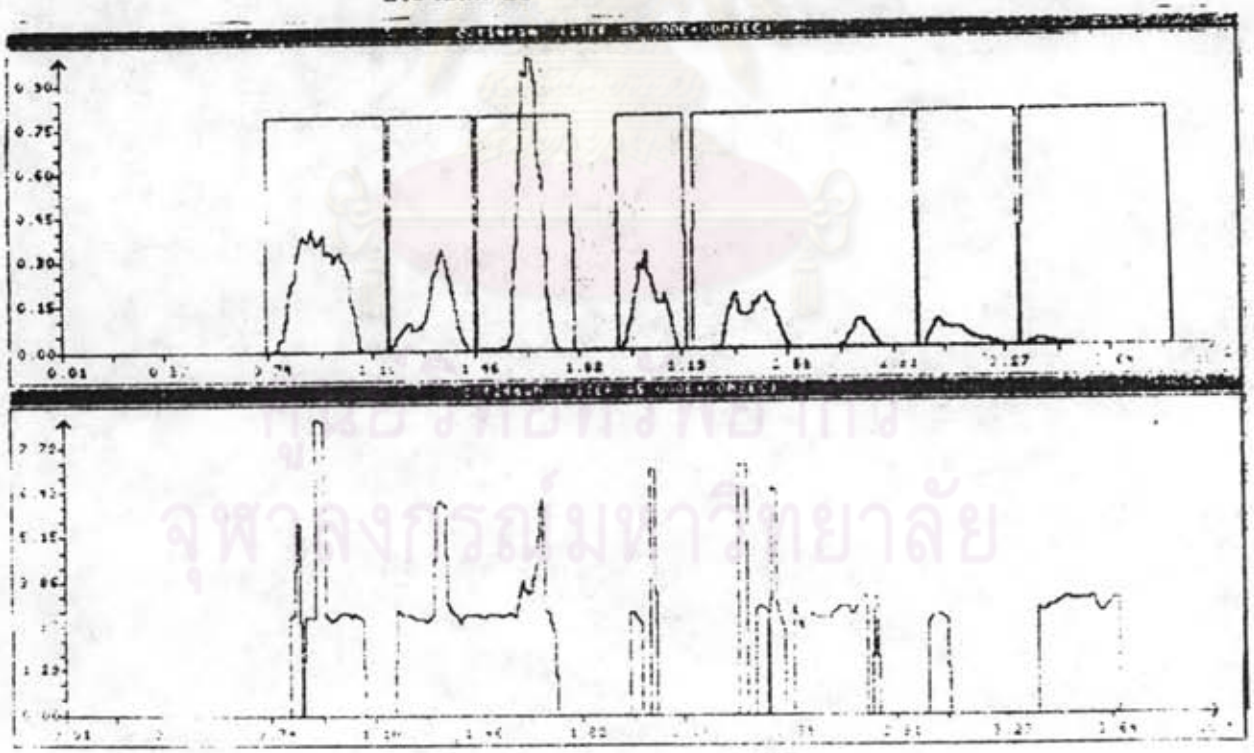
ประวัติที่ 21 "หาศัพท์ หาชั่วโมงชั่ว"



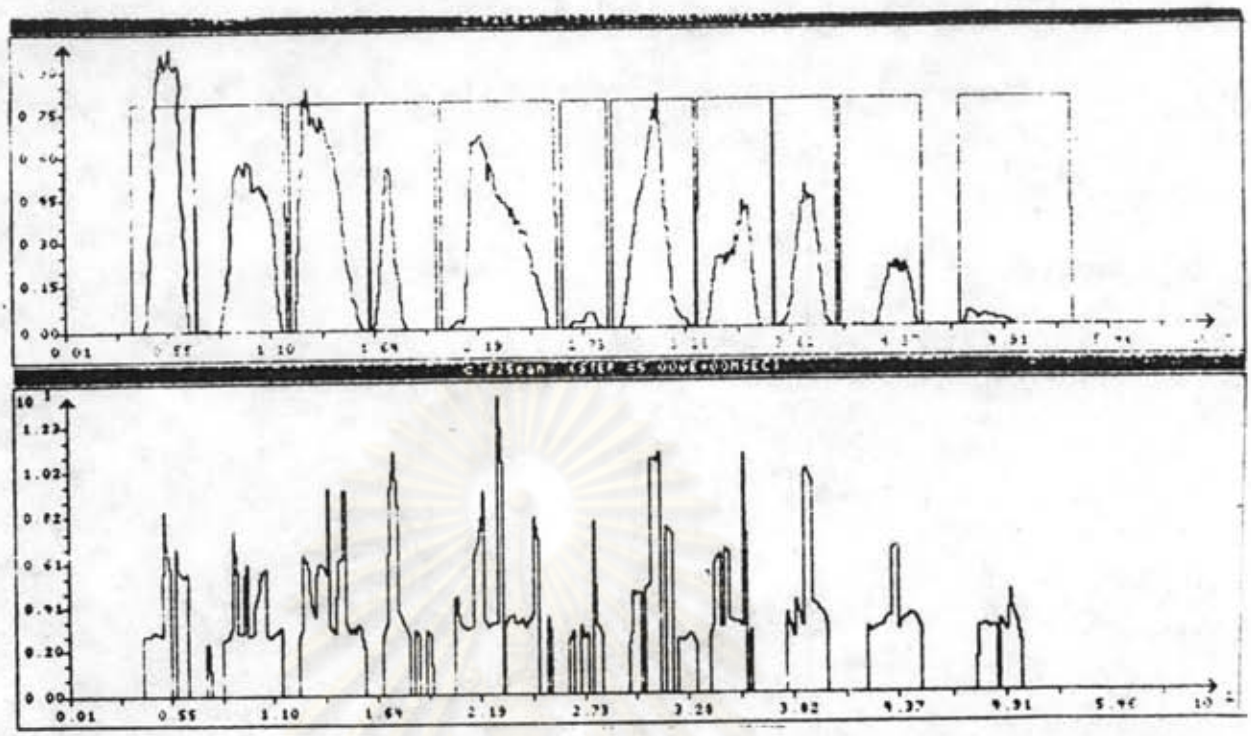
ประวัติที่ 22 "คณะผู้เชี่ยวชาญด้านคอมพิวเตอร์"



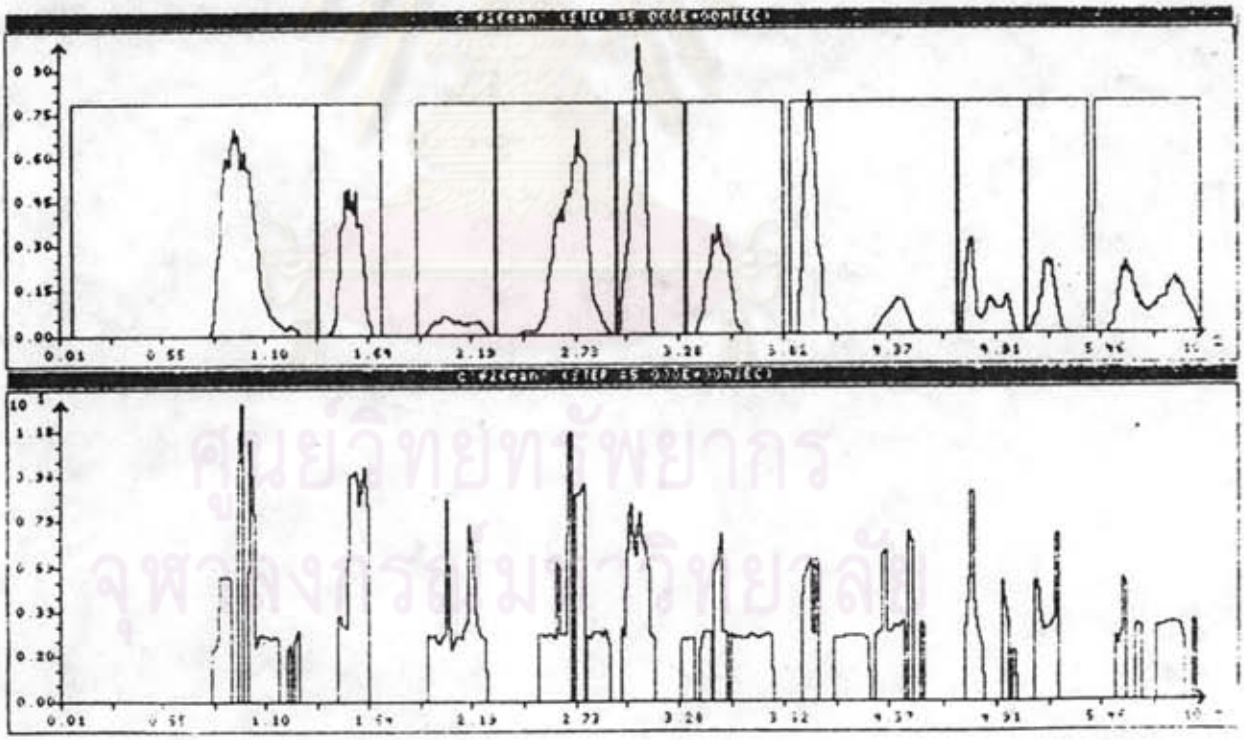
ประยุคที่ 23 "ตาคคะเนในสิ่งที่ยังหวังไม่ได้"



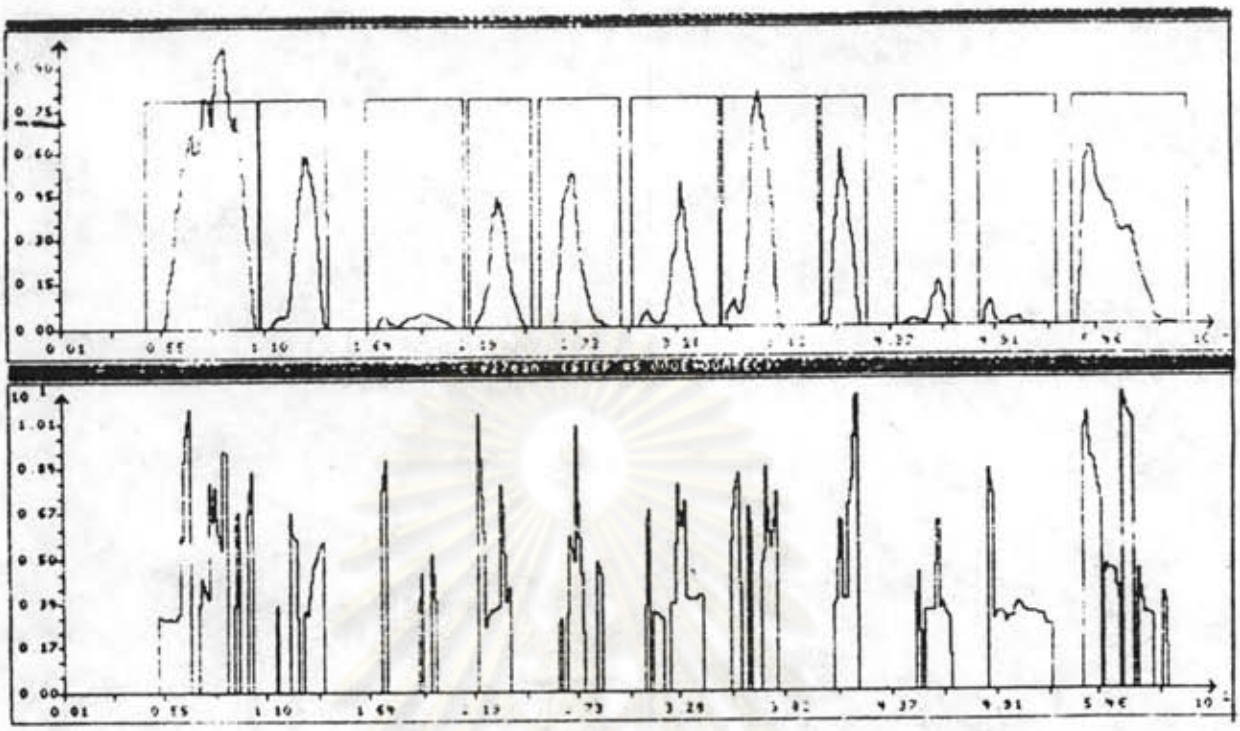
ประยุคที่ 24 "การเรียงลำดับคำยังไม่ดี"



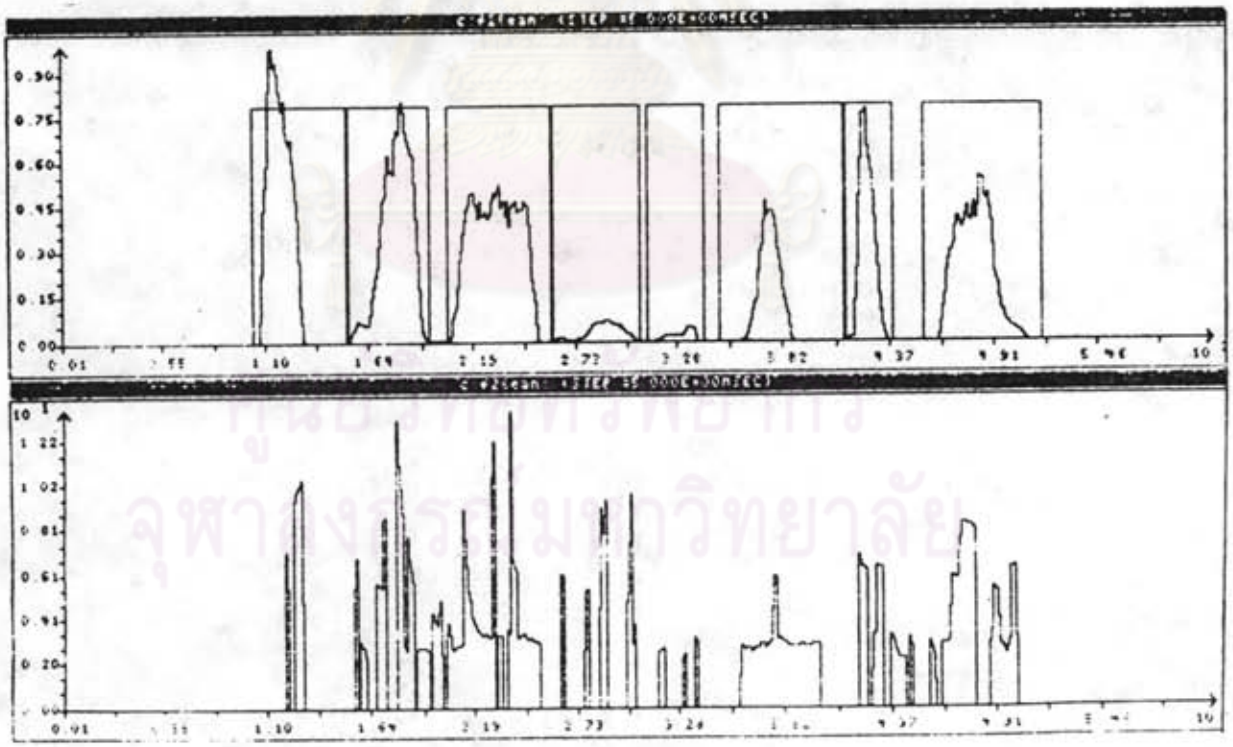
ประโยคที่ 25 "นายบายดังกล่าวมิได้มาจากความจริง"



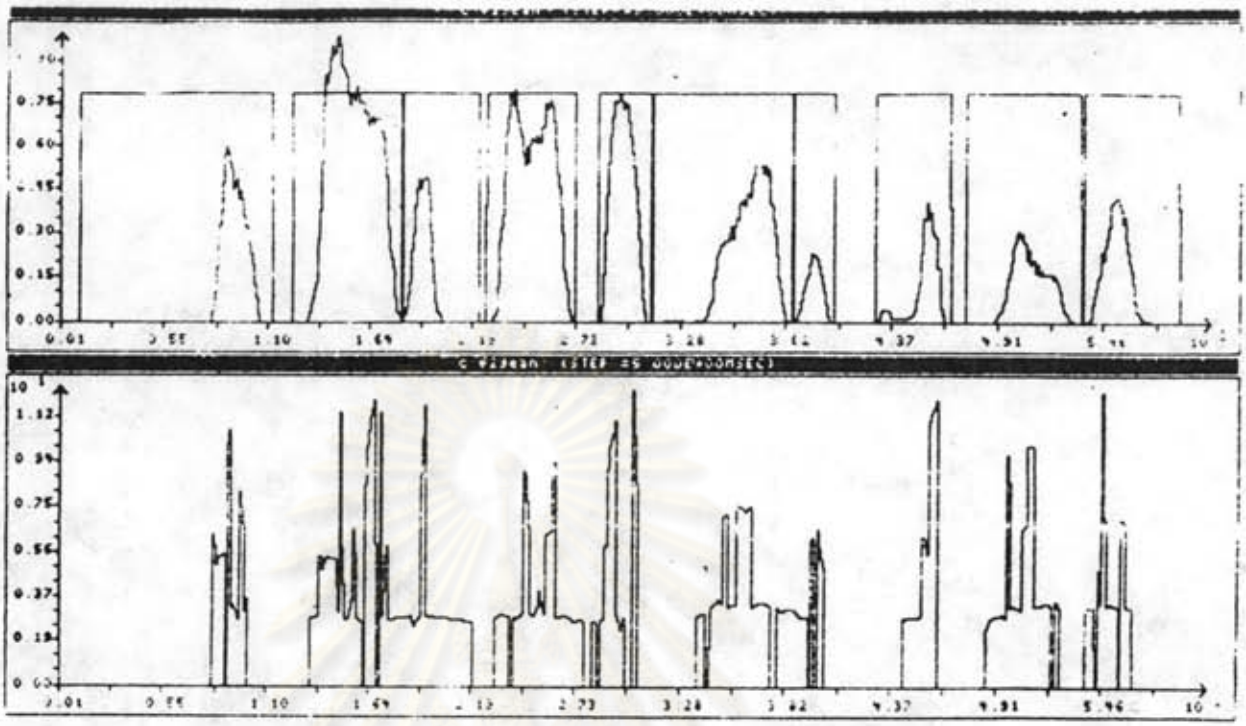
ประโยคที่ 26 "ในอดีตเคยดำรงตำแหน่งผู้จัดการ"



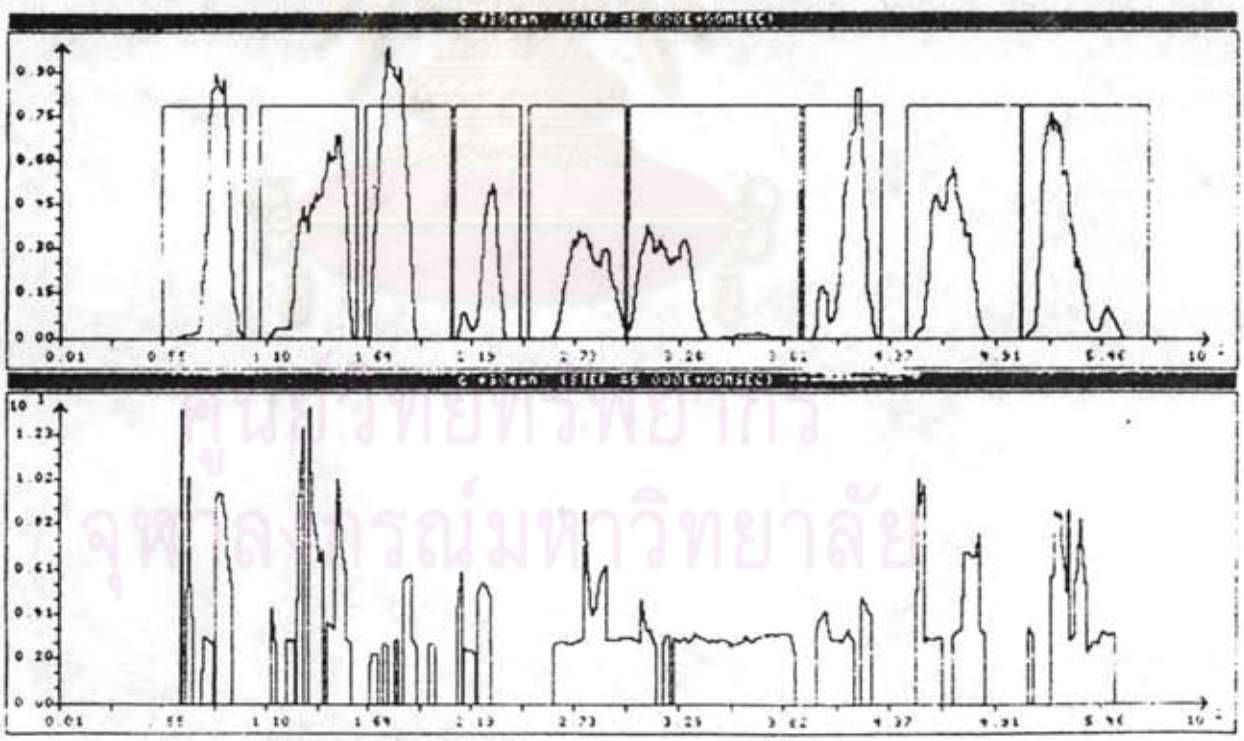
ประโยคที่ 27 "นิยมคนที่จะไม่เกี่ยวข้องกับคอมพิวเตอร์"



ประโยคที่ 28 "เป็นภาษาที่นิยมกันมาก"



ประจำคที่ 29 "จุฬาลงกรณ์มหาวิทยาลัย"



ประจำคที่ 30 "สถาบันเทคโนโลยีพระจอมเกล้า"

ภาคผนวก จ. ประโยคตัวอย่างที่ใช้ในการคัดพยางค์และตรวจรู้

ประโยคที่ใช้ในการทดลอง เพื่อคัดพยางค์และจดจำ เป็นประโยคเกี่ยวกับตัวเลข 0 ถึง 9 ดังต่อไปนี้

ประโยคที่ 1 "1578"

ประโยคที่ 2 "3629"

ประโยคที่ 3 "1791"

ประโยคที่ 4 "0741"

ประโยคที่ 5 "6550"

ประโยคที่ 6 "5667"

ประโยคที่ 7 "3028"

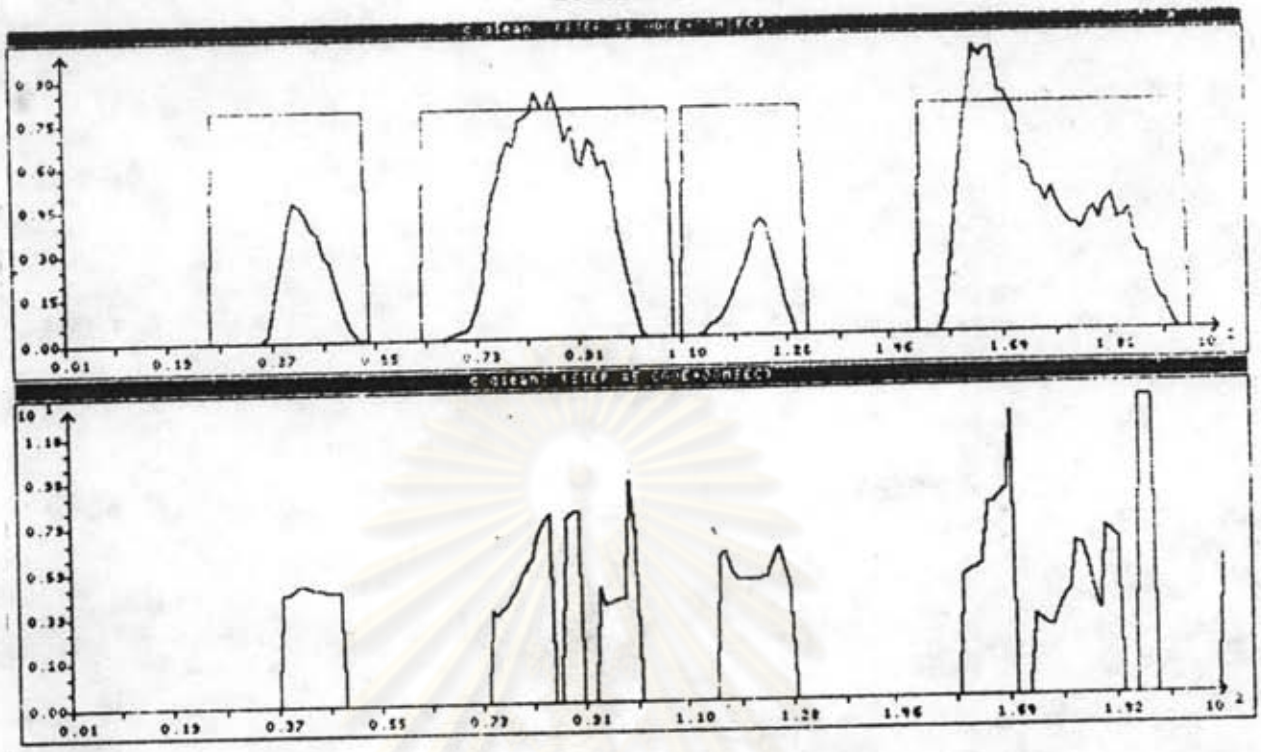
ประโยคที่ 8 "3201"

ภาคผนวก ฉ. ผลการศึกษาง่ายและตรวจรู้

ในแต่ละประโยค รูปบนคือ ค่าพลังงาน R

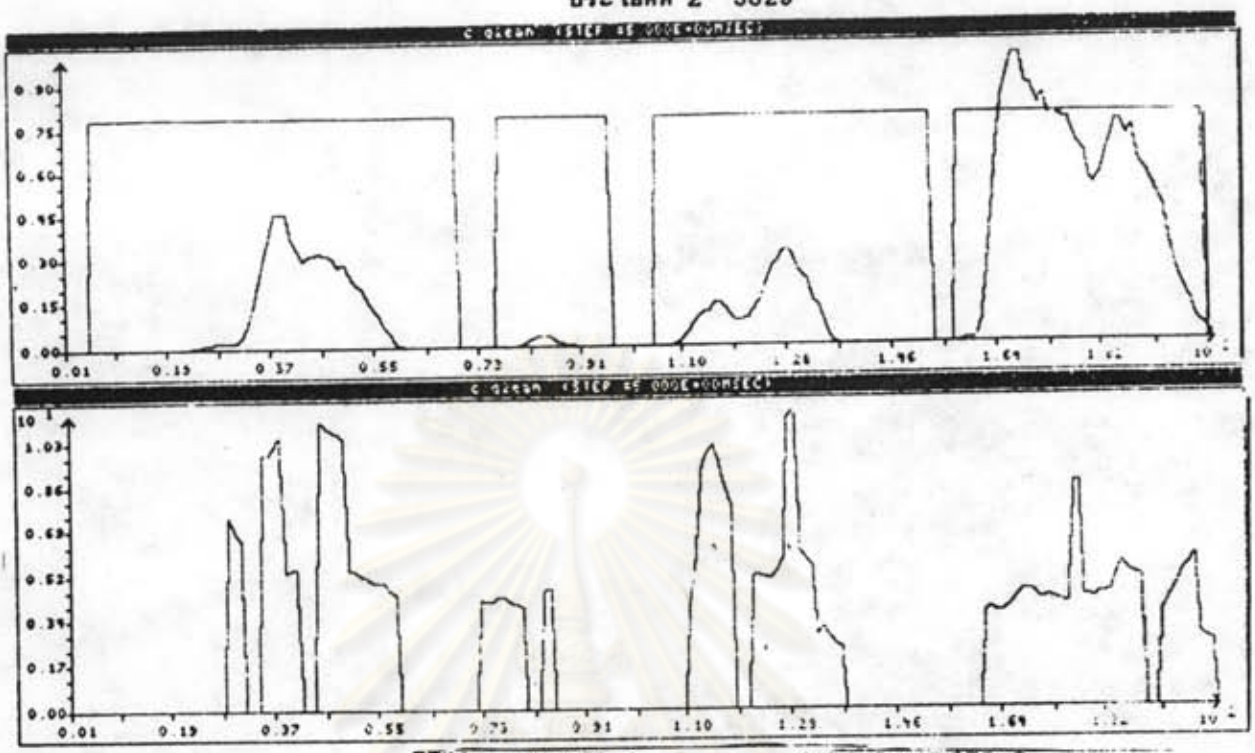
รูปล่างคือ Pitch Period

ประโยคที่ 1 "1578"



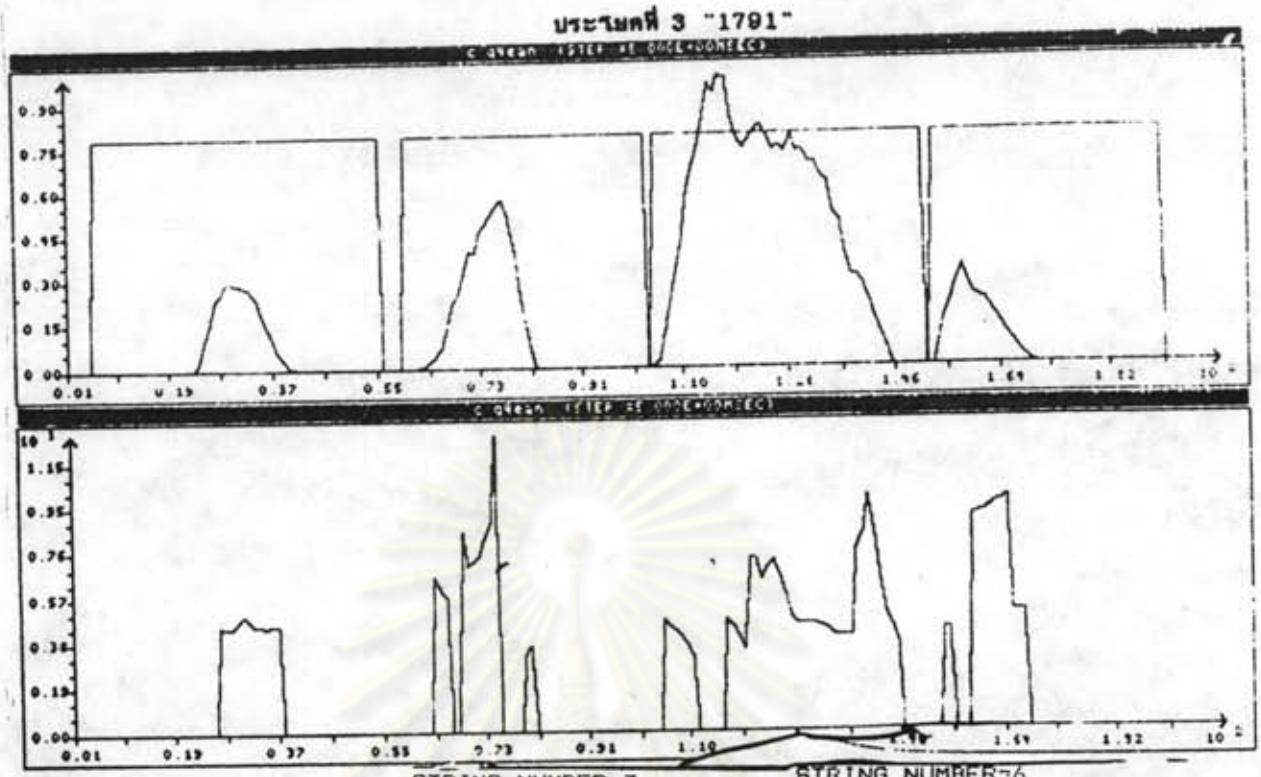
POSITION1=27
POSITION2=64
STRING NUMBER=1
PATH VALUE= 1.5277165925
STRING NUMBER=2
PATH VALUE= 2.2807559178
STRING NUMBER=3
PATH VALUE= 2.1057240092
STRING NUMBER=4
PATH VALUE= 2.0244717863
STRING NUMBER=5
PATH VALUE= 2.0997839689
STRING NUMBER=6
PATH VALUE= 2.5842010396
STRING NUMBER=7
PATH VALUE= 2.0465956246
STRING NUMBER=8
PATH VALUE= 2.1308186763
STRING NUMBER=9
PATH VALUE= 2.2161949090
STRING NUMBER=10
PATH VALUE= 2.3166670301
WORD=1
POSITION1=64
POSITION2=108
STRING NUMBER=1
PATH VALUE= 2.1772579801
STRING NUMBER=2
PATH VALUE= 2.7008306052
STRING NUMBER=3
PATH VALUE= 2.2417901959
STRING NUMBER=4
PATH VALUE= 2.3347757282
STRING NUMBER=5
PATH VALUE= 1.8000062647
STRING NUMBER=6
PATH VALUE= 2.4941943420
STRING NUMBER=7
PATH VALUE= 2.0563311370
STRING NUMBER=8
PATH VALUE= 1.8530619742
STRING NUMBER=9
PATH VALUE= 2.3326691076
STRING NUMBER=10
PATH VALUE= 2.5345944332
WORD=5
POSITION1=110
POSITION2=151
STRING NUMBER=1
PATH VALUE= 2.1997062973
STRING NUMBER=2
PATH VALUE= 2.1683169093
STRING NUMBER=3
PATH VALUE= 2.4585743862
STRING NUMBER=4
PATH VALUE= 1.8975235867
STRING NUMBER=5
PATH VALUE= 2.3801919220
STRING NUMBER=6
PATH VALUE= 1.9834089312
STRING NUMBER=7
PATH VALUE= 1.4970904440
STRING NUMBER=8
PATH VALUE= 1.9555575347
STRING NUMBER=9
PATH VALUE= 2.5630831646
STRING NUMBER=10
PATH VALUE= 2.4305632106
WORD=7
POSITION1=151
POSITION2=195
STRING NUMBER=1
PATH VALUE= 2.2929729455
STRING NUMBER=2
PATH VALUE= 2.5316574441
STRING NUMBER=3
PATH VALUE= 1.8528327758
STRING NUMBER=4
PATH VALUE= 2.5016974751
STRING NUMBER=5
PATH VALUE= 2.2097076639
STRING NUMBER=6
PATH VALUE= 2.1760513399
STRING NUMBER=7
PATH VALUE= 2.1749906819
STRING NUMBER=8
PATH VALUE= 1.4483905408
STRING NUMBER=9
PATH VALUE= 2.0637618578
STRING NUMBER=10
PATH VALUE= 2.6761354545
WORD=8

ประโยคที่ 2 "3629"



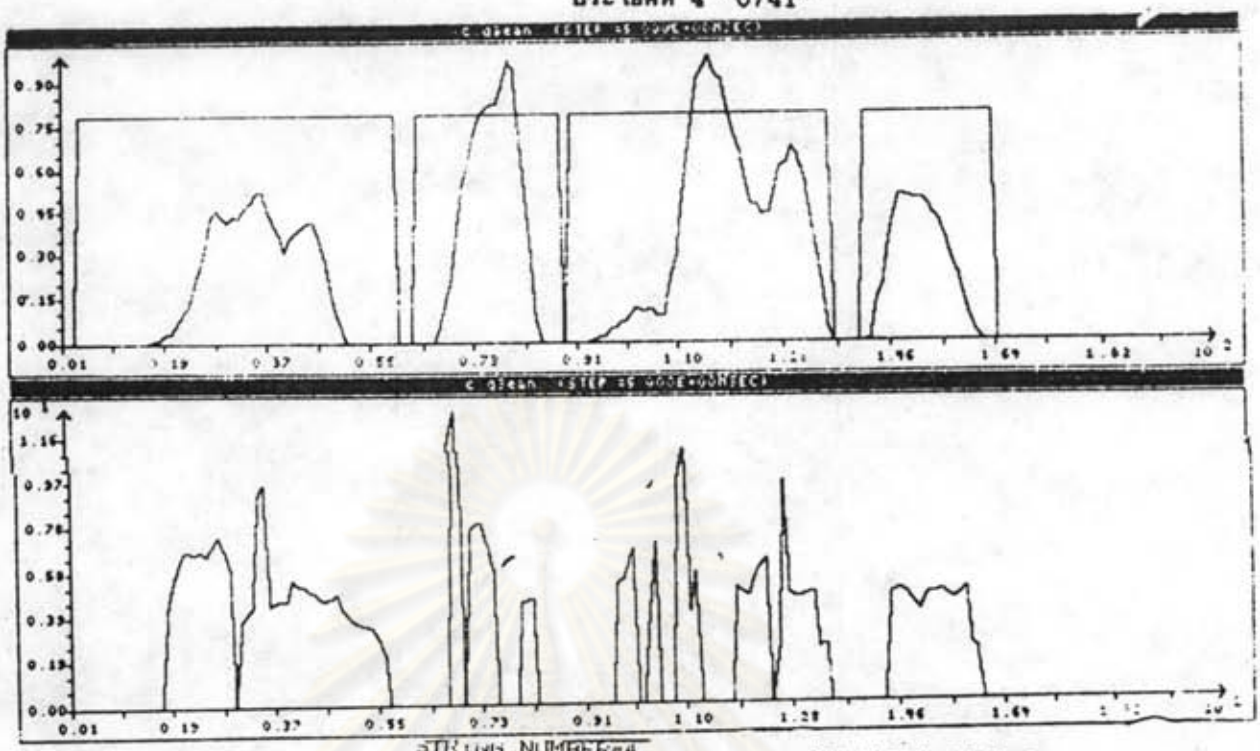
POSITION1=0
POSITION2=70
STRING NUMBER=1
PATH VALUE= 1.9330621656
STRING NUMBER=2
PATH VALUE= 2.2041438509
STRING NUMBER=3
PATH VALUE= 1.7520651706
STRING NUMBER=4
PATH VALUE= 2.1065477879
STRING NUMBER=5
PATH VALUE= 1.8536072267
STRING NUMBER=6
PATH VALUE= 2.3526390993
STRING NUMBER=7
PATH VALUE= 2.0656724079
STRING NUMBER=8
PATH VALUE= 1.8209727901
STRING NUMBER=9
PATH VALUE= 1.9297322870
STRING NUMBER=10
PATH VALUE= 2.3494432224
WORD=3
POSITION1=77
POSITION2=97
STRING NUMBER=1
PATH VALUE= 3.1389138867
STRING NUMBER=2
PATH VALUE= 2.6531961433
STRING NUMBER=3
PATH VALUE= 2.6493289509
STRING NUMBER=4
PATH VALUE= 3.1155872100
STRING NUMBER=5
PATH VALUE= 2.8058100606
STRING NUMBER=6
PATH VALUE= 2.0281804350
STRING NUMBER=7
PATH VALUE= 2.8885239122
STRING NUMBER=8
PATH VALUE= 3.1393642332
STRING NUMBER=9
PATH VALUE= 2.7189112383
STRING NUMBER=10
PATH VALUE= 2.7826069386
WORD=6
POSITION1=105
POSITION2=153
STRING NUMBER=1
PATH VALUE= 2.3772975097
STRING NUMBER=2
PATH VALUE= 1.9290362147
STRING NUMBER=3
PATH VALUE= 2.2325827285
STRING NUMBER=4
PATH VALUE= 2.2052541418
STRING NUMBER=5
PATH VALUE= 2.4048718670
STRING NUMBER=6
PATH VALUE= 2.0376422927
STRING NUMBER=7
PATH VALUE= 2.1650142014
STRING NUMBER=8
PATH VALUE= 2.2724540847
STRING NUMBER=9
PATH VALUE= 2.0467168907
STRING NUMBER=10
PATH VALUE= 2.2797547072
WORD=2
POSITION1=157
POSITION2=199
STRING NUMBER=1
PATH VALUE= 2.1266051284
STRING NUMBER=2
PATH VALUE= 2.1496227051
STRING NUMBER=3
PATH VALUE= 1.4194547546
STRING NUMBER=4
PATH VALUE= 2.2661535467
STRING NUMBER=5
PATH VALUE= 1.7221426615
STRING NUMBER=6
PATH VALUE= 2.6798246889
STRING NUMBER=7
PATH VALUE= 2.5759610240
STRING NUMBER=8
PATH VALUE= 2.0305278359
STRING NUMBER=9
PATH VALUE= 1.2788434678
STRING NUMBER=10
PATH VALUE= 2.3145836333
WORD=9

ประโยคที่ 3 "1791"



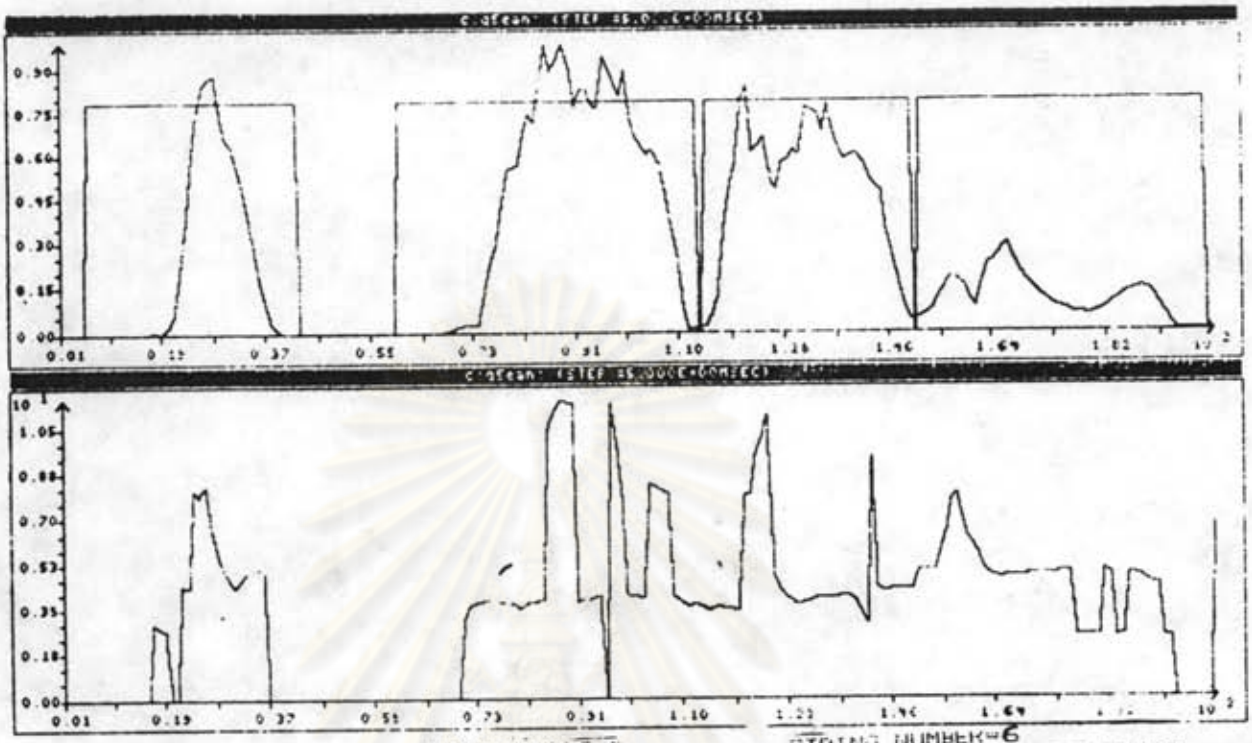
POSITION1=6
POSITION2=56
STRING NUMBER=1
PATH VALUE= 1.7113517959
STRING NUMBER=2
PATH VALUE= 2.3465976472
STRING NUMBER=3
PATH VALUE= 2.3248092902
STRING NUMBER=4
PATH VALUE= 2.0071670146
STRING NUMBER=5
PATH VALUE= 2.4016326592
STRING NUMBER=6
PATH VALUE= 2.1930087890
STRING NUMBER=7
PATH VALUE= 1.8398792935
STRING NUMBER=8
PATH VALUE= 2.1114622480
STRING NUMBER=9
PATH VALUE= 2.3618101068
STRING NUMBER=10
PATH VALUE= 2.3226864317
WORD=1
POSITION1=60
POSITION2=103
STRING NUMBER=1
PATH VALUE= 2.3708764380
STRING NUMBER=2
PATH VALUE= 2.4574392513
STRING NUMBER=3
PATH VALUE= 2.4244383517
STRING NUMBER=4
PATH VALUE= 2.0760137746
STRING NUMBER=5
PATH VALUE= 2.4434171289
STRING NUMBER=6
PATH VALUE= 1.9503828935
STRING NUMBER=7
PATH VALUE= 1.4395312210
STRING NUMBER=8
PATH VALUE= 1.9025785320
STRING NUMBER=9
PATH VALUE= 2.5572677908
STRING NUMBER=10
PATH VALUE= 2.6594363848
WORD=7
POSITION1=104
POSITION2=151
STRING NUMBER=1
PATH VALUE= 2.1446205083
STRING NUMBER=2
PATH VALUE= 2.1253369599
STRING NUMBER=3
PATH VALUE= 1.6827087138
STRING NUMBER=4
PATH VALUE= 2.1487958119
STRING NUMBER=5
PATH VALUE= 1.6253845198
STRING NUMBER=6
PATH VALUE= 2.6754265880
STRING NUMBER=7
PATH VALUE= 2.5641722360
STRING NUMBER=8
PATH VALUE= 2.3240801538
STRING NUMBER=9
PATH VALUE= 1.5508720803
STRING NUMBER=10
PATH VALUE= 2.2407421224
WORD=9
POSITION1=152
POSITION2=191
STRING NUMBER=1
PATH VALUE= 1.4619773385
STRING NUMBER=2
PATH VALUE= 2.2557133556
STRING NUMBER=3
PATH VALUE= 2.2895330725
STRING NUMBER=4
PATH VALUE= 1.8198680529
STRING NUMBER=5
PATH VALUE= 2.2853568151
STRING NUMBER=6
PATH VALUE= 2.2858668101
STRING NUMBER=7
PATH VALUE= 2.2879062092
STRING NUMBER=8
PATH VALUE= 2.3016196410
STRING NUMBER=9
PATH VALUE= 2.3231002445
STRING NUMBER=10
PATH VALUE= 2.0844661624
WORD=1

ประโยคที่ 4 "0741"



POSITION1=4
POSITION2=60
STRING NUMBER=1
PATH VALUE= 1.8599250625
STRING NUMBER=2
PATH VALUE= 1.8755370936
STRING NUMBER=3
PATH VALUE= 2.1180150781
STRING NUMBER=4
PATH VALUE= 1.7667097327
STRING NUMBER=5
PATH VALUE= 2.1130107297
STRING NUMBER=6
PATH VALUE= 1.8646665777
STRING NUMBER=7
PATH VALUE= 1.6610329503
STRING NUMBER=8
PATH VALUE= 1.8265270437
STRING NUMBER=9
PATH VALUE= 2.1262807997
STRING NUMBER=10
PATH VALUE= 1.7583285452
WORD=7
POSITION1=63
POSITION2=89
STRING NUMBER=1
PATH VALUE= 2.2477868463
STRING NUMBER=2
PATH VALUE= 2.1861417645
STRING NUMBER=3
PATH VALUE= 2.2066184156
STRING NUMBER=4
PATH VALUE= 1.9277632204
STRING NUMBER=5
PATH VALUE= 2.0934912941
STRING NUMBER=6
PATH VALUE= 2.0939011868
STRING NUMBER=7
PATH VALUE= 1.4276553341
STRING NUMBER=8
PATH VALUE= 1.6714188320
STRING NUMBER=9
PATH VALUE= 2.2227283472
STRING NUMBER=10
PATH VALUE= 2.4765918704
WORD=7
POSITION1=90
POSITION2=136
STRING NUMBER=1
PATH VALUE= 2.0079029198
STRING NUMBER=2
PATH VALUE= 2.1881355942
STRING NUMBER=3
PATH VALUE= 2.0198764173
STRING NUMBER=4
PATH VALUE= 1.5073041185
STRING NUMBER=5
PATH VALUE= 2.0650330402
STRING NUMBER=6
PATH VALUE= 2.3773850140
STRING NUMBER=7
PATH VALUE= 2.0450745155
STRING NUMBER=8
PATH VALUE= 1.9820870652
STRING NUMBER=9
PATH VALUE= 2.0134215592
STRING NUMBER=10
PATH VALUE= 2.2062865932
WORD=4
POSITION1=141
POSITION2=164
STRING NUMBER=1
PATH VALUE= 1.5705030058
STRING NUMBER=2
PATH VALUE= 2.0387279811
STRING NUMBER=3
PATH VALUE= 1.9974513160
STRING NUMBER=4
PATH VALUE= 1.6305786568
STRING NUMBER=5
PATH VALUE= 2.1561327644
STRING NUMBER=6
PATH VALUE= 2.2166366061
STRING NUMBER=7
PATH VALUE= 2.2612574518
STRING NUMBER=8
PATH VALUE= 2.2665946901
STRING NUMBER=9
PATH VALUE= 2.0061546782
STRING NUMBER=10
PATH VALUE= 1.9101915223
WORD=1

ประโยคที่ 5 "6550"

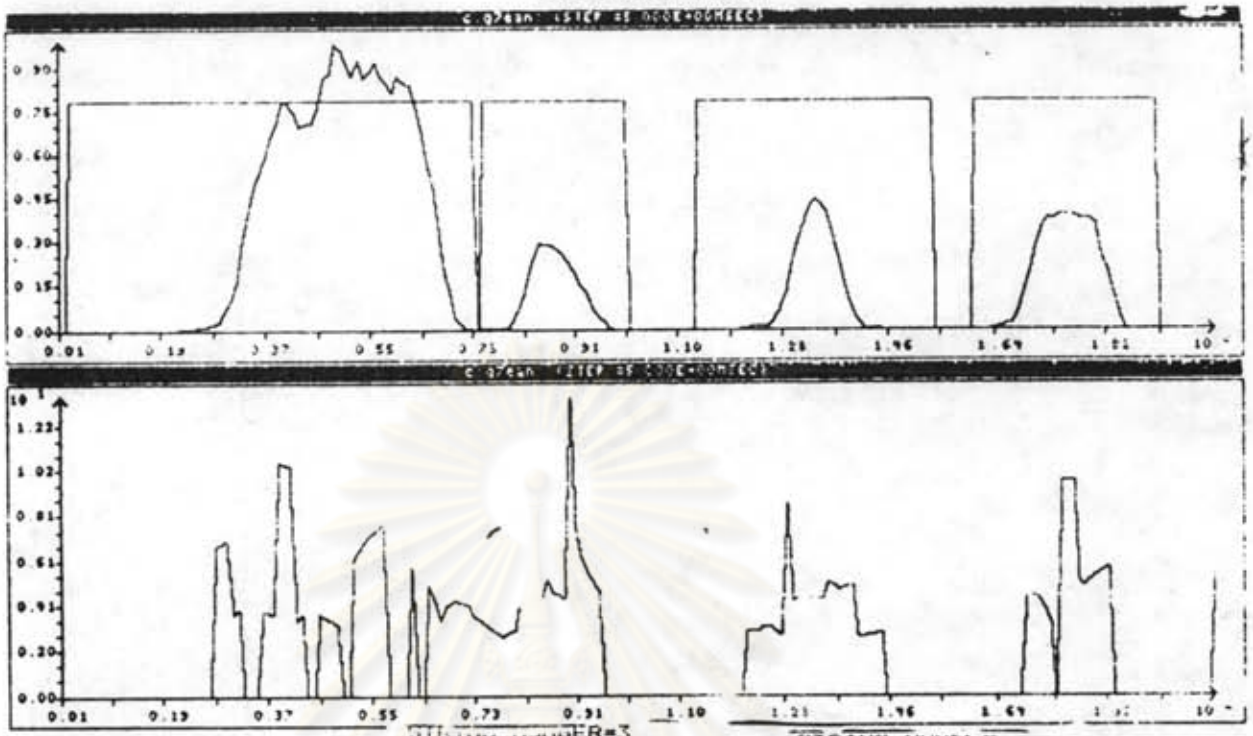


POSITION1=6
POSITION2=43
STRING NUMBER=1
PATH VALUE= 2.4558449350
STRING NUMBER=2
PATH VALUE= 2.3230422386
STRING NUMBER=3
PATH VALUE= 2.4428677139
STRING NUMBER=4
PATH VALUE= 2.2266339041
STRING NUMBER=5
PATH VALUE= 2.7810289918
STRING NUMBER=6
PATH VALUE= 1.5855343383
STRING NUMBER=7
PATH VALUE= 2.3310162574
STRING NUMBER=8
PATH VALUE= 1.8234065860
STRING NUMBER=9
PATH VALUE= 2.2268186597
STRING NUMBER=10
PATH VALUE= 2.3327070306
WORD=6
POSITION1=80
POSITION2=116
STRING NUMBER=1
PATH VALUE= 2.2633784360
STRING NUMBER=2
PATH VALUE= 2.5972644682

STRING NUMBER=3
PATH VALUE= 2.0373407054
STRING NUMBER=4
PATH VALUE= 2.4547240520
STRING NUMBER=5
PATH VALUE= 1.5313941174
STRING NUMBER=6
PATH VALUE= 2.8829750249
STRING NUMBER=7
PATH VALUE= 2.2016730641
STRING NUMBER=8
PATH VALUE= 2.0104486147
STRING NUMBER=9
PATH VALUE= 2.0561575655
STRING NUMBER=10
PATH VALUE= 2.5473937787
WORD=5
POSITION1=114
POSITION2=150
STRING NUMBER=1
PATH VALUE= 1.8617616494
STRING NUMBER=2
PATH VALUE= 2.0145863477
STRING NUMBER=3
PATH VALUE= 1.7904104558
STRING NUMBER=4
PATH VALUE= 2.1536989457
STRING NUMBER=5
PATH VALUE= 1.5966676364

STRING NUMBER=6
PATH VALUE= 2.6123756487
STRING NUMBER=7
PATH VALUE= 2.5081103614
STRING NUMBER=8
PATH VALUE= 2.1215735669
STRING NUMBER=9
PATH VALUE= 1.8763668733
STRING NUMBER=10
PATH VALUE= 2.1530647125
WORD=5
POSITION1=151
POSITION2=199
STRING NUMBER=1
PATH VALUE= 2.1010758172
STRING NUMBER=2
PATH VALUE= 1.6892993112
STRING NUMBER=3
PATH VALUE= 2.0013817415
STRING NUMBER=4
PATH VALUE= 1.9245274680
STRING NUMBER=5
PATH VALUE= 2.1166000601
STRING NUMBER=6
PATH VALUE= 2.3027304934
STRING NUMBER=7
PATH VALUE= 2.7782674336
STRING NUMBER=8
PATH VALUE= 2.4328413518
STRING NUMBER=9
PATH VALUE= 1.8937245023
STRING NUMBER=10
PATH VALUE= 1.4087906424
WORD=6

ประวัติที่ 6 "5667"

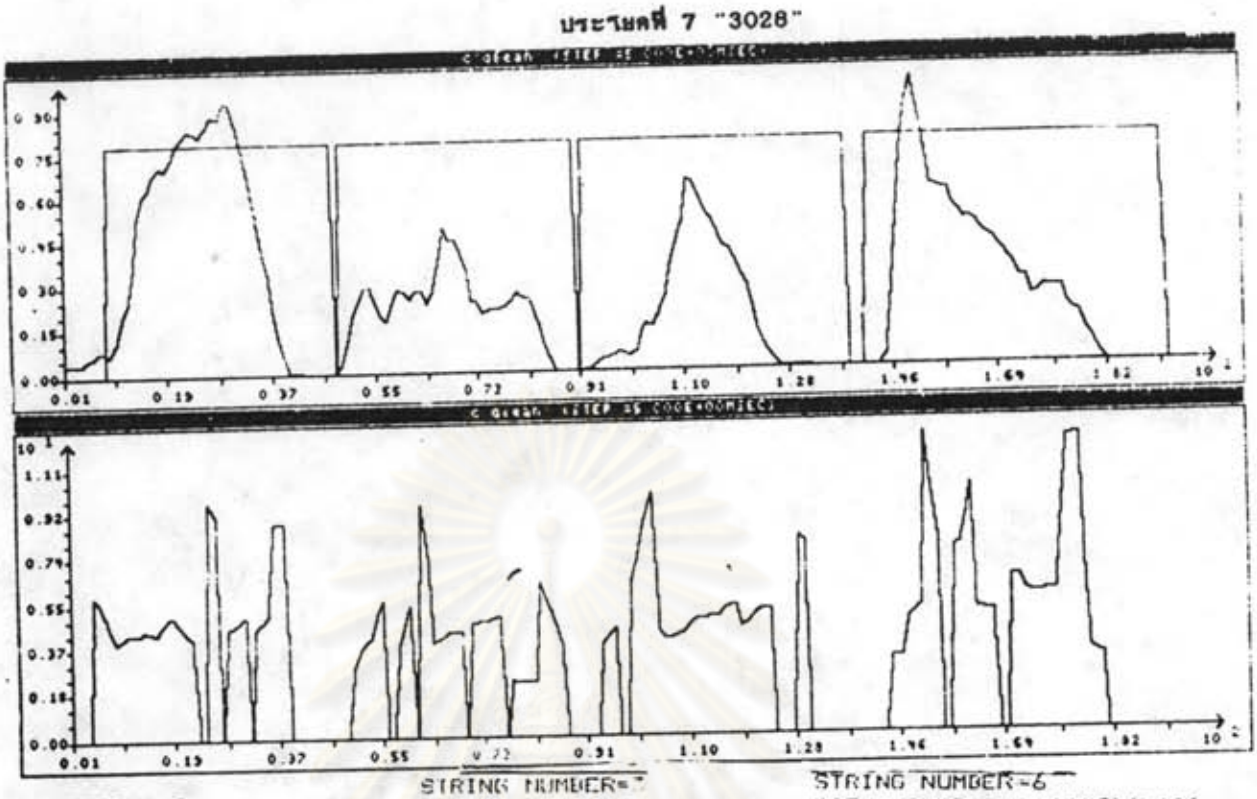


POSITION1=3
POSITION2=74
STRING NUMBER=1
PATH VALUE= 2.7258998787
STRING NUMBER=2
PATH VALUE= 2.2843924336
STRING NUMBER=3
PATH VALUE= 2.0828307782
STRING NUMBER=4
PATH VALUE= 2.3539859468
STRING NUMBER=5
PATH VALUE= 1.8141193052
STRING NUMBER=6
PATH VALUE= 2.6585147207
STRING NUMBER=7
PATH VALUE= 2.1176284406
STRING NUMBER=8
PATH VALUE= 2.0975301794
STRING NUMBER=9
PATH VALUE= 1.9766470076
STRING NUMBER=10
PATH VALUE= 2.4585293459
WORD=5
POSITION1=75
POSITION2=101
STRING NUMBER=1
PATH VALUE= 2.3565244186
STRING NUMBER=2
PATH VALUE= 2.2999221127

STRING NUMBER=3
PATH VALUE= 2.3004286575
STRING NUMBER=4
PATH VALUE= 1.7780017994
STRING NUMBER=5
PATH VALUE= 2.6460120427
STRING NUMBER=6
PATH VALUE= 1.4222606288
STRING NUMBER=7
PATH VALUE= 2.5180395901
STRING NUMBER=8
PATH VALUE= 2.4396216532
STRING NUMBER=9
PATH VALUE= 2.3226550787
STRING NUMBER=10
PATH VALUE= 1.9110343644
WORD=6
POSITION1=113
POSITION2=154
STRING NUMBER=1
PATH VALUE= 2.4277079371
STRING NUMBER=2
PATH VALUE= 2.6043885977
STRING NUMBER=3
PATH VALUE= 2.5135656779
STRING NUMBER=4
PATH VALUE= 2.3346691261
STRING NUMBER=5
PATH VALUE= 2.6967030474

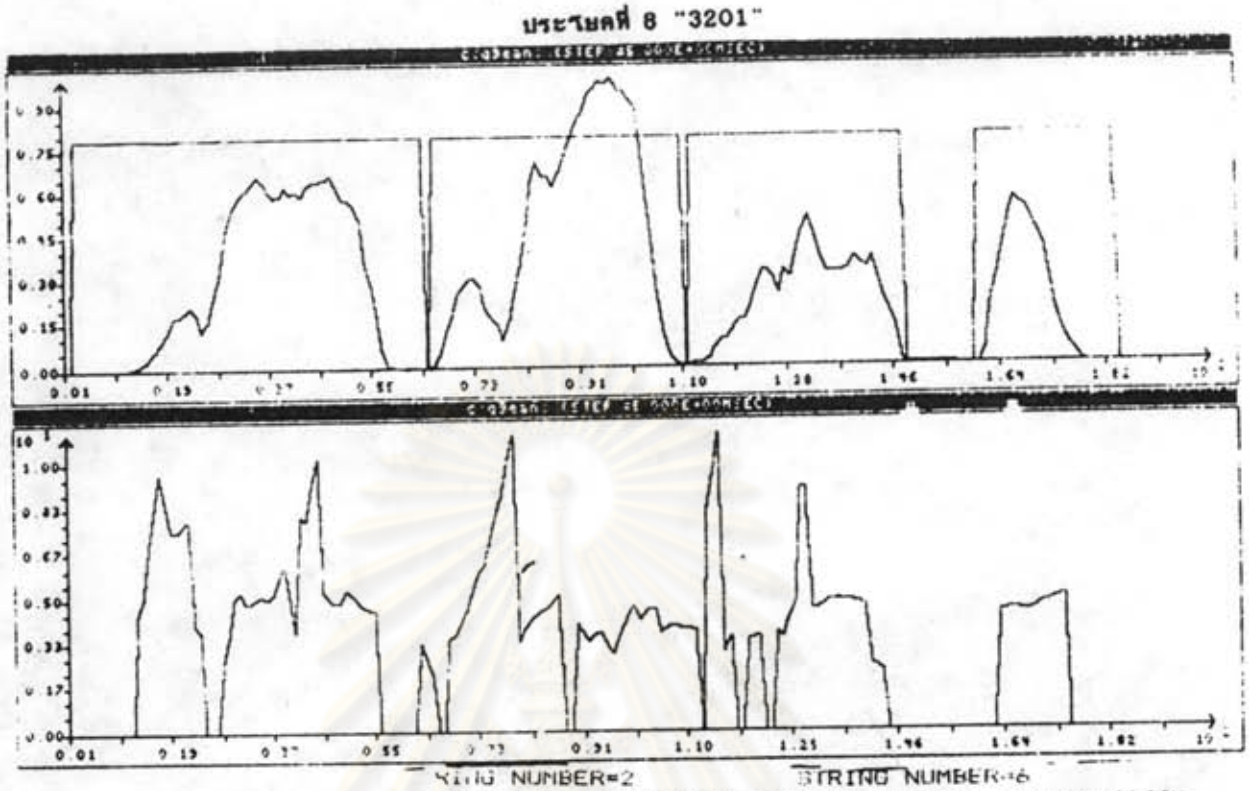
STRING NUMBER=6
PATH VALUE= 1.8438303867
STRING NUMBER=7
PATH VALUE= 2.3661994169
STRING NUMBER=8
PATH VALUE= 1.9136599431
STRING NUMBER=9
PATH VALUE= 2.3314936977
STRING NUMBER=10
PATH VALUE= 2.2893960177
WORD=6
POSITION1=161
POSITION2=191
STRING NUMBER=1
PATH VALUE= 2.1011907819
STRING NUMBER=2
PATH VALUE= 2.3160728501
STRING NUMBER=3
PATH VALUE= 2.2030493514
STRING NUMBER=4
PATH VALUE= 1.8901593339
STRING NUMBER=5
PATH VALUE= 2.2919365748
STRING NUMBER=6
PATH VALUE= 1.8321517194
STRING NUMBER=7
PATH VALUE= 1.5044779311
STRING NUMBER=8
PATH VALUE= 1.7273795772
STRING NUMBER=9
PATH VALUE= 2.2660701504
STRING NUMBER=10
PATH VALUE= 2.3801486580
WORD=7

ประโยคที่ 7 "3028"



POSITION1=9
POSITION2=46
STRING NUMBER=1
PATH VALUE= 1.8401664876
STRING NUMBER=2
PATH VALUE= 1.7544078299
STRING NUMBER=3
PATH VALUE= 1.2893109937
STRING NUMBER=4
PATH VALUE= 1.8960700194
STRING NUMBER=5
PATH VALUE= 1.7437160543
STRING NUMBER=6
PATH VALUE= 2.1184479859
STRING NUMBER=7
PATH VALUE= 2.3506252858
STRING NUMBER=8
PATH VALUE= 2.0413697251
STRING NUMBER=9
PATH VALUE= 1.7497911844
STRING NUMBER=10
PATH VALUE= 2.1159611488
WORD=3
POSITION1=49
POSITION2=91
STRING NUMBER=1
PATH VALUE= 1.7276383900
STRING NUMBER=2
PATH VALUE= 1.8256062464
STRING NUMBER=3
PATH VALUE= 1.9835297048
STRING NUMBER=4
PATH VALUE= 1.6102559139
STRING NUMBER=5
PATH VALUE= 2.0933984170
STRING NUMBER=6
PATH VALUE= 2.0489033743
STRING NUMBER=7
PATH VALUE= 2.2300692340
STRING NUMBER=8
PATH VALUE= 2.2864572199
STRING NUMBER=9
PATH VALUE= 2.0830344355
STRING NUMBER=10
PATH VALUE= 1.6000842337
WORD=0
POSITION1=92
POSITION2=138
STRING NUMBER=1
PATH VALUE= 2.0276136718
STRING NUMBER=2
PATH VALUE= 1.4392012020
STRING NUMBER=3
PATH VALUE= 1.9292089290
STRING NUMBER=4
PATH VALUE= 1.8073884024
STRING NUMBER=5
PATH VALUE= 2.2315437415
STRING NUMBER=6
PATH VALUE= 1.8267667108
STRING NUMBER=7
PATH VALUE= 2.0276658364
STRING NUMBER=8
PATH VALUE= 2.0505680793
STRING NUMBER=9
PATH VALUE= 2.0025014449
STRING NUMBER=10
PATH VALUE= 2.0525017776
WORD=2
POSITION1=141
POSITION2=192
STRING NUMBER=1
PATH VALUE= 2.2357156846
STRING NUMBER=2
PATH VALUE= 2.5183257767
STRING NUMBER=3
PATH VALUE= 2.0876377782
STRING NUMBER=4
PATH VALUE= 2.2394413007
STRING NUMBER=5
PATH VALUE= 2.1504506797
STRING NUMBER=6
PATH VALUE= 1.7816970007
STRING NUMBER=7
PATH VALUE= 1.8361959718
STRING NUMBER=8
PATH VALUE= 1.2500713718
STRING NUMBER=9
PATH VALUE= 2.1391315752
STRING NUMBER=10
PATH VALUE= 2.5307257304
WORD=0

ประโยคที่ 8 "3201"

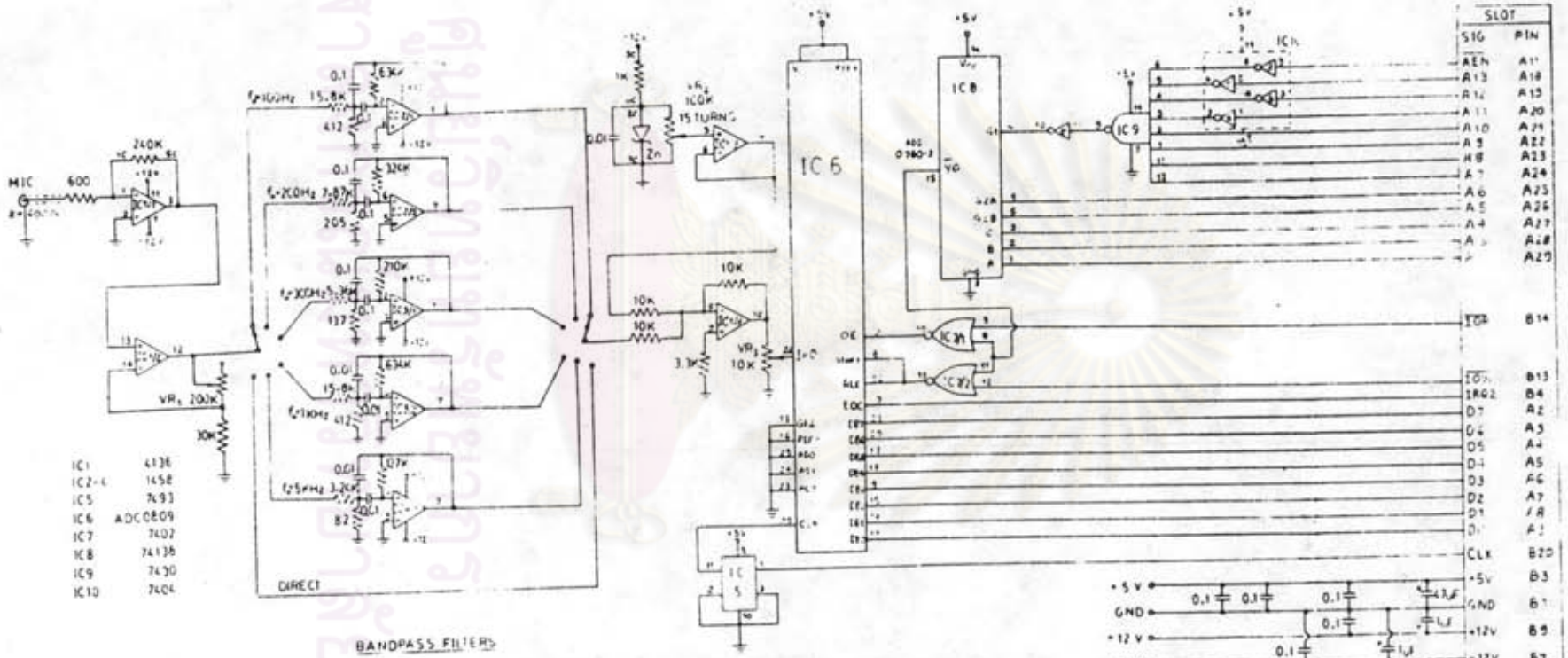


POSITION1=3
POSITION2=65
STRING NUMBER=1
PATH VALUE= 2.0053223535
STRING NUMBER=2
PATH VALUE= 2.0742954376
STRING NUMBER=3
PATH VALUE= 1.4342346588
STRING NUMBER=4
PATH VALUE= 2.0409295197
STRING NUMBER=5
PATH VALUE= 2.0161498073
STRING NUMBER=6
PATH VALUE= 2.1977093790
STRING NUMBER=7
PATH VALUE= 2.0811754159
STRING NUMBER=8
PATH VALUE= 1.7723667456
STRING NUMBER=9
PATH VALUE= 1.9232765866
STRING NUMBER=10
PATH VALUE= 2.2157370447
WORD=3
POSITION1=66
POSITION2=110
STRING NUMBER=1
PATH VALUE = 2.1240517607

STRING NUMBER=2
PATH VALUE= 1.3909980677
STRING NUMBER=3
PATH VALUE= 1.8032173759
STRING NUMBER=4
PATH VALUE= 1.8110530223
STRING NUMBER=5
PATH VALUE= 2.1083434343
STRING NUMBER=6
PATH VALUE= 2.1741872639
STRING NUMBER=7
PATH VALUE= 2.2148003896
STRING NUMBER=8
PATH VALUE= 2.0904608016
STRING NUMBER=9
PATH VALUE= 1.8118089671
STRING NUMBER=10
PATH VALUE= 2.0969441777
WORD=2
POSITION1=111
POSITION2=148
STRING NUMBER=1
PATH VALUE= 2.0992543149
STRING NUMBER=2
PATH VALUE= 1.8398645971
STRING NUMBER=3
PATH VALUE= 1.9840220657
STRING NUMBER=4
PATH VALUE= 1.5844235340
STRING NUMBER=5
PATH VALUE= 2.1970514465

STRING NUMBER=6
PATH VALUE= 2.3180655752
STRING NUMBER=7
PATH VALUE= 2.3974896075
STRING NUMBER=8
PATH VALUE= 2.2271475744
STRING NUMBER=9
PATH VALUE= 2.0647784[illegible]
STRING NUMBER=10
PATH VALUE= 1.6781[illegible]
WORD=0
POSITION1=160
POSITION2=184
STRING NUMBER=1
PATH VALUE= 1.3773941[illegible]
STRING NUMBER=2
PATH VALUE= 2.3213889[illegible]
STRING NUMBER=3
PATH VALUE= 2.1804005[illegible]
STRING NUMBER=4
PATH VALUE 1.9567960[illegible]
STRING NUMBER=5
PATH VALUE 2.0138[illegible]
STRING NUMBER=6
PATH VALUE 2.2501833[illegible]
STRING NUMBER=7
PATH VALUE= 2.2852409329
STRING NUMBER=8
PATH VALUE 2.1874197208
STRING NUMBER=9
PATH VALUE 2.1996252806
STRING NUMBER=10
PATH VALUE= 2.1060230128
WORD=1

ภาคผนวก ช. วงจรแปลงสัญญาณเสียงพูดต่อเนื่องให้เป็นสัญญาณไม่ต่อเนื่อง (A/D)



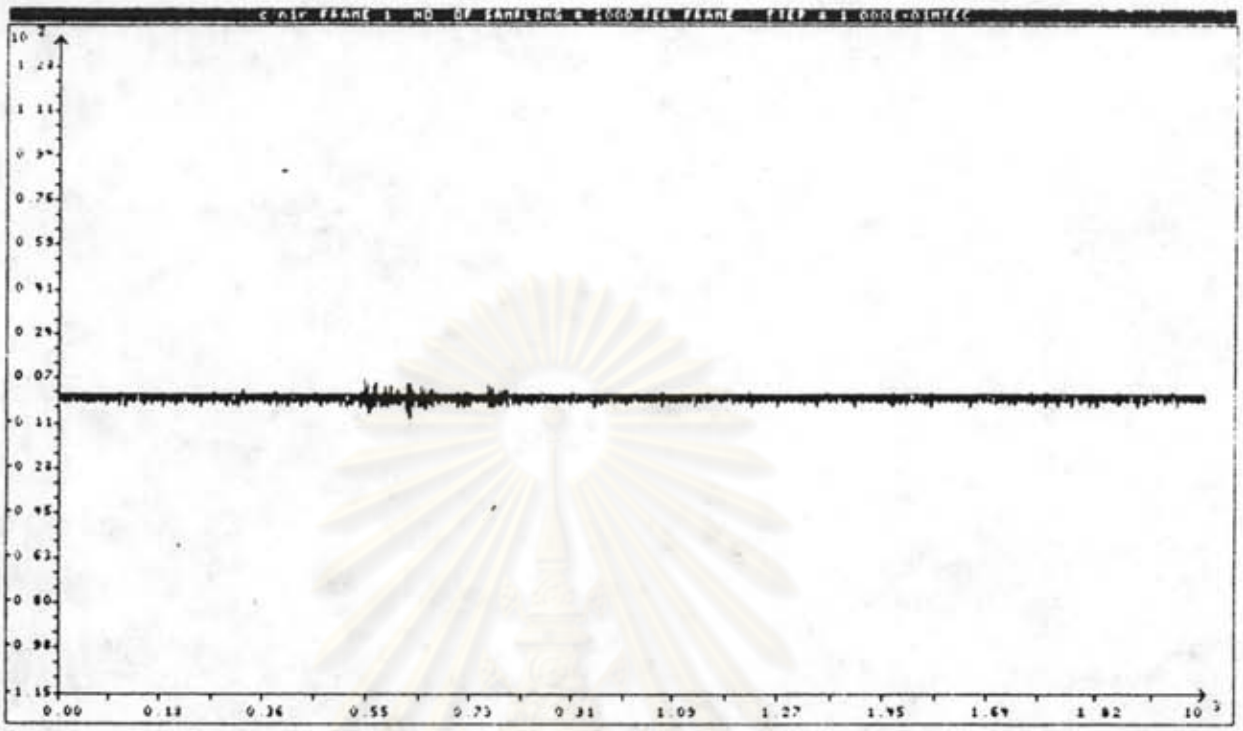
BANDPASS FILTER

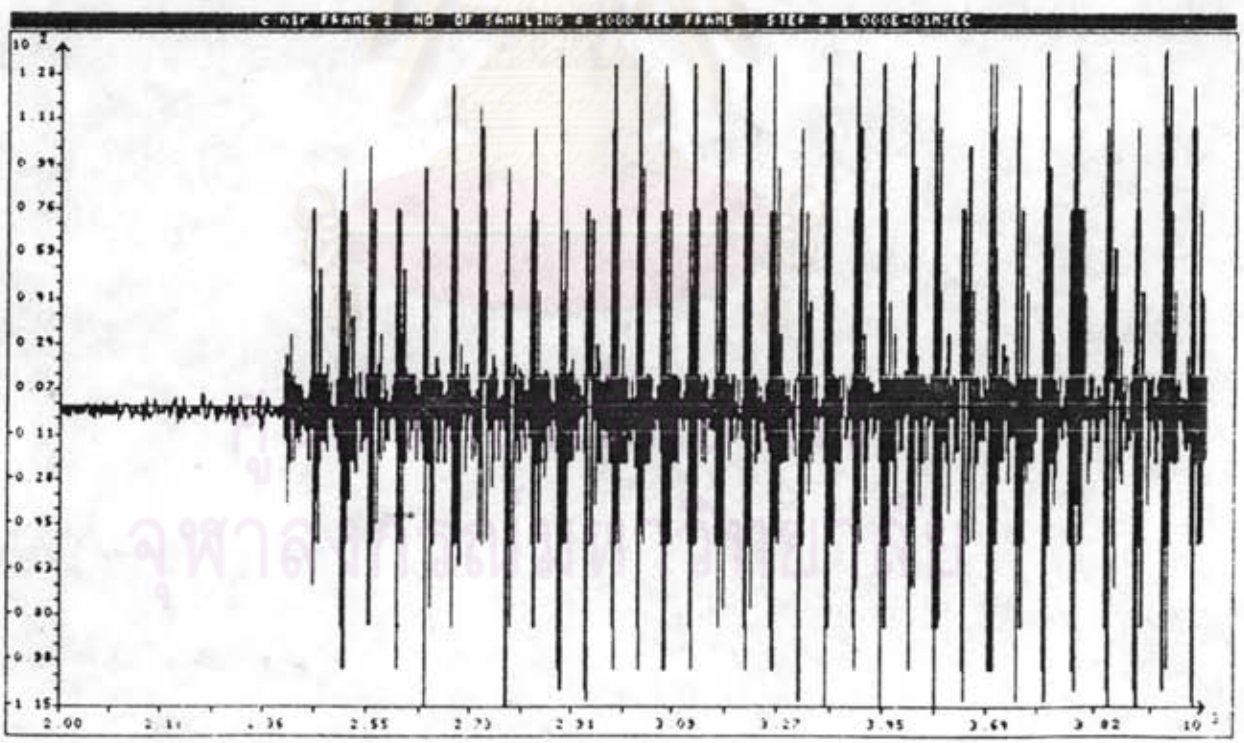
A TO D CIRCUIT

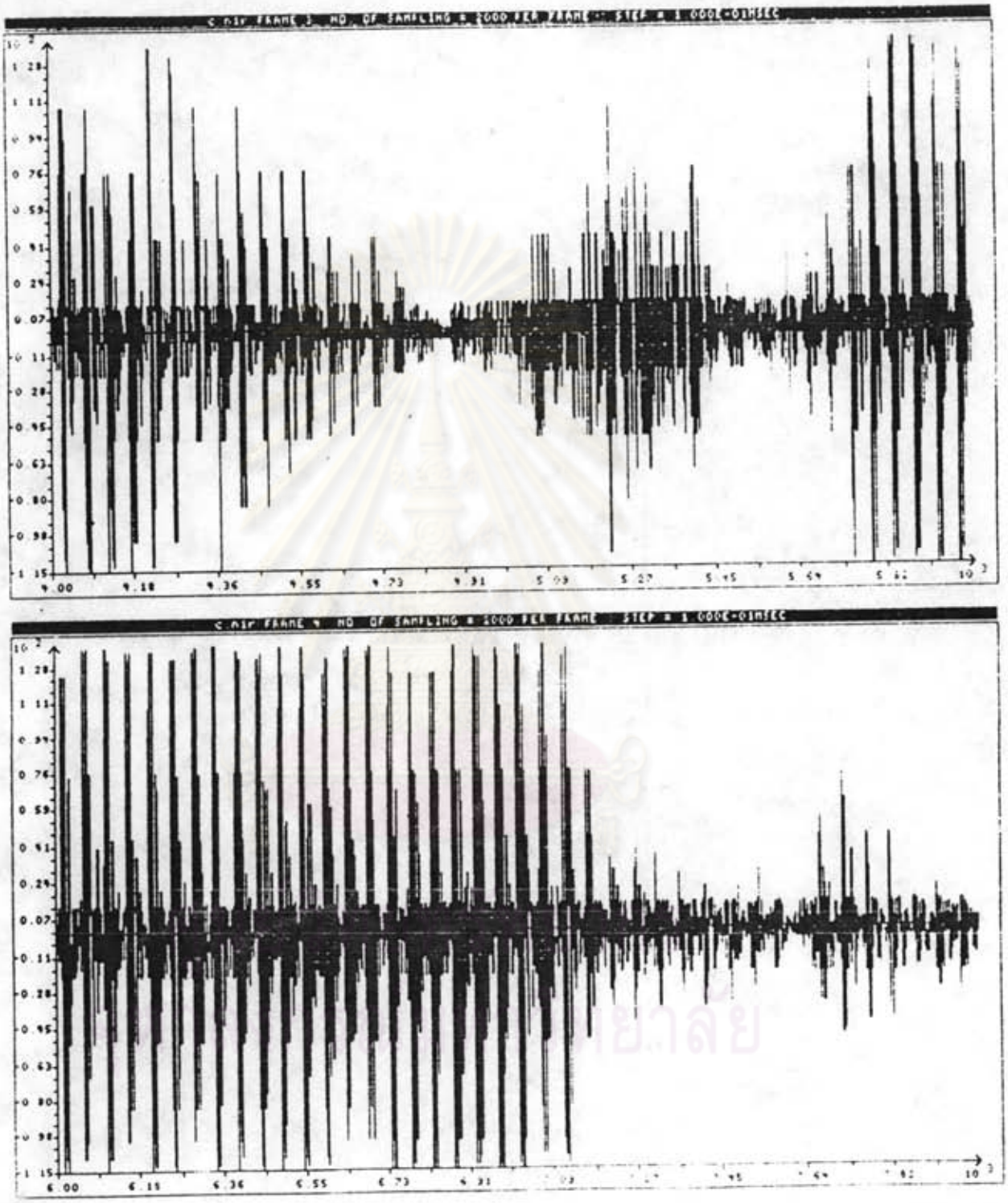
IBM PC I/O BUS

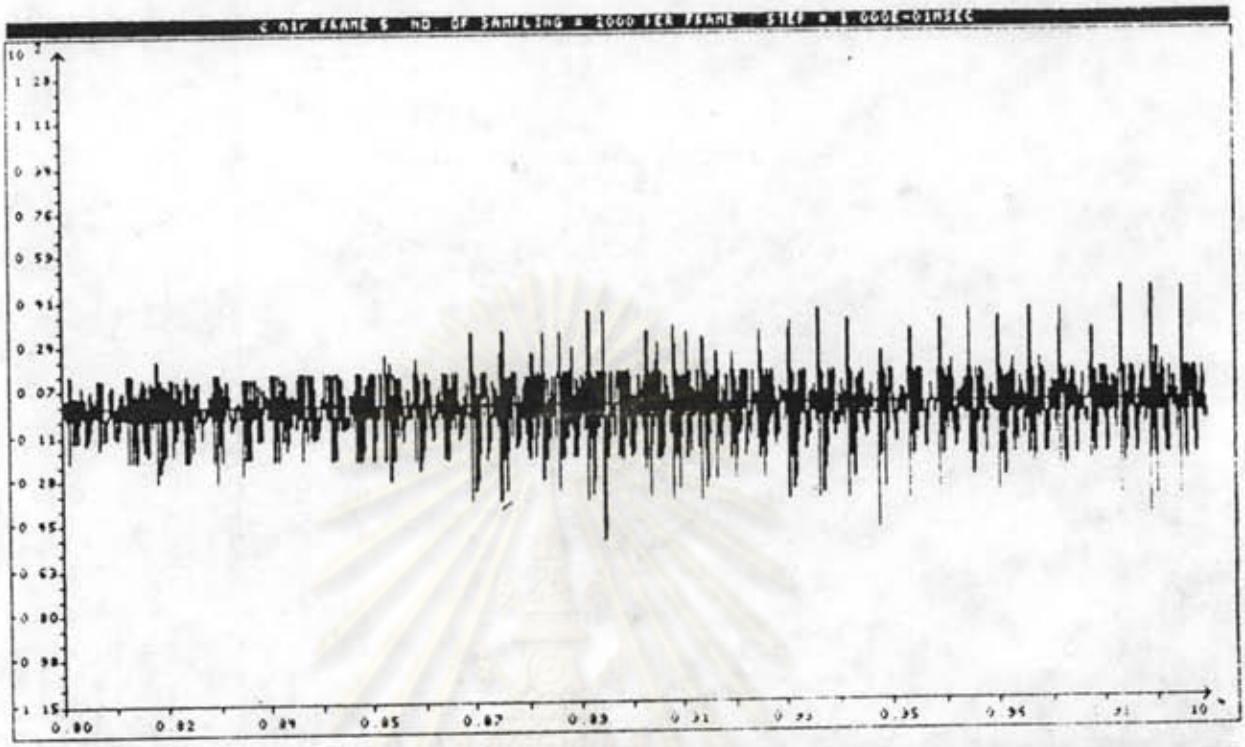
(PORT ADDRESS 0780 - 831

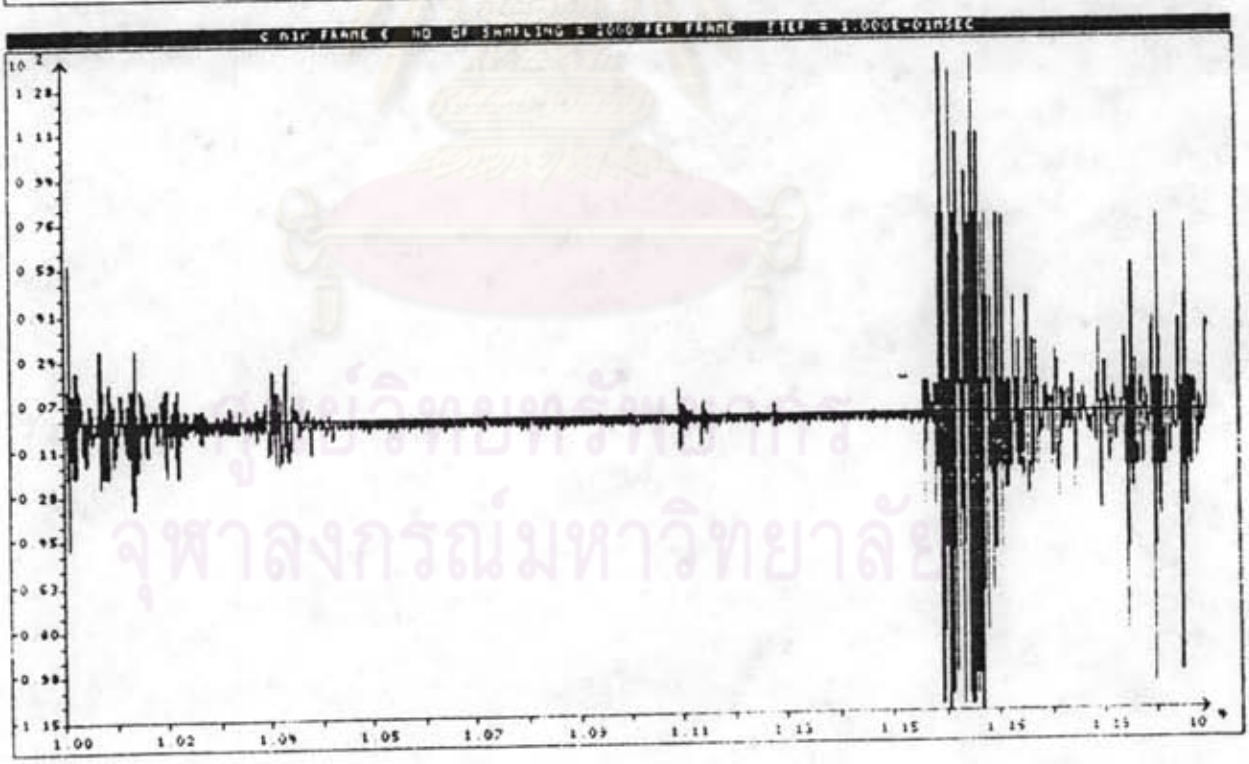
ภาคผนวก ช. ตัวอย่างเสียงพูดในประโยคที่ 1 "นาย ชาลี พงษ์สมบูรณ์"

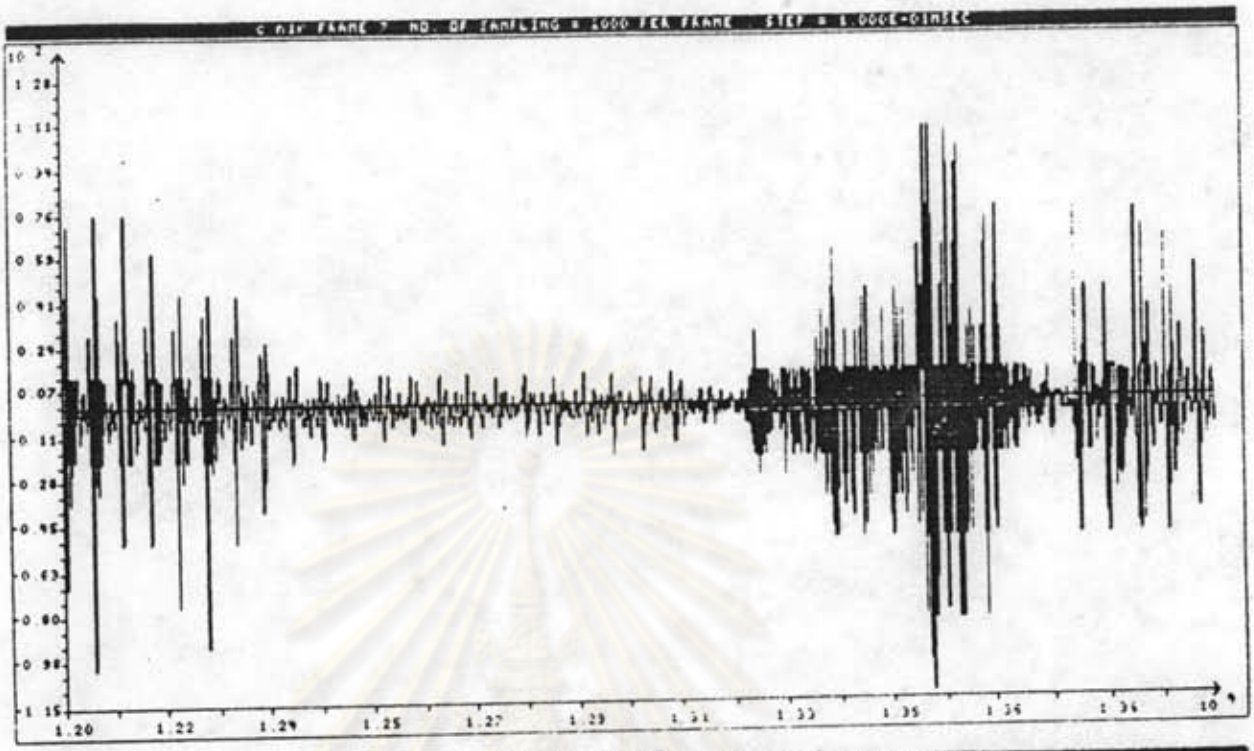


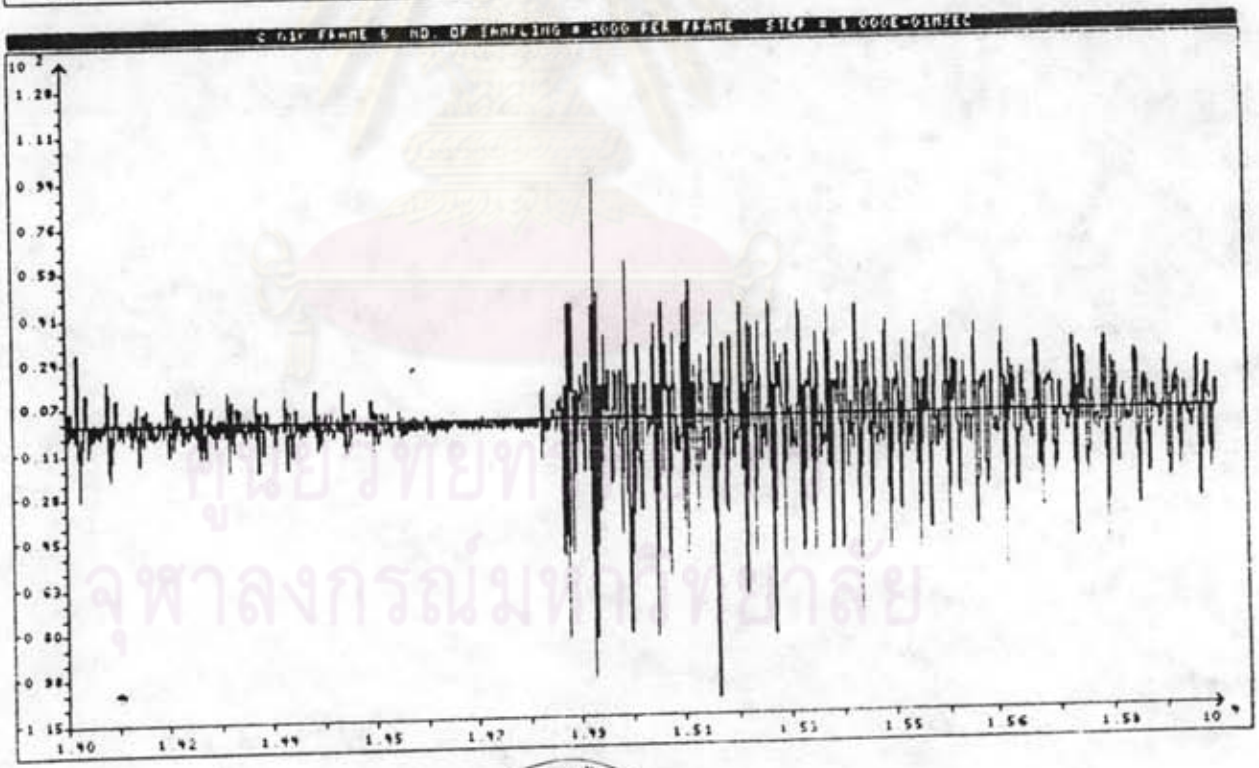


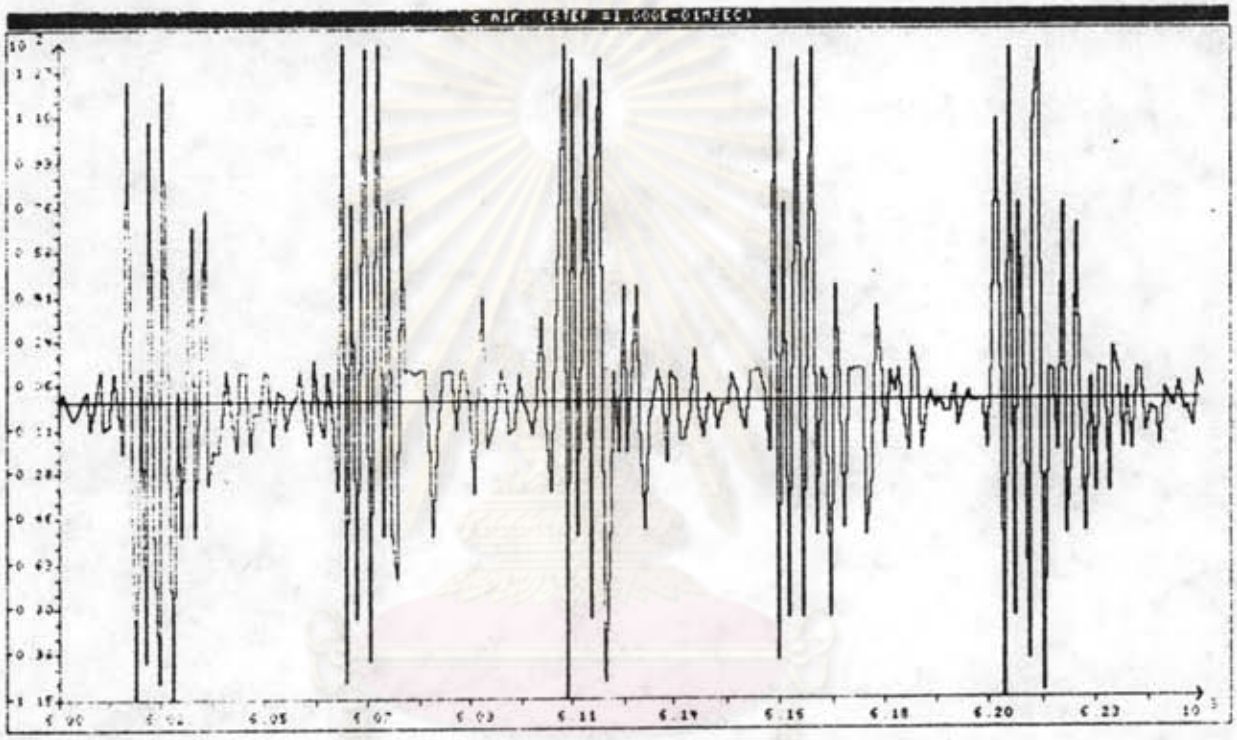












ภาคผนวก ฌ. รายละเอียดโปรแกรม

```
program VOICE;

{ I typedef.sys}
{ I graphix.sys}
{ I kernel.sys}
{ I windows.sys}
{ I findwrld.hgh}
{ I axis.hgh}
{ I polygon.hgh}
{ I spline.hgh}
{ I tawi4.sys}
{ I tawi41.sys}

{THIS PROCEDURE FOR CREATE PATTERN FILE AND LABEL FILE}

OVERLAY procedure CREATE_PATTERN_FILE;
type string3=string[3];
     fileofstring=file of string3;
     fileofreal=file of real;
     record_of_rc=record
                  RC1  :real;
                  RC2  :real;
                  RC3  :real;
                  RC4  :real;
                  RC5  :real;
                  RC6  :real;
                  RC7  :real;
                  RC8  :real;
                  RC9  :real;
                  RC10 :real;
                  RC11 :real;
                  RC12 :real;
                  RC13 :real;
                  RC14 :real;
                  RC15 :real;
                  end;
      fileofrc=file of record_of_rc;

var tempfileofstring,infileofstring,outfileofstring:fileofstring;
    tempfileofreal,infileofreal,outfileofreal:fileofreal;
    tempfileofrc,infileofrc,outfileofrc:fileofrc;
    temprecordofrc:record_of_rc;
    tempreal:real;
    tempstring:string3;
    filenameofstring,filenameofreal,filenameofrc:str12;

function CREATEFILE_STRING(var Fi:FILEOFSTRING):Boolean;
begin
  {$I-} reset(Fi); {$I+}
  createfile_string:=IOresult=0;
```

```
end;

procedure CREATEFILEOFSTRING(var filenameofstring:STR12);
label LABEL1;
begin
LABEL1:;
  assign(tempfileofstring,filenameofstring);
  if not CREATEFILE_STRING(tempfileofstring) then
  begin
    rewrite(tempfileofstring);
    close(tempfileofstring);
  end
  else
    begin
      writeln('***OLD DATA FILE***');
      write('DO YOU WANT TO WRITE OVER IT?(Y/N)');
      repeat
      readln(ch);
      until ch in ['n','y'];
      if ch='y' then
      begin
        rewrite(tempfileofstring);close(tempfileofstring);
      end;
      if ch='n' then
      begin
        write('WHAT NEW FILENAME?');
        readln(filenameofstring);
        goto LABEL1;
      end;
    end;
end;

procedure RESETINFILEOFSTRING(filenameofstring:STR12);
begin
  assign(infileofstring,filenameofstring);
  if CREATEFILE_STRING(infileofstring) then
  begin
    reset(infileofstring);
  end
  else
    writeln('NO FILE',filenameofstring);
end;

procedure RESETOUTFILEOFSTRING(filenameofstring:STR12);
begin
  assign(outfileofstring,filenameofstring);
  if CREATEFILE_STRING(outfileofstring) then
  begin
    reset(outfileofstring);
  end
  else
```

```
    writeln('NO FILE',filenameofstring);
end;

function CREATEFILE_REAL(var Fi:FILEOFREAL):Boolean;
begin
  {$I-} reset(Fi); {$I+}
  createfile_real:=IOresult=0;
end;

procedure CREATEFILEOFREAL(var filenameofreal:STR12);
label LABEL1;
begin
LABEL1:;
  assign(tempfileofreal,filenameofreal);
  if not CREATEFILE_REAL(tempfileofreal) then
  begin
    rewrite(tempfileofreal);
    close(tempfileofreal);
  end
  else
    begin
      writeln('***OLD DATA FILE***');
      write('DO YOU WANT TO WRITE OVER IT?(Y/N)');
      repeat
      readln(ch);
      until ch in ['n','y'];
      if ch='y' then
      begin
        rewrite(tempfileofreal);close(tempfileofreal);
      end;
      if ch='n' then
      begin
        write('WHAT NEW FILENAME?');
        readln(filenameofreal);
        goto LABEL1;
      end;
    end;
end;

procedure RESETINFILEOFREAL(filenameofreal:STR12);
begin
  assign(infileofreal,filenameofreal);
  if CREATEFILE_REAL(infileofreal) then
  begin
    reset(infileofreal);
  end
  else
    writeln('NO FILE',filenameofreal);
end;
```

```
procedure RESETOUTFILEOFREAL(filenameofreal:STR12);
begin
  assign(outfileofreal,filenameofreal);
  if CREATEFILE_REAL(outfileofreal) then
  begin
    reset(outfileofreal);
  end
  else
    writeln('NO FILE',filenameofreal);
end;

function CREATEFILE_RC(var Fi:FILEOFRC):Boolean;
begin
  {$I-} reset(Fi); {$I+}
  createfile_rc:=IOresult=0;
end;

procedure CREATEFILEOFRC(var filenameofrc:STR12);
label LABEL1;
begin
LABEL1:;
  assign(tempfileofrc,filenameofrc);
  if not CREATEFILE_RC(tempfileofrc) then
  begin
    rewrite(tempfileofrc);
    close(tempfileofrc);
  end
  else
    begin
      writeln('***OLD DATA FILE***');
      write('DO YOU WANT TO WRITE OVER IT?(Y/N)');
      repeat
      readln(ch);
      until ch in ['n','y'];
      if ch='y' then
      begin
        rewrite(tempfileofrc);close(tempfileofrc);
      end;
      if ch='n' then
      begin
        write('WHAT NEW FILENAME?');
        readln(filenameofrc);
        goto LABEL1;
      end;
    end;
end;

procedure RESETINFILEOFRC(filenameofrc:STR12);
begin
  assign(infileofrc,filenameofrc);
  if CREATEFILE_RC(infileofrc) then
```

```
  begin
    reset(infileofrc);
  end
  else
    writeln('NO FILE',filenameofrc);
end;

procedure RESETOUTFILEOFRC(filenameofrc:STR12);
begin
  assign(outfileofrc,filenameofrc);
  if CREATEFILE_RC(outfileofrc) then
  begin
    reset(outfileofrc);
  end
  else
    writeln('NO FILE',filenameofrc);
end;
begin
  nosound;
  write('WHAT PATTERN LABEL FILE NAME?');
  readln(filenameofstring);
  write('WHAT PATTERN RC. FILE NAME?');
  readln(filenameofrc);
  write('WHAT PATTERN ALPHA FILE NAME?');
  readln(filenameofreal);
  createfileofstring(filenameofstring);
  resetoutfileofstring(filenameofstring);
  write(outfileofstring,tempstring);
  close(outfileofstring);
  createfileofrc(filenameofrc);
  resetoutfileofrc(filenameofrc);
  write(outfileofrc,temprecordofrc);
  close(outfileofrc);
  createfileofreal(filenameofreal);
  resetoutfileofreal(filenameofreal);
  write(outfileofreal,tempreal);
  close(outfileofreal);
end;

{THIS PROCEDURE FOR ADD REFLECTION COEFFICIENT TO PATTERN FILE}
{AND ADD LABEL TO LABEL FILE}

OVERLAY procedure ADD_PATTERN_FILE;
type string3=string[3];
     fileofstring=file of string3;
     fileofreal=file of real;
     record of rc=record
                    RC1   :real;
                    RC2   :real;
                    RC3   :real;
                    RC4   :real;
```

```
                    RC5   :real;
                    RC6   :real;
                    RC7   :real;
                    RC8   :real;
                    RC9   :real;
                    RC10  :real;
                    RC11  :real;
                    RC12  :real;
                    RC13  :real;
                    RC14  :real;
                    RC15  :real;
                    end;
      fileofrc=file of record of rc;

var tempfileofstring,infileofstring,outfileofstring:fileofstring;
    tempfileofreal,infileofreal,outfileofreal:fileofreal;
    tempfileofrc,infileofrc,outfileofrc:fileofrc;
    temprecordofrc:record of rc;
    tempreal:real;
    tempstring:string3;
    filenameofstring,filenameofalpha,filenameofrc,filenameofdata:str12

function CREATEFILE_STRING(var Fi:FILEOFSTRING):Boolean;
begin
  {$I-} reset(Fi); {$I+}
  createfile_string:=IOresult=0;
end;

procedure CREATEFILEOFSTRING(var filenameofstring:STR12);
label LABEL1;
begin
LABEL1:;
  assign(tempfileofstring,filenameofstring);
  if not CREATEFILE_STRING(tempfileofstring) then
  begin
    rewrite(tempfileofstring);
    close(tempfileofstring);
  end
  else
    begin
      writeln('***OLD DATA FILE***');
      write('DO YOU WANT TO WRITE OVER IT?(Y/N)');
      repeat
      readln(ch);
      until ch in ['n','y'];
      if ch='y' then
      begin
        rewrite(tempfileofstring);close(tempfileofstring);
      end;
      if ch='n' then
      begin
```

```
          write('WHAT NEW FILENAME?');
          readln(filenameofstring);
          goto LABEL1;
        end;
    end;
end;

procedure RESETINFILEOFSTRING(filenameofstring:STR12);
begin
  assign(infileofstring,filenameofstring);
  if CREATEFILE_STRING(infileofstring) then
  begin
    reset(infileofstring);
  end
  else
    writeln('NO FILE',filenameofstring);
end;

procedure RESETOUTFILEOFSTRING(filenameofstring:STR12);
begin
  assign(outfileofstring,filenameofstring);
  if CREATEFILE_STRING(outfileofstring) then
  begin
    reset(outfileofstring);
  end
  else
    writeln('NO FILE',filenameofstring);
end;


function CREATEFILE_REAL(var Fi:FILEOFREAL):Boolean;
begin
  {$I-} reset(Fi); {$I+}
  createfile_real:=IOresult=0;
end;

procedure CREATEFILEOFREAL(var filenameofreal:STR12);
label LABEL1;
begin
LABEL1:;
  assign(tempfileofreal,filenameofreal);
  if not CREATEFILE_REAL(tempfileofreal) then
  begin
    rewrite(tempfileofreal);
    close(tempfileofreal);
  end
  else
    begin
      writeln('***OLD DATA FILE***');
      write('DO YOU WANT TO WRITE OVER IT?(Y/N)');
```

```
        readln(ch);
        until ch in ['n','y'];
        if ch='y' then
        begin
          rewrite(tempfileofreal);close(tempfileofreal);
        end;
        if ch='n' then
        begin
          write('WHAT NEW FILENAME?');
          readln(filenameofreal);
          goto LABEL1;
        end;
      end;
end;

procedure RESETINFILEOFREAL(filenameofreal:STR12);
begin
  assign(infileofreal,filenameofreal);
  if CREATEFILE_REAL(infileofreal) then
  begin
    reset(infileofreal);
  end
  else
    writeln('NO FILE',filenameofreal);
end;

procedure RESETOUTFILEOFREAL(filenameofreal:STR12);
begin
  assign(outfileofreal,filenameofreal);
  if CREATEFILE_REAL(outfileofreal) then
  begin
    reset(outfileofreal);
  end
  else
    writeln('NO FILE',filenameofreal);
end;

function CREATEFILE_RC(var Fi:FILEOFRC):Boolean;
begin
  {$I-} reset(Fi); {$I+}
  createfile_rc:=IOresult=0;
end;

procedure CREATEFILEOFRC(var filenameofrc:STR12);
label LABEL1;
begin
LABEL1:;
  assign(tempfileofrc,filenameofrc);
  if not CREATEFILE_RC(tempfileofrc) then
  begin
    rewrite(tempfileofrc);
```

```
    close(tempfileofrc);
  end
  else
    begin
      writeln('***OLD DATA FILE***');
      write('DO YOU WANT TO WRITE OVER IT?(Y/N)');
      repeat
      readln(ch);
      until ch in ['n','y'];
      if ch='y' then
      begin
        rewrite(tempfileofrc);close(tempfileofrc);
      end;
      if ch='n' then
      begin
        write('WHAT NEW FILENAME?');
        readln(filenameofrc);
        goto LABEL1;
      end;
    end;
end;

procedure RESETINFILEOFRC(filenameofrc:STR12);
begin
  assign(infileofrc,filenameofrc);
  if CREATEFILE_RC(infileofrc) then
  begin
    reset(infileofrc);
  end
  else
    writeln('NO FILE',filenameofrc);
end;

procedure RESETOUTFILEOFRC(filenameofrc:STR12);
begin
  assign(outfileofrc,filenameofrc);
  if CREATEFILE_RC(outfileofrc) then
  begin
    reset(outfileofrc);
  end
  else
    writeln('NO FILE',filenameofrc);
end;
begin
  nosound;
  write('WHAT PATTERN LABEL FILE NAME?');
  readln(filenameofstring);
  write('WHAT PATTERN RC. FILE NAME?');
  readln(filenameofrc);
  write('WHAT PATTERN ALPHA FILE NAME?');
  readln(filenameofalpha);
```

```
  write('WHAT DATA FILE NAME?');
  readln(filenameofdata);
  write('WHAT LABEL DO YOU WANT TO ASSIGN?');
  readln(tempstring);
  resetoutfileofstring(filenameofstring);
  seek(outfileofstring,filesize(outfileofstring));
  write(outfileofstring,tempstring);
  close(outfileofstring);
  ENERGYALPHA_RC_PATTERN(filenameofrc,filenameofalpha,filenameofdata);
end;

{THIS PROCEDURE FOR RECOGNITION BY COMPARE RC.OF SYLLABLE WITH}
{RC.OF PATTERN}

OVERLAY procedure RECOGNITION;
type string3=string[3];
     fileofstring=file of string3;
     fileofreal=file of real;
     fileofinteger=file of integer;
     record_of_rc=record
                  RC1  :real;
                  RC2  :real;
                  RC3  :real;
                  RC4  :real;
                  RC5  :real;
                  RC6  :real;
                  RC7  :real;
                  RC8  :real;
                  RC9  :real;
                  RC10 :real;
                  RC11 :real;
                  RC12 :real;
                  RC13 :real;
                  RC14 :real;
                  RC15 :real;
                  end;
      fileofrc=file of record_of_rc;

var tempfileofstring,infileofstring,outfileofstring:fileofstring;
    tempfileofreal,infileofreal,outfileofreal:fileofreal;
    tempfileofrc,infileofrc,outfileofrc:fileofrc;
    tempfileofrc1,infileofrc1,outfileofrc1:fileofrc;
    tempfileofinteger,infileofinteger,outfileofinteger:fileofinteger;
    temprecordrc,temprecordrc1:record_of_rc;
    tempreal,A,B,C,D,F:real;
    tempstring,tempstring1,tempstring2:string3;
    filenameofstring,filenameofalpha,filenameofrc:str12;
    filenameofrc1,filenameofdata,filenameofinteger,filenameofboundary:str12
    distance_difference,path:array[0..100,0..50] of real;
    position1,position2,string_no:integer;
```

```
function CREATEFILE_STRING(var Fi:FILEOFSTRING):Boolean;
begin
  {$I-} reset(Fi); {$I+}
  createfile_string:=IOresult=0;
end;

procedure CREATEFILEOFSTRING(var filenameofstring:STR12);
label LABEL1;
begin
LABEL1:;
  assign(tempfileofstring,filenameofstring);
  if not CREATEFILE_STRING(tempfileofstring) then
  begin
    rewrite(tempfileofstring);
    close(tempfileofstring);
  end
  else
    begin
      writeln('***OLD DATA FILE***');
      write('DO YOU WANT TO WRITE OVER IT?(Y/N)');
      repeat
      readln(ch);
      until ch in ['n','y'];
      if ch='y' then
      begin
        rewrite(tempfileofstring);close(tempfileofstring);
      end;
      if ch='n' then
      begin
        write('WHAT NEW FILENAME?');
        readln(filenameofstring);
        goto LABEL1;
      end;
    end;
end;

procedure RESETINFILEOFSTRING(filenameofstring:STR12);
begin
  assign(infileofstring,filenameofstring);
  if CREATEFILE_STRING(infileofstring) then
  begin
    reset(infileofstring);
  end
  else
    writeln('NO FILE',filenameofstring);
end;

procedure RESETOUTFILEOFSTRING(filenameofstring:STR12);
begin
  assign(outfileofstring,filenameofstring);
  if CREATEFILE_STRING(outfileofstring) then
```

```
  begin
    reset(outfileofstring);
  end
  else
    writeln('NO FILE',filenameofstring);
end;


function CREATEFILE_REAL(var Fi:FILEOFREAL):Boolean;
begin
  {$I-} reset(Fi); {$I+}
  createfile_real:=IOresult=0;
end;

procedure CREATEFILEOFREAL(var filenameofreal:STR12);
label LABEL1;
begin
LABEL1:;
  assign(tempfileofreal,filenameofreal);
  if not CREATEFILE_REAL(tempfileofreal) then
  begin
    rewrite(tempfileofreal);
    close(tempfileofreal);
  end
  else
    begin
      writeln('***OLD DATA FILE***');
      write('DO YOU WANT TO WRITE OVER IT?(Y/N)');
      repeat
      readln(ch);
      until ch in ['n','y'];
      if ch='y' then
      begin
        rewrite(tempfileofreal);close(tempfileofreal);
      end;
      if ch='n' then
      begin
        write('WHAT NEW FILENAME?');
        readln(filenameofreal);
        goto LABEL1;
      end;
    end;
end;

procedure RESETINFILEOFREAL(filenameofreal:STR12);
begin
  assign(infileofreal,filenameofreal);
  if CREATEFILE_REAL(infileofreal) then
  begin
    reset(infileofreal);
  end
```

```
  else
    writeln('NO FILE',filenameofreal);
end;

procedure RESETOUTFILEOFREAL(filenameofreal:STR12);
begin
  assign(outfileofreal,filenameofreal);
  if CREATEFILE_REAL(outfileofreal) then
  begin
    reset(outfileofreal);
  end
  else
    writeln('NO FILE',filenameofreal);
end;

function CREATEFILE_RC1(var Fi:FILEOFRC):Boolean;
begin
  {$I-} reset(Fi); {$I+}
  createfile_rc1:=IOresult=0;
end;

procedure RESETINFILEOFRC1(filenameofrc1:STR12);
begin
  assign(infileofrc1,filenameofrc1);
  if CREATEFILE_RC1(infileofrc1) then
  begin
    reset(infileofrc1);
  end
  else
    writeln('NO FILE',filenameofrc1);
end;


function CREATEFILE_RC(var Fi:FILEOFRC):Boolean;
begin
  {$I-} reset(Fi); {$I+}
  createfile_rc:=IOresult=0;
end;

procedure CREATEFILEOFRC(var filenameofrc:STR12);
label LABEL1;
begin
LABEL1:;
  assign(tempfileofrc,filenameofrc);
  if not CREATEFILE_RC(tempfileofrc) then
  begin
    rewrite(tempfileofrc);
    close(tempfileofrc);
  end
  else
    begin
```

```
      writeln('***OLD DATA FILE***');
      write('DO YOU WANT TO WRITE OVER IT?(Y/N)');
      repeat
      readln(ch);
      until ch in ['n','y'];
      if ch='y' then
      begin
        rewrite(tempfileofrc);close(tempfileofrc);
      end;
      if ch='n' then
      begin
        write('WHAT NEW FILENAME?');
        readln(filenameofrc);
        goto LABEL1;
      end;
    end;
end;

procedure RESETINFILEOFRC(filenameofrc:STR12);
begin
  assign(infileofrc,filenameofrc);
  if CREATEFILE_RC(infileofrc) then
  begin
    reset(infileofrc);
  end
  else
    writeln('NO FILE',filenameofrc);
end;

procedure RESETOUTFILEOFRC(filenameofrc:STR12);
begin
  assign(outfileofrc,filenameofrc);
  if CREATEFILE_RC(outfileofrc) then
  begin
    reset(outfileofrc);
  end
  else
    writeln('NO FILE',filenameofrc);
end;

function CREATEFILE_INTEGER(var Fi:FILEOFINTEGER):Boolean;
begin
  {$I-} reset(Fi); {$I+}
  createfile_integer:=IOresult=0;
end;

procedure CREATEFILEOFINTEGER(var filenameofinteger:STR12);
label LABEL1;
begin
LABEL1:;
  assign(tempfileofinteger,filenameofinteger);
```

```
  if not CREATEFILE_INTEGER(tempfileofinteger) then
  begin
    rewrite(tempfileofinteger);
    close(tempfileofinteger);
  end
  else
    begin
      writeln('***OLD DATA FILE***');
      write('DO YOU WANT TO WRITE OVER IT?(Y/N)');
      repeat
      readln(ch);
      until ch in ['n','y'];
      if ch='y' then
      begin
        rewrite(tempfileofinteger);close(tempfileofinteger);
      end;
      if ch='n' then
      begin
        write('WHAT NEW FILENAME?');
        readln(filenameofinteger);
        goto LABEL1;
      end;
    end;
end;

procedure RESETINFILEOFINTEGER(filenameofinteger:STR12);
begin
  assign(infileofinteger,filenameofinteger);
  if CREATEFILE_INTEGER(infileofinteger) then
  begin
    reset(infileofinteger);
  end
  else
    writeln('NO FILE',filenameofinteger);
end;

procedure RESETOUTFILEOFINTEGER(filenameofinteger:STR12);
begin
  assign(outfileofinteger,filenameofinteger);
  if CREATEFILE_INTEGER(outfileofinteger) then
  begin
    reset(outfileofinteger);
  end
  else
    writeln('NO FILE',filenameofinteger);
end;


begin
nosound;
write('WHAT PATTERN LABEL FILE NAME?');
```

```
readln(filenameofstring);
write('WHAT PATTERN RC. FILE NAME?');
readln(filenameofrc);
write('WHAT RC. FILE NAME TO RECOGNIZE?');
readln(filenameofrc1);
filenameofboundary:='c:boundary';
resetinfileofinteger(filenameofboundary);
seek(infileofinteger,1);
while not EOF(infileofinteger) do
  begin
    WRITELN('READ INFILEOFINTEGER');
    read(infileofinteger,position1);
    read(infileofinteger,position2);
    WRITELN(LST,'POSITION1=',POSITION1);
    WRITELN(LST,'POSITION2=',POSITION2);

    resetinfileofstring(filenameofstring);
    seek(infileofstring,1);
    string_no:=0;
    F:=1000000.0;
    while not EOF(infileofstring) do
      begin
        string_no:=string_no+1;
        WRITELN(lst,'STRING NUMBER=',string_no);
        WRITELN('READ INFILEOFSTRING');
        read(infileofstring,tempstring1);
        resetinfileofrc1(filenameofrc1);
        seek(infileofrc1,position1);
        if position2-position1+1 > 100 then position2:=position1+100-1;
        for i:=position1 to position2 do
          begin
            {WRITELN('READ INFILEOFRC1');}
            read(infileofrc1,temprecordrc1);
            resetinfileofrc(filenameofrc);
            seek(infileofrc,50*(string_no-1)+1);
            for j:=1 to 50 do
              begin
                tempreal:=0;
                {WRITELN('READ INFILEOFRC');}
                read(infileofrc,temprecordrc);
                tempreal:=tempreal+abs(temprecordrc.rc1-temprecordrc1.rc1);
                tempreal:=tempreal+abs(temprecordrc.rc2-temprecordrc1.rc2);
                tempreal:=tempreal+abs(temprecordrc.rc3-temprecordrc1.rc3);
                tempreal:=tempreal+abs(temprecordrc.rc4-temprecordrc1.rc4);
                tempreal:=tempreal+abs(temprecordrc.rc5-temprecordrc1.rc5);
                tempreal:=tempreal+abs(temprecordrc.rc6-temprecordrc1.rc6);
                tempreal:=tempreal+abs(temprecordrc.rc7-temprecordrc1.rc7);
                tempreal:=tempreal+abs(temprecordrc.rc8-temprecordrc1.rc8);
                tempreal:=tempreal+abs(temprecordrc.rc9-temprecordrc1.rc9);
                tempreal:=tempreal+abs(temprecordrc.rc10-temprecordrc1.rc10);
                tempreal:=tempreal+abs(temprecordrc.rc11-temprecordrc1.rc11);
```

```
                tempreal:=tempreal+abs(temprecordrc.rc12-temprecordrc1.rc12)
                tempreal:=tempreal+abs(temprecordrc.rc13-temprecordrc1.rc13)
                tempreal:=tempreal+abs(temprecordrc.rc14-temprecordrc1.rc14
                tempreal:=tempreal+abs(temprecordrc.rc15-temprecordrc1.rc15)
                distance_difference[i-position1+1,j]:=tempreal;
               { WRITE(lst,distance_difference[i-position1+1,j]); }
              end;
           { WRITELN(LST); }
            close(infileofrc);
          end;
        path[0,0]:=0;
        for j:=1 to 50 do path[0,j]:=10000000.0;
        for i:=1 to position2-position1+1 do
          begin
            path[i,0]:=1000000.0;
            for j:=1 to 50 do
              begin
                d:=distance_difference[i,j];
                A:=path[i-1,j]+d;
                B:=path[i-1,j-1]+d+d;
                C:=path[i,j-1]+d;
                if A<=B then if A<=C then path[i,j]:=A;
                if B<=A then if B<=C then path[i,j]:=B;
                if C<=A then if C<=B then path[i,j]:=C;
              end;
          end;
        tempreal:=path[i,j]/(i+j);
        WRITELN(lst,'PATH VALUE=',tempreal);
        if tempreal<=F then
                        begin
                          F:=tempreal;
                          tempstring2:=tempstring1;
                        end;
        close(infileofrc1);
        {WRITELN(lst,'F VALUE=',F); }
        {WRITELN(lst,'TEMPSTRING2=',tempstring2);}
      end;
    writeln(lst,'WORD=',tempstring2);
    writeln('WORD=',tempstring2);
    close(infileofstring);
  end;
 close(infileofinteger),
end;

{THIS PROCEDURE IS THE MANUE}

OVERLAY procedure MENU;
begin
  clrscr;
  writeln('                ***********************************************');
  writeln('                *                                             *')
```

```
  writeln('               * THAI SPEECH RECOGNITION USING SYLLABLE UNIT *');
  writeln('               *                          by  TAWI PRATHUMTHAN *')
  writeln('               *                                               *');
  writeln('      ***********************************************************
***');

  writeln('      *---->ANALYSIS CHOICE<------* * *----->SEGMENT SYLLABLE CHOICE
--*');

  writeln('      ***********************************************************
***');

  writeln('      * 1=PLOT DATA                *   * f=NORMALIZE
  *');

  writeln('      * 2=TRANSFORM DATA           *   * g=SEGMENT
  *');

  writeln('      * 3=HARDCOPY SPEECH SIGNAL  ********************************
***');

  writeln('      * 4=ENERGYALPHA              * * *------>RECOGNITION CHOICE<---
--*');

  writeln('      * 5=FORMANT                  ********************************
***');

  writeln('      * 6=PITCH PERIOD             *   * i=CREATE PATTERN FILE
  *');

  writeln('      * 7=INFORMATION              *   * k=ADD PATTERN FILE
  *');

  writeln('      * 9=ENERGY SQUARE            *   * r=RECOGNITION
  *');

  writeln('      * c=ROOT MEAN SQUARE ENERGY ********************************
***');

  writeln('      * d=SMOOTH BY 3 VALUE        * * *
  *');

  writeln('      *   AVERAGE                  * * *---------> e=END <----------
--*');

  writeln('      *                            * * *
  *');

  writeln('      ***********************************************************
***');

end;

{MAIN PROGRAM}

begin
  repeat
  MENU;
  sound(500);
    read(kbd,choice);
    case choice of
      '1':PLOT;
      '2':TRANSFORM_DATA;
      '3':HARDCOPY_SPEECH_SIGNAL;
      '4':ENERGYALPHA;
      '5':FORMNT;
      '6':SIFT;
      '7':INFORMATION;
      '9':ENERGYSQUARE;
      'c':RMS;
      'd':SMOOTH;
      'f':NORMALIZE;
      'g':SEGMENT;
      'i':CREATE_PATTERN_FILE;
      'k':ADD_PATTERN_FILE;
      'r':RECOGNITION;
    end;
  until choice='e';
  nosound;
end.
```

```
var flag1:integer;
type wrkst=string[100];
const yes='y';
var no_to_plot,start_no,i,n,code : integer;
    a_plot:plotarray;
    wind : integer;
    pomp,head : string[200];
    ch : char;
    wrkst2 : string[5];
    wrkst1,wrkst3 : string[9];

const PI=3.14159265;
type ARR400=array[1..400] of real;
     ARR80=array[1..80] of real;
     ARR5=array[1..5] of real;
     ARR33=array[1..33] of real;
     STR12=string[12];
     FILEOFREAL=file of real;
     FILEOFBYTE=file of byte;
var SPCH:ARR400;
    PBUF:ARR80;
    PITCH:ARR5;
    PITCHACTUAL,BUFFER1:real;
    J,WINDOW_FREQ,DOWN_SAMPL,FS,WINDOW_SIZE,SIZE_OF_FILE:integer;
    TEMPFILE,INFILE,OUTFILE:fileofreal;
    SPEECHFILE:fileofbyte;
    TEMP1FILE_NAME,TEMP2FILE_NAME,FILENAME,RMSFILE_NAME:str12
    DATAFILE_NAME,PITCHFILE_NAME,ENERGYFILE_NAME,FILENAMETEMP1:str12;
    FULL:boolean;
    CHOICE:char;


type XX= array[1..1024] of real;
     YY= array[1..1024] of real;
     FF= array[1..600] of real;
     IXIX= array[1..600] of real;
     AA= array[1..21] of real;
     RCRC = array[1..21] of real;
     ARR3= array[1..3] of real;
     ARR4=array[1..4] of real;
var   M,NPTS,MP,Z,IP,K1,K2,npeak,POWER2 : integer;
      x:XX;
      y:YY;
      alpha,FSK,STEP:real;
      a:AA;
      rc:RCRC;
      f:FF;
      ix:IXIX;
      if1,iff:ARR3;
```

```
      formnt1file_name,formnt2file_name,formnt3file_name:str12;

var   bytetemp:byte;
      realtemp,minus_value,x1,x2,x3:real;

function CREATE(var Fi:FILEOFREAL):Boolean;
begin
  {$I-} reset(Fi); {$I+}
  create:=IOresult=0;
end;

procedure CREATEFILE(var filename:STR12);
label LABEL1;
begin
LABEL1:;
  assign(tempfile,filename);
  if not CREATE(tempfile) then
  begin
    rewrite(tempfile);
    close(tempfile);
  end
  else
    begin
      writeln('***OLD DATA FILE***');
      write('DO YOU WANT TO WRITE OVER IT?(Y/N)');
      repeat
      readln(ch);
      until ch in ['n','y'];
      if ch='y' then
      begin
        rewrite(tempfile);close(tempfile);
      end;
      if ch='n' then
      begin
        write('WHAT NEW FILENAME?');
        readln(filename);
        goto LABEL1;
      end;
    end;
end;

procedure RESETINFILE(filename:STR12);
begin
  assign(infile,filename);
  if CREATE(infile) then
  begin
    reset(infile);
  end
  else
    writeln('NO FILE',filename);
end;
```

```
procedure RESETOUTFILE(filename:STR12);
begin
  assign(outfile,filename);
  if CREATE(outfile) then
  begin
    reset(outfile);
  end
  else
    writeln('NO FILE',filename);
end;


procedure WriteError;
begin
  Writeln('ERROR');
end;

procedure Pompt(Pomp:wrkst);
begin
  SelectWindow(10);
  SetBackGround(0);
  DrawBorder;
  DrawText(2,(YMaxGlb+1)-5,1,Pomp);
 end;

{THIS PROCEDURE IS THE AUTOCORRELATION FOR SIFT}

procedure AUTOSIFT(n,m:integer;var alpha:real;var a:ARR5;
                   var rc:ARR5;var x:ARR80);
var R : array[1..21] of real;
    B : array[1..21] of real;
    IP,JB,NPK,MINC,MF,MP,K,NP,NK,J,MIP : integer;
    S:real;
begin
  for J:=1 to 21 do R[J]:=0.0;
  for J:=1 to 21 do B[J]:=0.0;
  for J:=1 to 5 do A[J]:=0.0;
  for J:=1 to 5 do RC[J]:=0.0;
  alpha:=0.0;
  MP:=M+1;
  for K:=1 to MP do
  begin
    R[K]:=0;
    NK:=N-K+1;
    NP:=1;
    for NP := 1 to NK do
    begin
      NPK:=NP+K-1;
      R[K]:=R[K]+X[NP]*X[NPK];
```

```
    end;
  end;
  ALPHA:=R[1];
  RC[1]:=(-R[2])/R[1] ;
  A[1]:=1;
  A[2]:=RC[1];
  ALPHA:=ALPHA-RC[1]*RC[1]*ALPHA;
  MF:=M;
  for MINC:=2 to MF do
  begin
    M:=MINC-1;
    for J:=1 to MINC do
    begin
      JB:=MINC-J+1;
      B[J]:=A[JB];
    end;
    M:=M+1;
    S:=0;
    for IP:=1 to M do
    begin
      MIP:=M-IP+2;
      S:=S+R[MIP]*A[IP];
    end;
    RC[M]:=-S/ALPHA;
    for IP:=2 to M do
    begin
      A[IP]:=A[IP]+RC[M]*B[IP-1];
    end;
    A[M+1]:=RC[M];
    ALPHA:=ALPHA-RC[M]*RC[M]*ALPHA;
    if ALPHA<=0 then Writeerror;
end;
end;

procedure OVERLAY1;
begin
end;


OVERLAY procedure DIRECT(var a:ARR5;var p:ARR5; m:integer;
                         var d:ARR5;var xin,xout:real);
var J,JJ:integer;

begin
  XOUT:=0.0;
  D[1]:=XIN;
  for J:=1 to M do
  begin
    JJ:=M+1-J;
    XOUT:=XOUT+D[JJ+1]*P[JJ+1];
    D[1]:=D[1]-A[JJ+1]*D[JJ+1];
```

```
    D[JJ+1]:=D[JJ];
  end;
  XOUT:=XOUT+D[1]*P[1];
end;

{THIS PROCEDURE FOR PLOT DATA}

OVERLAY procedure Data_PLOT;
label LABEL1;
begin
  Pompt('WHAT FILE DO YOU WANT TO PLOT?');
  readln(kbd);
  read(kbd,filename);
  Pompt(concat('PLOT FILE ','''',filename,''''));
  readln(kbd);
  RESETINFILE(filename);
  read(infile,step);
  n:=FileSize(infile);
  seek(infile,start_no);
  for i:=1 to no_to_plot do
    begin
      if (start_no + i)>n then
                                  begin
                                    for j:=i to no_to_plot do
                                    begin
                                      a_plot[j,1]:=start_no+j-1;
                                      a_plot[j,2]:=a_plot[j-1,2];
                                    end;
                                    goto LABEL1;
                                  end;
      a_plot[i,1]:=(start_no+i-1)/1.0;
      read(infile,a_plot[i,2]);
    end;
LABEL1:;
  close(infile);
end;

OVERLAY procedure RedefineWindow_Test;
begin
   DefineWindow(1,0,0,trunc((XMaxG1b+1)/3-1),trunc((YMaxG1b+1)/2-IVStepG1b));
   DefineWindow(2,trunc((XMaxG1b+1)/3),0,trunc((XMaxG1b+1)/1.5-1),
                trunc((YMaxG1b+1)/2-IVStepG1b));
   DefineWindow(3,trunc((XMaxG1b+1)/1.5),0,XMaxG1b,
                trunc((YMaxG1b+1)/2-IVStepG1b));
   DefineWindow(4,0,trunc((YMaxG1b+1)/2-IVStepG1b+1),
                trunc((XMaxG1b+1)/3-1),YMaxG1b-IVStepG1b-1);
   DefineWindow(5,trunc((XMaxG1b+1)/3),trunc((YMaxG1b+1)/2-IVStepG1b+1),
                trunc((XMaxG1b+1)/1.5-1),YMaxG1b-IVStepG1b-1);
   DefineWindow(6,trunc((XMaxG1b+1)/1.5),trunc((YMaxG1b+1)/2-IVStepG1b+1),
                XMaxG1b,YMaxG1b-IVStepG1b-1);
   DefineWindow(7,0,0,trunc((XMaxG1b+1)/2-1),YMaxG1b-IVStepG1b-1);
```

```
    DefineWindow(8,trunc((XMaxGlb+1)/2+1),0,XMaxGlb,YMaxGlb-IVStepGlb-1);
    DefineWindow(9,0,0,XMaxGlb,YMaxGlb-IVStepGlb-1);
    DefineWindow(11,0,0,XMaxGlb,trunc((YMaxGlb+1)/2-IVStepGlb));
    DefineWindow(12,0,trunc((YMaxGlb+1)/2-IVStepGlb+1),
                 XMaxGlb,YMaxGlb-IVStepGlb-1);
end;

{THIS PROCEDURE CALCULATE FAST FOURIER TRANSFORM}

OVERLAY procedure FFT(var x:XX;var y:YY;l:integer);
label LABEL1,LABEL2,LABEL3;
var NP,LMX,LM,LO,LI,J1,J,NV2,NPM1,I,K,J2,LIX:integer;
    SCL,ARG,C,S,T1,T2:real;
begin
  NP:=round(exp(l*ln(2)));
  LMX:=NP;
  SCL:=6.283185303/NP;
  for LO:=1 to L do
  begin
    LIX:=LMX;
    LMX:=trunc(LMX/2);
    ARG:=0.0;
    for LM:=1 to LMX do
    begin
      C:=cos(ARG);
      S:=sin(ARG);
      ARG:=ARG+SCL;
      LI:=LIX;
      while LI<= NP do
      begin
        J1:=LI-LIX+LM;
        J2:=J1+LMX;
        T1:=X[J1]-X[J2];
        T2:=Y[J1]-Y[J2];
        X[J1]:=X[J1]+X[J2];
        Y[J1]:=Y[J1]+Y[J2];
        X[J2]:=C*T1+S*T2;
        Y[J2]:=C*T2-S*T1;
        LI:=LI+LIX;
      end;
    end;
    SCL:=2.0*SCL;
  end;

  {BIT REVERSAL}
  J:=1;
  NV2:=trunc(NP/2);
  NPM1:=NP-1;
  for I:=1 to NPM1 do
  begin
    if I>=J then goto LABEL1 else
```

```
      begin
      T1:=X[J];
      T2:=Y[J];
      X[J]:=X[I];
      Y[J]:=Y[I];
      X[I]:=T1;
      Y[I]:=T2;
      end;
      LABEL1:K:=NV2;
      LABEL2:if K>=J then goto LABEL3 else
      begin
      J:=J-K;
      K:=trunc(K/2);
      end;
      goto LABEL2;
      LABEL3:J:=J+K;
    end;
end;

procedure OVERLAY2;
begin
end;

{PROCEDURE FOR CALCULATE ERROR SIGNAL OF SIFT STEP1}

OVERLAY procedure SIFTSTEP1(spch:ARR400;var pbuf:ARR80);
label LABEL20,LABEL30;
const AF:ARR5=(1.0,-2.34036589,2.01190019,-0.614109218,0.0);
      PF:ARR5=(0.0357081667,-0.0069956244,-0.0069956244,0.0357081667,0.0)
      P:ARR5=(1.0,0.0,0.0,0.0,0.0);
var DF,D,A,RC:ARR5;
    AL,SOUT,UPREV,FOUT:real;
    K,J:integer;
    U:ARR80;
begin
  for J:=1 to 5 do
  begin
    DF[J]:=0.0;
    D[J]:=0.0;
  end;
  UPREV:=0.0;
  for J:=1 to window_size do
  begin
    DIRECT(af,pf,3,df,spch[j],sout);
    if (J mod down_sampl)<>0 then goto LABEL20;
    K:=trunc(J/down_sampl);
    PBUF[K]:=SOUT;
    U[K]:=(SOUT-UPREV)*(0.54-0.46*cos((K-1)*
          6.28318/(trunc(window_size/down_sampl)-1)));
    UPREV:=SOUT;
    LABEL20:;
```

```
    end;
    AUTOSIFT(trunc(window_size/down_sampl),4,al,a,rc,u);
    for J:=1 to trunc(window_size/down_sampl) do
    begin
      DIRECT(p,a,4,d,pbuf[j],fout);
      if J<=4 then goto LABEL30;
      PBUF[J-4]:=FOUT*(0.54-0.46*cos((J-5)*6.28318/
                       (trunc(window_size/down_sampl)-5)));
      LABEL30:;
    end;
  end;

  {DEFINE HEADER OF WINDOW}

  OVERLAY procedure DefineHeader_Test;
  begin
    Pompt('WHAT HEADER DO YOU WANT TO DEFINE?');
    read(kbd,head);
    Pompt(concat('HEAD IS ','"',head,'"'));
    readln(kbd);
    Pompt('WHAT WINDOW DO YOU WANT TO SELECT?');
    read(kbd,wind);
    Str(wind,wrkst2);
    Pompt(concat('WINDOW No.',wrkst2));
    readln(kbd);
    SelectWindow(wind);
    DefineHeader(wind,head);
    SetHeaderOn;
    DrawBorder;
    head:=' ';
  end;

  {CHOICE FOR PLOT DATA}

  OVERLAY procedure Plot_Test;
  label LABEL1,LABEL2;
  var XLow,XHi,YLow,YHi:real;
  begin
    Pompt('WHAT WINDOW DO YOU WANT TO PLOT?');
    read(kbd,wind);
    Str(wind,wrkst2);
    Pompt(concat('PLOT AT WINDOW No. ',wrkst2));
    readln(kbd);
    Pompt('START AT SAMPLING No.?');
    read(kbd,Start_no);
    Str(Start_no,wrkst2);
    Pompt(concat('START AT SAMPLING No. ',wrkst2));
    readln(kbd);
    Pompt('No.OF SAMPLING TO PLOT?');
    read(kbd,no_to_plot);
    Str(no_to_plot,wrkst2);
```

```
    Pompt(concat('PLOT AT ',wrkst2,' SAMPLING'));
    readln(kbd);
    Data_PLOT;
    Pompt('DO YOU WANT AUTO PLOT?(Y/N)');
    repeat
      read(kbd,ch);
    until ch in ['n','y'];
    if ch='y' then goto LABEL1;
    Pompt(' Y MINIMUM?');
    read(kbd,YLow);
    Pompt(' Y MAXIMUM?');
    read(kbd,YHi);
    SelectWindow(wind);
    XLow:=start_no/1.0;
    XHi:=(start_no+no_to_plot-1)/1.0;
    defineworld(1,XLow,YLow,XHi,YHi);
    selectworld(1);
    goto LABEL2;
  LABEL1:;
    SelectWindow(wind);
    FindWorld(1,a_plot,no_to_plot,1,1);
  LABEL2:;
    str(step:9,wrkst3);
    head:=filename;
    DefineHeader(wind,concat(head,': (STEP =',wrkst3,'MSEC)'));
    head:=' ';
    SetHeaderOn;
    DrawBorder;
    SetLineStyle(0);
    DrawAxis(-8,7,0,0,0,0,0,1,true);
    DrawPolygon(a_plot,1,no_to_plot,code,2,0);
    RedefineWindow_Test;
  end;

  OVERLAY procedure ClearWindow_Test;
  begin
    Pompt('WHAT WINDOW DO YOU WANT TO CLEAR?');
    read(kbd,wind);
    Str(wind,wrkst2);
    Pompt(concat('CLEAR WINDOW No. ',wrkst2));
    readln(kbd);
    SelectWindow(wind);
    SetBackGround(0);
    DrawBorder;
  end;

  {PROCEDURE FOR MOVE WINDOW}

  OVERLAY procedure MoveWindow_Test;
  begin
    Pompt('WHAT WINDOW DO YOU WANT TO MOVE?');
```

```
  read(kbd,wind);
  Str(wind,wrkst2);
  Pompt(concat('MOVE WINDOW No. ',wrkst2));
     CopyScreen;
     SetBreakOff;
     SetMessageOff;
     SelectWindow(wind);
     Repeat
       read(kbd,ch);
       case ord(ch) of
         72:MoveVer(-4,true);
         75:MoveHor(-1,true);
         77:MoveHor(1,true);
         80:MoveVer(4,true);
       end;
     until ch=' ';
  RedefineWindow_Test;
end;

{PROCEDURE FOR CLEAR SCREEN}

OVERLAY procedure ClearScreen_Test;
begin
  ClearScreen;
end;

{PROCEDURE FOR COPY SCREEN TO PRINTER}

OVERLAY procedure HardCopy_Test;
begin
  HardCopy(false,6);
end;

{PROCEDURE FOR DEFINE POINT OF GRAPH}

OVERLAY procedure Define_point;
begin
  Pompt('0=DRAW LINE , 9=DON''T DRAW LINE');
  read(kbd,ch);
  if ch = '0' then code:=0;
  if ch = '9' then code:=9;
end;

{FIND PEAK OF OUTPUT OF FFT.}

OVERLAY procedure FINDPK (var f:FF;npts:integer;fsk:real;
                          var npeak:integer;var ix:IXIX);
label LABEL1,LABEL2,LABEL3,LABEL4;
var NPTS2,J00,NF,KCD,K,J,NTS2,NPTM1:integer;
    TS,TSS,YM,YP,FN,FJ,FJM1,FJP1,AA,BB,CC,PK,DD,GG,BW,FI:real;
begin
```

```
JOO:=0;
NPEAK:=0;
NPTS2:=NPTS*2;
NPTM1:=NPTS-1;
TS:=FSK/NPTS2;
TSS:=TS*TS;
NF:=0;
YM:=F[2];
YP:=F[3];
if YP>YM then goto LABEL1;
KCD:=1;
YM:=YP;
goto LABEL2;
LABEL1:KCD:=2;YM:=YP;
LABEL2:for K:=4 to NPTM1 do
       begin
       YP:=F[K];
       if KCD=2 then
       begin
       goto LABEL3;
       end;
       if YP<=YM then
      begin
       goto LABEL4;
       end;
       KCD:=2;
       NF:=NF+1;
       J:=K-1;
       FN:=(J-1)*TS;
       FJ:=1.0/F[J];
       FJM1:=1.0/F[J-1];
       FJP1:=1.0/F[J+1];
       AA:=FJM1-2.0*FJ+FJP1;
       AA:=AA/(2.0*TSS);
       BB:=(FJP1-FJM1)/(2.0*TS);
       CC:=FJ;
       PK:=CC-BB*BB/(4.0*AA);
       DD:=CC-PK/2.0;
       GG:=BB*BB-4.0*AA*DD;
       BW:=-sqrt(GG)/AA;
       FI:=-BB/(2.0*AA)+FN;
       IX[JOO+1]:=FI;
       IX[JOO+2]:=BW;
       IX[JOO+3]:=434.0*ln(PK);
       JOO:=JOO+3;
       NPEAK:=NPEAK+1;
       goto LABEL4;
       LABEL3:if YP<YM then KCD:=1 else goto LABEL4;
       LABEL4:YM:=YP;
       end;
end;
```

```
{AUTOCORRELATION FOR FIND OUT FORMANT}

OVERLAY procedure AUTO_FORMNT(n,m:integer;var alpha:real;
                              var a:AA;var rc:RCRC;var x:XX);
var R : array[1..21] of real;
    B : array[1..21] of real;
    IP,JB,NPK,MINC,MF,MP,K,NP,NK,J,MIP : integer;
    S:real;
begin
  for J:=1 to 21 do R[J]:=0.0;
  for J:=1 to 21 do B[J]:=0.0;
  MP:=M+1;
  for K:=1 to MP do
  begin
    R[K]:=0.0;
    NK:=N-K+1;
    NP:=1;
    for NP := 1 to NK do
    begin
      NPK:=NP+K-1;
      R[K]:=R[K]+X[NP]*X[NPK];
    end;
  end;
  ALPHA:=R[1];
  RC[1]:=(-R[2])/R[1];
  A[1]:=1;
  A[2]:=RC[1];
  ALPHA:=ALPHA-RC[1]*RC[1]*ALPHA;
  MF:=M;
  for MINC:=2 to MF do
  begin
    M:=MINC-1;
    for J:=1 to MINC do
    begin
      JB:=MINC-J+1;
      B[J]:=A[JB];
    end;
    M:=M+1;
    S:=0.0;
    for IP:=1 to M do
    begin
      MIP:=M-IP+2;
      S:=S+R[MIP]*A[IP];
    end;
    RC[M]:=-S/ALPHA;
    for IP:=2 to M do
    begin
      A[IP]:=A[IP]+RC[M]*B[IP-1];
    end;
```

```
      A[M+1]:=RC[M];
      ALPHA:=ALPHA-RC[M]*RC[M]*ALPHA;
      if ALPHA<=0 then writeln('ALPHA ERROR');
  end;
  end;

  {FIND OUT SPRECTRUM OF SIGNAL}

  OVERLAY procedure SPECTRUM(ip,n,m:integer;var x:XX;var a:AA;var f:FF
  var J:integer;
      B,R:array[1..21] of real;
  begin
    for J:=1 to 21 do R[J]:=0.0;
    for J:=1 to 21 do B[J]:=0.0;
    I:=round(exp(ip*ln(2)));
    for J:=1 to I do
    begin
      Y[J]:=0.0;
    end;
    for J:=1 to I do
    begin
      X[J]:=0.0;
    end;
    for J:=1 to  M+1 do
    begin
      X[J]:=A[J];
    end;
    FFT(x,y,ip);
    for J:=1 to trunc(I/2)+1 do
    begin
      F[J]:=(X[J]*X[J])+(Y[J]*Y[J]);
    end;
  end;

  {FOR FIND OUT FORMANT}

  OVERLAY procedure FORMNT_TEST(var ix:IXIX;var iff,if1:ARR3;
                                iuv,iuv1,npeak:integer);
  label LABEL1,LABEL10,LABEL20,LABEL30,LABEL50,LABEL70,LABEL80,
        LABEL90,LABEL100,LABEL110,LABEL120,LABEL130,LABEL140,
        LABEL141,LABEL150,LABEL160,LABEL170,LABEL180,LABEL190,LABEL200
  var I,J,IHLF,ICD,L:integer;
      IP:ARR4;

  function IV(i,j:integer;ip:ARR4;if1:ARR3):integer;
  begin
    IV:=trunc(abs(IP[I]-IF1[J]));
  end;

  begin
    for j:=1 to 4 do IP[j]:=0.0;
```

```
  if NPEAK=0 then goto LABEL1;
  IHLF:=500;
  if IUV1=1 then goto LABEL10;
LABEL1:;
  IFF[1]:=0.0;
  IFF[2]:=0.0;
  IFF[3]:=0.0;
  goto LABEL200;
LABEL10:;
  L:=0;
  for J:=1 to NPEAK do
  begin
    I:=3*J-2;
    if IX[I]>3000.0 then goto LABEL20;
    if IX[i]<=0.0 then goto LABEL20;
    if IX[I+1]>500.0 then goto LABEL20;
    L:=L+1;
    IP[L]:=IX[I];
    LABEL20:;
  end;
  if L=3 then goto LABEL30;
  if L=1 then goto LABEL50;
  goto LABEL90;
LABEL30:;
  for J:=1 to 3 do
  begin
    IFF[J]:=IP[J];
  end;
  goto LABEL200;
LABEL50:;
  for J:=1 to 3 do
  begin
    IFF[J]:=IF1[J];
  end;
  if IV(1,1,ip,if1)>IV(1,2,ip,if1) then goto LABEL70;
  IFF[1]:=IP[1];
  goto LABEL200;
LABEL70:;
  if IV(1,3,ip,if1)>IV(1,2,ip,if1) then goto LABEL80;
  IFF[3]:=IP[1];
  goto LABEL200;
LABEL80:;
  IFF[2]:=IP[1];
  goto LABEL200;
LABEL90:;
  if IV(1,1,ip,if1)<IV(1,2,ip,if1) then goto LABEL110;
  if (IP[2]-IP[1])<=IHLF then goto LABEL100;
  ICD:=2;
  IFF[2]:=IP[1];
  IFF[1]:=IF1[1];
  goto LABEL160;
```

```
LABEL100:;
  IFF[1]:=IP[1];
  IFF[2]:=IP[2];
  ICD:=3;
  goto LABEL160;
LABEL110:;
  IFF[1]:=IP[1];
  if L=2 then goto LABEL140;
  if IV(2,2,ip,if1)>IV(3,2,ip,if1) then goto LABEL120;
  if IV(2,2,ip,if1)<IV(2,3,ip,if1) then goto LABEL130;
  goto LABEL141;
LABEL120:;
  IFF[2]:=IP[3];
  IFF[3]:=IP[4];
  goto LABEL200;
LABEL130:;
  IFF[2]:=IP[2];
  ICD:=3;
  goto LABEL170;
LABEL140:;
  if IV(2,2,ip,if1)<IV(2,3,ip,if1) then goto LABEL150;
LABEL141:;
  IFF[3]:=IP[2];
  IFF[2]:=IF1[2];
  goto LABEL200;
LABEL150:;
  IFF[2]:=IP[2];
  IFF[3]:=IF1[3];
  goto LABEL200;
LABEL160:;
  if ICD>L then goto LABEL190;
  if ICD=L then goto LABEL180;
LABEL170:;
  if IV(ICD,3,ip,if1)<IV(ICD+1,3,ip,if1) then goto LABEL180;
  IFF[3]:=IP[ICD+1];
  goto LABEL200;
LABEL180:;
  IFF[3]:=IP[ICD];
  goto LABEL200;
LABEL190:;
  IFF[3]:=IF1[3];
LABEL200:;
  IF1[1]:=IFF[1];
  IF1[2]:=IFF[2];
  IF1[3]:=IFF[3];
end;

procedure OVERLAY3;
begin
end;
```

```
{FIND PEAK OF SIFT. IN STEP2}

OVERLAY procedure SIFTSTEP2(var pbuf:ARR80;var pitch:ARR5;
                            var pitchactual:real;ii:integer;fs:intege
label LABEL30,LABEL40,LABEL50,LABEL60,LABEL70,LABEL80,LABEL90,LABEL10
var J,JJ,NMJ,I,IPJ,L:integer;
    SUM,P0,P1,P2,P3,AMAX,AA,BB,AP,AL,V,D:real;
    ABUF:ARR33;
begin
  for JJ:=1 to 33 do
  begin
    J:=JJ-1;
    NMJ:=(trunc(window_size/down_sampl)-4)-J;
    SUM:=0.0;
    for I:=1 to NMJ do
    begin
      IPJ:=I+J;
      SUM:=SUM+PBUF[I]*PBUF[IPJ];
    end;
    ABUF[JJ]:=SUM;
  end;
{  RESETOUTFILE(energyfile_name);
  seek(outfile,FileSize(outfile));
  for j:=1 to 33 do
  begin
  write(outfile,abuf[j]);
  end;
  close(outfile);}
  P1:=PITCH[1];
  P2:=PITCH[2];
  P3:=PITCH[3];
  L:=6;
  AMAX:=ABUF[L];
  for J:=6 to 32 do
  begin
    if ABUF[J]<=AMAX then goto LABEL30;
    AMAX:=ABUF[J];
    L:=J;
    LABEL30:;
  end;
  writeln('amax=',amax,'   L=',1);
{  writeln(lst,'amax=',amax,'   L=',1); }
  if AMAX=0 then goto LABEL60;
  if ABUF[L]<ABUF[L-1] then goto LABEL60;
  AA:=ABUF[L-1]-ABUF[L];
  AA:=(AA+ABUF[L+1]-ABUF[L])/2.0;
  BB:=(ABUF[L+1]-ABUF[L-1])/4.0;
  AP:=ABUF[L]-BB*BB/AA;
  AL:=L-BB/AA;
  V:=AP/ABUF[1];
  if L>=19 then goto LABEL40;
```

```
    D:=-1.0*(L-6)/13.0+2.0;
    goto LABEL50;
    LABEL40:;
    D:=-0.1*(L-19)/13.0+1.0;
    LABEL50:;
    V:=V/D;
    writeln('v=',v);
{   writeln(lst,'v=',v); }
    {if V>=0.3 then goto LABEL70;}
    {if P1=0.0 then goto LABEL60; }
    {if V>=0.25 then goto LABEL70; }
    if V>=0.0 then goto LABEL70;
    if P1=0.0 then goto LABEL60;
    if V>=0.0 then goto LABEL70;
    LABEL60:;
    writeln('LABEL 60');
{   writeln(lst,'LABEL 60');}
    P0:=0.0;
    goto LABEL80;
    LABEL70:;
    writeln('LABEL 70');
{   writeln(lst,'LABEL 70'); }
    P0:=AL;
    LABEL80:;
    if abs(P1-P3)<=0.375*P3 then P2:=(P1+P3)/2.0;
    if P3<>0.0 then goto LABEL90;
    if P2=0.0 then goto LABEL90;
    if abs(P0-P1)>0.2*P1 then goto LABEL90;
    P2:=2.0*P1-P0;
    LABEL90:;
    if P1<>0.0 then goto LABEL100;
    if abs(P2-P3)>=(0.375*P3) then P2:=0.0;
    LAbel100:;
    PITCH[3]:=P2;
    PITCH[2]:=P1;
    PITCH[1]:=P0;
  writeln('pitch[1]=',pitch[1]);
  writeln('pitch[2]=',pitch[2]);
  writeln('pitch[3]=',pitch[3]);
    PITCHACTUAL:=(PITCH[3]-1)*II/FS;
  writeln('pitch actaul=',pitchactual);
  {writeln(lst,'pitch[1]=',pitch[1]);
  writeln(lst,'pitch[2]=',pitch[2]);
  writeln(lst,'pitch[3]=',pitch[3]);
  writeln(lst,'pitch actaul=',pitchactual);
  writeln(lst);}
  end;


  procedure OVERLAY4;
  begin
```

```
end;

{CHOICE FOR FIND OUT FORMANT}

OVERLAY procedure FORMNT;
label LABEL1,LABEL2;
begin
  nosound;
  for J:=1 to 3 do IF1[j]:=0.0;
  write('WHAT FORMANT1 FILE NAME?');
  readln(formnt1file_name);
  CREATEFILE(formnt1file_name);
  STEP:=(window_freq/fs);
  RESETOUTFILE(formnt1file_name);
  seek(outfile,FileSize(outfile));
  write(outfile,step);
  close(outfile);
  write('WHAT FORMANT2 FILE NAME?');
  readln(formnt2file_name);
  CREATEFILE(formnt2file_name);
  STEP:=(window_freq/fs);
  RESETOUTFILE(formnt2file_name);
  seek(outfile,FileSize(outfile));
  write(outfile,step);
  close(outfile);
  write('WHAT FORMANT3 FILE NAME?');
  readln(formnt3file_name);
  CREATEFILE(formnt3file_name);
  STEP:=(window_freq/fs);
  RESETOUTFILE(formnt3file_name);
  seek(outfile,FileSize(outfile));
  write(outfile,step);
  close(outfile);
  write('HOW MANY ORDER?');
  readln(M);
  write('HOW MANY POWER2?(1-10)');
  readln(power2);
  RESETINFILE(datafile_name);
  size_of_file:=filesize(infile);
  seek(infile,1);
  flag1:=1;
LABEL1:;
  for J:=1 to window_size do x[j]:=0.0;
  J:=1;
  full:=false;
  while not EOF(infile) and not full do
  begin
    read(infile,x[j]);
    full:=true;
    if J<window_size then full:=false;
    j:=j+1;
```

```
    end;

  {PREEMPHASIS}
    for J:=window_size downto 2 do
    begin
      x[j]:=x[j]-x[j-1];
    end;
    N:=window_size;
    for J:=1 to 21 do A[J]:=0.0;
    for J:=1 to 21 do RC[J]:=0.0;
    for J:=1 to 600 do Y[J]:=0.0;
    alpha:=0.0;
    Auto_formnt(n,m,alpha,a,rc,x);
    MP:=M+1;
    IP:=power2;
    for J:=1 to 600 do F[J]:=0.0;
    SPECTRUM(ip,n,m,x,a,f);
    for J:=1 to 600 do IX[J]:=0.0;
    NPTS:=trunc(round(exp(ip*ln(2)))/2);FSK:=fs*1000.0;
    FINDPK(f,npts,fsk,npeak,ix);
    FORMNT_TEST(ix,iff,if1,1,1,npeak);
    for J:=1 to 3 do writeln('FORMANT',J,'=',iff[j]);
    RESETOUTFILE(formnt1file_name);
    seek(outfile,FileSize(outfile));
    write(outfile,iff[1]);
    close(outfile);
    RESETOUTFILE(formnt2file_name);
    seek(outfile,FileSize(outfile));
    write(outfile,iff[2]);
    close(outfile);
    RESETOUTFILE(formnt3file_name);
    seek(outfile,FileSize(outfile));
    write(outfile,iff[3]);
    close(outfile);
    flag1:=flag1+window_freq;
    if flag1>=size_of_file then goto LABEL2;
    close(infile);
    resetinfile(datafile_name);
    seek(infile,flag1);
    goto LABEL1;
  LABEL2:;
    close(infile);
  end;

  {FIND OUT ENERGY SQUARE}

  OVERLAY procedure ENERGYSQUARE;
  begin
    nosound;
    write('WHAT ENERGYSQUARE FILE NAME?');
    readln(energyfile_name);
```

```
  CREATEFILE(energyfile_name);
  RESETINFILE(datafile_name);
  seek(infile,1);
  RESETOUTFILE(energyfile_name);
  step:=1.0/fs;
  write(outfile,step);
  while not EOF(infile) do
  begin
    read(infile,buffer1);
    buffer1:=buffer1*buffer1;
    write(outfile,buffer1);
  end;
  close(outfile);
  close(infile);
end;

{FIND OUT ENERGY ALPHA}

OVERLAY procedure ENERGYALPHA;
type record_of_RC=record
                    RC1 :real;
                    RC2 :real;
                    RC3 :real;
                    RC4 :real;
                    RC5 :real;
                    RC6 :real;
                    RC7 :real;
                    RC8 :real;
                    RC9 :real;
                    RC10:real;
                    RC11:real;
                    RC12:real;
                    RC13:real;
                    RC14:real;
                    RC15:real;
                    end;
      fileofrc=file of record_of_rc;

var  tempfileofrc,infileofrc,outfileofrc:fileofrc;
     temprecordofrc:record_of_rc;
     filenameofrc:str12;
     i:integer;

label LABEL1,LABEL2;

function CREATEFILE_RC(var Fi:FILEOFRC):Boolean;
begin
  {$I-} reset(Fi); {$I+}
  createfile_rc:=IOresult=0;
end;
```

```
procedure CREATEFILEOFRC(var filenameofrc:STR12);
label LABEL1;
begin
LABEL1:;
  assign(tempfileofrc,filenameofrc);
  if not CREATEFILE_RC(tempfileofrc) then
  begin
    rewrite(tempfileofrc);
    close(tempfileofrc);
  end
  else
    begin
      writeln('***OLD DATA FILE***');
      write('DO YOU WANT TO WRITE OVER IT?(Y/N)');
      repeat
      readln(ch);
      until ch in ['n','y'];
      if ch='y' then
      begin
        rewrite(tempfileofrc);close(tempfileofrc);
      end;
      if ch='n' then
      begin
        write('WHAT NEW FILENAME?');
        readln(filenameofrc);
        goto LABEL1;
      end;
    end;
end;

procedure RESETINFILEOFRC(filenameofrc:STR12);
begin
  assign(infileofrc,filenameofrc);
  if CREATEFILE_RC(infileofrc) then
  begin
    reset(infileofrc);
  end
  else
    writeln('NO FILE',filename);
end;

procedure RESETOUTFILEOFRC(filenameofrc:STR12);
begin
  assign(outfileofrc,filenameofrc);
  if CREATEFILE_RC(outfileofrc) then
  begin
    reset(outfileofrc);
  end
  else
    writeln('NO FILE',filename);
end;
```

```
begin
  nosound;
  write('WHAT ENERGYALPHA FILE NAME?');
  readln(energyfile_name);
  write('WHAT RC. FILE NAME?');
  readln(filenameofrc);
  createfile(energyfile_name);
  STEP:=(window_freq/fs);
  RESETOUTFILE(energyfile_name);
  seek(outfile,FileSize(outfile));
  write(outfile,step);
  close(outfile);
  createfileofrc(filenameofrc);
  RESETOUTFILEOFRC(filenameofrc);
  seek(outfileofrc,FileSize(outfileofrc));
  temprecordofrc.rc1:=step;
  temprecordofrc.rc2:=window_size;
  temprecordofrc.rc3:=window_freq;
  write(outfileofrc,temprecordofrc);
  close(outfileofrc);
  write('HOW MANY ORDER?');
  readln(M);
  RESETINFILE(datafile_name);
  size_of_file:=filesize(infile);
  seek(infile,1);
  flag1:=1;
LABEL1:;
  for j:=1 to window_size do x[j]:=0.0;
  J:=1;
  full:=false;
  while not EOF(infile) and not full do
  begin
    read(infile,x[j]);
    full:=true;
    if J<window_size then full:=false;
    j:=j+1;
  end;
  N:=window_size;
  for J:=1 to 21 do A[J]:=0.0;
  for J:=1 to 21 do RC[J]:=0.0;
  for J:=1 to 600 do Y[J]:=0.0;
  alpha:=0.0;
  auto_formnt(n,m,alpha,a,rc,x);
  writeln('ALPHA=',alpha);
  RESETOUTFILE(energyfile_name);
  seek(outfile,FileSize(outfile));
  write(outfile,alpha);
  close(outfile);

  resetoutfileofrc(filenameofrc);
```

```
  seek(outfileofrc,FileSize(outfileofrc));
  temprecordofrc.rc1:=rc[1];
  temprecordofrc.rc2:=rc[2];
  temprecordofrc.rc3:=rc[3];
  temprecordofrc.rc4:=rc[4];
  temprecordofrc.rc5:=rc[5];
  temprecordofrc.rc6:=rc[6];
  temprecordofrc.rc7:=rc[7];
  temprecordofrc.rc8:=rc[8];
  temprecordofrc.rc9:=rc[9];
  temprecordofrc.rc10:=rc[10];
  temprecordofrc.rc11:=rc[11];
  temprecordofrc.rc12:=rc[12];
  temprecordofrc.rc13:=rc[13];
  temprecordofrc.rc14:=rc[14];
  temprecordofrc.rc15:=rc[15];
  write(outfileofrc,temprecordofrc);
  close(outfileofrc);
  for i:=1 to 15 do
    begin
      {writeln('RC ',i,' =',rc[i]);}
      if rc[i]<-1 then sound(500);{ writeln(lst,'ERROR RC ',i,' =',rc[i]);
      if rc[i]>1 then sound(500);{writeln(lst,'ERROR RC ',i,' =',rc[i]); }
    end;

  flag1:=flag1+window_freq;
  if flag1>=size_of_file then goto LABEL2;
  close(infile);
  resetinfile(datafile_name);
  seek(infile,flag1);
  goto LABEL1;
LABEL2:;
  close(infile);
end;

{CREATE PATTERN OF ENERGY ALPHA}

OVERLAY procedure ENERGYALPHA_RC_PATTERN(var filenameofrc,
                                         filenameofalpha,
                                         filenameofdata:str12);

type record_of_RC=record
                    RC1 :real;
                    RC2 :real;
                    RC3 :real;
                    RC4 :real;
                    RC5 :real;
                    RC6 :real;
                    RC7 :real;
                    RC8 :real;
                    RC9 :real;
                    RC10:real;
```

```
                    RC11:real;
                    RC12:real;
                    RC13:real;
                    RC14:real;
                    RC15:real;
                    end;
        fileofrc=file of record_of_rc;

var  tempfileofrc,infileofrc,outfileofrc:fileofrc;
     temprecordofrc:record of rc;
     i:integer;
label LABEL1,LABEL2;
function CREATEFILE_RC(var Fi:FILEOFRC):Boolean;
begin
  {$I-} reset(Fi); {$I+}
  createfile_rc:=IOresult=0;
end;

procedure CREATEFILEOFRC(var filenameofrc:STR12);
label LABEL1;
begin
LABEL1:;
  assign(tempfileofrc,filenameofrc);
  if not CREATEFILE_RC(tempfileofrc) then
  begin
    rewrite(tempfileofrc);
    close(tempfileofrc);
  end
  else
    begin
      writeln('***OLD DATA FILE***');
      write('DO YOU WANT TO WRITE OVER IT?(Y/N)');
      repeat
      readln(ch);
      until ch in ['n','y'];
      if ch='y' then
      begin
        rewrite(tempfileofrc);close(tempfileofrc);
      end;
      if ch='n' then
      begin
        write('WHAT NEW FILENAME?');
        readln(filenameofrc);
        goto LABEL1;
      end;
    end;
end;

procedure RESETINFILEOFRC(filenameofrc:STR12);
begin
  assign(infileofrc,filenameofrc);
```

```
  if CREATEFILE_RC(infileofrc) then
  begin
    reset(infileofrc);
  end
  else
    writeln('NO FILE',filename);
end;

procedure RESETOUTFILEOFRC(filenameofrc:STR12);
begin
  assign(outfileofrc,filenameofrc);
  if CREATEFILE_RC(outfileofrc) then
  begin
    reset(outfileofrc);
  end
  else
    writeln('NO FILE',filename);
end;

begin
  nosound;
  window_size:=250;
  window_freq:=50;
  energyfile_name:=filenameofalpha;
  filenameofrc:=filenameofrc;
  m:=15;
  datafile_name:=filenameofdata;
  RESETINFILE(datafile_name);
  size_of_file:=filesize(infile);
  seek(infile,1);
  flag1:=1;
LABEL1:;
  for j:=1 to window_size do x[j]:=0.0;
  J:=1;
  full:=false;
  while not EOF(infile) and not full do
  begin
    read(infile,x[j]);
    full:=true;
    if J<window_size then full:=false;
    j:=j+1;
  end;
  N:=window_size;
  for J:=1 to 21 do A[J]:=0.0;
  for J:=1 to 21 do RC[J]:=0.0;
  for J:=1 to 600 do Y[J]:=0.0;
  alpha:=0.0;
  auto_formnt(n,m,alpha,a,rc,x);
  writeln('ALPHA=',alpha);
  RESETOUTFILE(energyfile_name);
  seek(outfile,FileSize(outfile));
```

```
    write(outfile,alpha);
    close(outfile);
    resetoutfileofrc(filenameofrc);
    seek(outfileofrc,FileSize(outfileofrc));
    temprecordofrc.rc1:=rc[1];
    temprecordofrc.rc2:=rc[2];
    temprecordofrc.rc3:=rc[3];
    temprecordofrc.rc4:=rc[4];
    temprecordofrc.rc5:=rc[5];
    temprecordofrc.rc6:=rc[6];
    temprecordofrc.rc7:=rc[7];
    temprecordofrc.rc8:=rc[8];
    temprecordofrc.rc9:=rc[9];
    temprecordofrc.rc10:=rc[10];
    temprecordofrc.rc11:=rc[11];
    temprecordofrc.rc12:=rc[12];
    temprecordofrc.rc13:=rc[13];
    temprecordofrc.rc14:=rc[14];
    temprecordofrc.rc15:=rc[15];
    write(outfileofrc,temprecordofrc);
    close(outfileofrc);
    for i:=1 to 15 do
      begin
        {writeln('RC ',i,' =',rc[i]);}
        if rc[i]<-1 then sound(500);{ writeln(lst,'ERROR RC ',i,' =',rc[i]);
        if rc[i]>1 then sound(500);{writeln(lst,'ERROR RC ',i,' =',rc[i]);
      end;
    flag1:=flag1+window_freq;
    if flag1>=size_of_file then goto LABEL2;
    close(infile);
    resetinfile(datafile_name);
    seek(infile,flag1);
    goto LABEL1;
  LABEL2:;
    close(infile);
  end;

  {SET VALUE FOR ANALYSIS}

  OVERLAY procedure INFORMATION;
  begin
    nosound;
    write('WHAT WINDOW FREQUENCY?');
    readln(window_freq);
    write('WHAT DATA FILE NAME?');
    readln(datafile_name);
    write('HOW MANY WINDOW SIZE?');
    readln(window_size);
    write('HOW MANY SAMPLING FREQUENCY(IN KHz.)?');
    readln(fs);
  end;
```

```
{TRANSFORM DATA FROM BYTE TO REAL}

OVERLAY procedure TRANSFORM_DATA;
label LABEL1;
var flag:boolean;
begin
  nosound;
  flag:=false;
  write('WHAT DATA FILE NAME TO TRANSFORM TO REAL?');
  readln(filename);
  assign(speechfile,filename);
  write('WHAT DATA FILE NAME OF REAL?');
  readln(filename);
  write('MINUS BY ?');
  readln(minus_value);
  writeln('DO YOU WANT TO SHOW DATA AT SCREEN?(Y/N)');
  repeat
    read(kbd,ch);
  until ch in ['n','y'];
  if ch='y' then flag:=true;
  write('PLEASE WAIT MANY MANY MINITE!!!!!');
  createfile(filename);
  resetoutfile(filename);
  step:=1/fs;
  write(outfile,step);
  reset(speechfile);
  seek(speechfile,20);
  while not EOF(speechfile) do
  begin
    read(speechfile,bytetemp);
    realtemp:=bytetemp;
    realtemp:=realtemp-minus_value;
    if flag then write(realtemp);
    write(outfile,realtemp);
  end;
  close(outfile);
  close(speechfile);
end;

{CHOICE FOR HARDCOPY IN GRAPHIC}

OVERLAY procedure HARDCOPY_SPEECH_SIGNAL;
label LABEL1,LABEL2;
var m,n:integer;
    temp,YMax,YMin:real;
    flag:boolean;
begin
  nosound;
  flag:=true;
  write('WHAT FILE DO YOU WANT TO PLOT?');
```

```
    readln(filename);
    write('HOW MANY SAMPLING PER FRAME?');
    readln(no_to_plot);
    writeln('DO YOU WANT ALIGNMENT PAPER?(Y/N)');
    repeat
      read(kbd,ch);
    until ch in ['n','y'];
    if ch='y' then flag:=false;
    write('PLEASE WAIT MANY MANY MINITE!!!!!!');
    RESETINFILE(filename);
    read(infile,YMax);
    YMin:=YMax;
    While not EOF(infile) do
    begin
      read(infile,temp);
      if temp>YMax then YMax:=temp
      else if temp<Ymin then YMin:=temp;
    end;
    close(infile);
    RESETINFILE(filename);
    n:=filesize(infile);
    read(infile,step);
    str(step:9,wrkst1);
    InitGraphic;
    DefineWindow(9,0,0,XMaxGlb,YMaxGlb-IVStepGlb-1);
    DefineWindow(10,0,YMaxGlb-6,XMaxGlb,YMaxGlb);
    clearscreen;
    DrawBorder;
    for m:=1 to trunc(n/no_to_plot)+1 do
    begin
      for i:=1 to no_to_plot do
      begin
        if((m-1)*no_to_plot+i)>=n then
                                    begin
                                      for j:=i to no_to_plot do
                                      begin
                                        a_plot[j,1]:=(m-1)*no_to_plot+j;
                                        a_plot[j,2]:=a_plot[j-1,2];
                                      end;
                                      goto LABEL1;
                                    end;
        a_plot[i,1]:=(m-1)*no_to_plot+i;
        read(infile,a_plot[i,2]);
      end;
LABEL1:;
    str(m,wrkst2);
    str(no_to_plot,wrkst3);
    defineheader(9,concat(filename,' FRAME ',wrkst2,
                          '  NO. OF SAMPLING = ',wrkst3,
                          ' PER FRAME : STEP = ',wrkst1,'MSEC'));
    selectwindow(9);
```

```
  setheaderon;
  defineworld(1,(m-1)*no_to_plot+1,YMin,(m*no_to_plot),YMax);
  selectworld(1);
  selectwindow(9);
  DrawBorder;
  SetLineStyle(0);
  DrawAxis(-8,7,0,0,0,0,0,1,true);
  DrawPolygon(a_plot,1,no_to_plot,0,2,0);
  DefineWindow(9,0,0,XMaxG1b,YMaxG1b-IVStepG1b-1);
  DefineWindow(10,0,YMaxG1b-6,XMaxG1b,YMaxG1b);
  if flag then
          begin
            HardCopy_Test;
            goto LABEL2;
            end;
  Pompt('DO YOU WANT HARDCOPY?(Y/N)');
  repeat
    read(kbd,ch);
  until ch in ['n','y'];
  if ch='y' then HardCopy_Test;
LABEL2:;
  selectwindow(9);
  setbackground(0);
  selectwindow(10);
  setbackground(0);
  end;
 leavegraphic;
 close(infile);
 end;

{HAMMING WINDOW}

OVERLAY procedure FILTER_WITH_HAMMING_WINDOW;
const AF:ARR5=(1.0,-2.34036589,2.01190019,-0.614109218,0.0);
      PF:ARR5=(0.0357081667,-0.0069956244,-0.0069956244,0.0357081667,0.0)
      P:ARR5=(1.0,0.0,0.0,0.0,0.0);
var DF,D,A,RC:ARR5;
    AL,SOUT,UPREV,FOUT:real;
    J,K:integer;
    REALTEMP:ARR400;
label label1,label2;
begin
  nosound;
  for j:=1 to 5 do d[j]:=0.0;
  write('WHAT DATA FILE NAME TO BE FILTERED?');
  readln(temp1file_name);
  resetinfile(temp1file_name);
  size_of_file:=filesize(infile);
  write('WHAT FILE NAME AFTER FILTER?');
  readln(filename);
  write('PLEASE WAIT MANY MANY MINITE!!!!!');
```

```
    createfile(filename);
    resetoutfile(filename);
    read(infile,REALTEMP[1]);
    write(outfile,REALTEMP[1]);
    flag1:=1;
  LABEL1:;
    for J:=1 to window_size do REALTEMP[j]:=0.0;
    J:=1;
    full:=false;
    while not EOF(infile) and not full do
    begin
      read(infile,REALTEMP[j]);
      full:=true;
      if J<window_size then full:=false;
      j:=j+1;
    end;
      for k:=1 to window_size do
      begin
        DIRECT(af,pf,3,df,REALTEMP[k],sout);
        al:=sout*(0.54-0.46*cos((k-1)*6.28318/(window_size-1)));
        write(outfile,al);
        uprev:=sout;
      end;
    flag1:=flag1+window_size;
    if flag1>=size_of_file then goto label2;
    close(infile);
    resetinfile(temp1file_name);
    seek(infile,flag1);
    goto label1;
  label2:;
    close(outfile);
    close(infile);
  end;

  {RECTANGULAR WINDOW}

  OVERLAY procedure FILTER_WITH_RECTANGULAR_WINDOW;
  const AF:ARR5=(1.0,-2.34036589,2.01190019,-0.614109218,0.0);
        PF:ARR5=(0.0357081667,-0.0069956244,-0.0069956244,0.0357081667,0.0);
        P:ARR5=(1.0,0.0,0.0,0.0,0.0);
  var DF,D,A,RC:ARR5;
      AL,SOUT,UPREV,FOUT,REALTEMP:real;
      J,K:integer;
  begin
    nosound;
    for j:=1 to 5 do d[j]:=0.0;
    write('WHAT DATA FILE NAME TO BE FILTERED?');
    readln(temp1file_name);
    resetinfile(temp1file_name);
    size_of_file:=filesize(infile);
    write('WHAT FILE NAME AFTER FILTER?');
```

```
  readln(filename);
  write('PLEASE WAIT MANY MANY MINITE!!!!!');
  createfile(filename);
  resetoutfile(filename);
  read(infile,realtemp);
  write(outfile,realtemp);
  while not EOF(infile) do
  begin
    read(infile,realtemp);
    DIRECT(af,pf,3,df,realtemp,sout);
    write(outfile,sout);
  end;
  close(outfile);
  close(infile);
end;

{FIND OUT ROOT MEAN SQUARE ENERGY}

OVERLAY procedure RMS;
label LABEL1,LABEL2;
begin
  nosound;
  write('WHAT FILE NAME TO BE ROOT MEAN SQUARE?');
  readln(filename);
  resetinfile(filename);
  size_of_file:=filesize(infile);
  write('WHAT FILE NAME AFTER ROOT MEAN SQUARE?');
  readln(rmsfile_name);
  write('PLEASE WAIT MANY MANY MINITE!!!!!');
  createfile(rmsfile_name);
  resetoutfile(rmsfile_name);
  step:=window_freq/fs;
  write(outfile,step);
  close(outfile);
  seek(infile,1);
  flag1:=1;
LABEL1:;
  for j:=1 to window_size do x[j]:=0.0;
  j:=1;
  full:=false;
  while not EOF(infile) and not full do
  begin
    read(infile,x[j]);
    full:=true;
    if j<window_size then full:=false;
    j:=j+1;
  end;
  realtemp:=0.0;
  for j:=1 to window_size do
  begin
    x[j]:=x[j]*x[j];
```

```
    realtemp:=realtemp+x[j];
  end;
  realtemp:=sqrt(realtemp/window_size);
  resetoutfile(rmsfile_name);
  seek(outfile,filesize(outfile));
  write(outfile,realtemp);
{writeln('ENERGY ROOT MEAN SQUARE = ',realtemp);}
  close(outfile);
  flag1:=flag1+window_freq;
  if flag1>=size_of_file then goto LABEL2;
  close(infile);
  resetinfile(filename);
  seek(infile,flag1);
  goto LABEL1;
LABEL2:;
  close(infile);
end;

{PROCEDURE FOR SMOOTH DATA}

OVERLAY procedure SMOOTH;
begin
  nosound;
  write('WHAT FILE NAME TO BE SMOOTHED?');
  readln(temp1file_name);
  resetinfile(temp1file_name);
  write('WHAT FILE NAME AFTER SMOOTH?');
  readln(filename);
  write('PLEASE WAIT MANY MANY MINITE!!!!!');
  createfile(filename);
  resetoutfile(filename);
  read(infile,step);
  write(outfile,step);
  read(infile,x1);
  read(infile,x2);
  read(infile,x3);
  write(outfile,x1);
  realtemp:=(x1+x2+x3)/3;
  write(outfile,realtemp);
  while not EOF(infile) do
  begin
    x1:=x2;
    x2:=x3;
    read(infile,x3);
    realtemp:=(x1+x2+x3)/3;
    write(outfile,realtemp);
  end;
  write(outfile,x3);
  close(outfile);
  close(infile);
end;
```

```
{PROCEDURE FOR NORMALIZE DATA}

OVERLAY procedure NORMALIZE;
var temp,YMax,YMin,step:real;
begin
  nosound;
  write('WHAT ENERGY FILE NAME TO BE NORMALIZE?');
  readln(filename);
  assign(infile,filename);
  write('WHAT FILE NAME AFTER NORMALIZE?');
  readln(filename);
  write('PLEASE WAIT MANY MANY MINITE!!!!!');
  reset(infile);
  seek(infile,1);
  read(infile,temp);
  YMin:=temp;
  YMax:=temp;
  while not EOF(infile) do
  begin
    read(infile,temp);
    if temp>YMax then YMax:=temp;
  end;
  close(infile);
  reset(infile);
  createfile(filename);
  resetoutfile(filename);
  read(infile,step);
  write(outfile,step);
  while not EOF(infile) do
  begin
    read(infile,temp);
    temp:=temp/YMax;
    write(outfile,temp);
  end;
  close(outfile);
  close(infile);
end;
```

```
{PROCEDURE FOR SEGMENT SYLLABLE}

OVERLAY procedure segment;
type string3=string[3];
     record_for_segment=record
                        value:real;
                        position:integer;
                        boundary:char;
                        maxormin:string3;
                        number:integer;
                        left:char;
                        right:char;
                       end;
      fileofrecord=file of record_for_segment;
      FILEOFINTEGER=file of integer;

var tempfileofrecord,infileofrecord,outfileofrecord:fileofrecord;
    tempfileofinteger,infileofinteger,outfileofinteger:fileofinteger;
    temprecord:record_for_segment;
    filenameoftemp1,filenameoftemp2,filenameofrecord,filenameofinteger:str
    x1,x2,x3,step,YMinGlb,range1,range2,realtemp1,realtemp2,
    range_lo_everage,range_hi_everage:real;
    realtempmax,realtempmin,realtemp:real;
    number_of_boundary,integertemp1,integertemp2,integertemp3:integer;
    no_of_step,countmax,countmin,count1,count2,n,
    nn,j,k,number,size_of_file:integer;
    position,position1,position2,no_of_syllable,duration_value:integer;

label LABEL1,LABEL2,LABEL3,LABEL4,LABEL5,LABEL6,LABEL7,LABEL8,LABEL9,LABEL
      LABEL11,LABEL12,LABEL13,LABEL14,LABEL15,LABEL16,
      LABEL17,LABEL18,LABEL19,LABEL20,LABEL30;
      LABEL40,LABEL41,LABEL42,LABEL43,LABEL44,LABEL45;
      LABEL50,LABEL51,LABEL52,LABEL53,LABEL54,LABEL55;
      LABEL60,LABEL61,LABEL62,LABEL63,LABEL64,LABEL65;
      LABEL100,LABEL101,LABEL102,LABEL103;
      LABEL200,LABEL300,LABEL400,LABEL500;
      LABEL600,LABEL601;
      LABEL700,LABEL701,LABEL702,LABEL703;

function CREATE(var Fi:FILEOFRECORD):Boolean;
begin
  {$I-} reset(Fi); {$I+}
  create:=IOresult=0;
end;

procedure CREATEFILEOFRECORD(var filenameofrecord:STR12);
label LABEL1;
begin
LABEL1:;
  assign(tempfileofrecord,filenameofrecord);
```

```
  if not CREATE(tempfileofrecord) then
  begin
    rewrite(tempfileofrecord);
    close(tempfileofrecord);
  end
  else
    begin
      writeln('***OLD DATA FILE***');
      write('DO YOU WANT TO WRITE OVER IT?(Y/N)');
      repeat
      readln(ch);
      until ch in ['n','y'];
      if ch='y' then
      begin
        rewrite(tempfileofrecord);close(tempfileofrecord);
      end;
      if ch='n' then
      begin
        write('WHAT NEW FILENAME?');
        readln(filenameofrecord);
        goto LABEL1;
      end;
    end;
end;

procedure RESETINFILEOFRECORD(filenameofrecord:STR12);
begin
  assign(infileofrecord,filenameofrecord);
  if CREATE(infileofrecord) then
  begin
    reset(infileofrecord);
  end
  else
    writeln('NO FILE',filenameofrecord);
end;

procedure RESETOUTFILEOFRECORD(filenameofrecord:STR12);
begin
  assign(outfileofrecord,filenameofrecord);
  if CREATE(outfileofrecord) then
  begin
    reset(outfileofrecord);
  end
  else
    writeln('NO FILE',filenameofrecord);
end;

function CREATEINTEGER(var Fi:FILEOFINTEGER):Boolean;
begin
  {$I-} reset(Fi); {$I+}
```

```
  createinteger:=IOresult=0;
end;

procedure CREATEFILEOFINTEGER(var filenameofinteger:STR12);
label LABEL1;
begin
LABEL1:;
  assign(tempfileofinteger,filenameofinteger);
  if not CREATEINTEGER(tempfileofinteger) then
  begin
    rewrite(tempfileofinteger);
    close(tempfileofinteger);
  end
  else
    begin
      writeln('***OLD DATA FILE***');
      write('DO YOU WANT TO WRITE OVER IT?(Y/N)');
      repeat
      readln(ch);
      until ch in ['n','y'];
      if ch='y' then
      begin
        rewrite(tempfileofinteger);close(tempfileofinteger);
      end;
      if ch='n' then
      begin
        write('WHAT NEW FILENAME?');
        readln(filenameofinteger);
        goto LABEL1;
      end;
    end;
end;

procedure RESETINFILEOFINTEGER(filenameofinteger:STR12);
begin
  assign(infileofinteger,filenameofinteger);
  if CREATEINTEGER(infileofinteger) then
  begin
    reset(infileofinteger);
  end
  else
    writeln('NO FILE',filenameofinteger);
end;

procedure RESETOUTFILEOFINTEGER(filenameofinteger:STR12);
begin
  assign(outfileofinteger,filenameofinteger);
  if CREATEINTEGER(outfileofinteger) then
  begin
    reset(outfileofinteger);
  end
```

```
  else
    writeln('NO FILE',filenameofinteger);
end;

begin
  nosound;
  write('WHAT FILE DO YOU WANT TO SEGMENT?');
  readln(filename);
  resetinfile(filename);
  read(infile,step);
  write('WHAT PITCH NAME DO YOU WANT FOR SUPPORT SEGMENT?');
  readln(filenametemp1);
  filenameofrecord:='c:file';
  createfileofrecord(filenameofrecord);
  resetoutfileofrecord(filenameofrecord);
  temprecord.value:=step;
  write(outfileofrecord,temprecord);
  close(outfileofrecord);

{CREATE FILE YMAXFILE AND YMINFILE}

  no_of_step:=1;
  countmax:=0;
  countmin:=0;
  read(infile,x1);
  read(infile,x2);
  read(infile,x3);
  if x1>x2 then
                        begin
                          countmax:=countmax+1;
                          resetoutfileofrecord(filenameofrecord);
                          seek(outfileofrecord,filesize(outfileofrecord));
                          temprecord.value:=x1;
                          temprecord.position:=no_of_step;
                          temprecord.boundary:='n';
                          temprecord.maxormin:='max';
                          temprecord.number:=countmax;
                          temprecord.left:='n';
                          temprecord.right:='n';
                          write(outfileofrecord,temprecord);
                          close(outfileofrecord);
                        end;
  if x1<x2 then
                        begin
                          countmin:=countmin+1;
                          resetoutfileofrecord(filenameofrecord);
                          seek(outfileofrecord,filesize(outfileofrecord));
                          temprecord.value:=x1;
                          temprecord.position:=no_of_step;
                          temprecord.boundary:='n';
                          temprecord.maxormin:='min';
```

```
                        temprecord.number:=countmin;
                        temprecord.left:='n';
                        temprecord.right:='n';
                        write(outfileofrecord,temprecord);
                        close(outfileofrecord);
                      end;

no_of_step:=no_of_step+1;
if x2>x1 then if x2>x3 then
                      begin
                        countmax:=countmax+1;
                        resetoutfileofrecord(filenameofrecord);
                        seek(outfileofrecord,filesize(outfileofrecord));
                        temprecord.value:=x2;
                        temprecord.position:=no_of_step;
                        temprecord.boundary:='n';
                        temprecord.maxormin:='max';
                        temprecord.number:=countmax;
                        temprecord.left:='n';
                        temprecord.right:='n';
                        write(outfileofrecord,temprecord);
                        close(outfileofrecord);
                      end;
if x2=x3 then if x3<x2 then
                      begin
                        countmax:=countmax+1;
                        resetoutfileofrecord(filenameofrecord);
                        seek(outfileofrecord,filesize(outfileofrecord));
                        temprecord.value:=x2;
                        temprecord.position:=no_of_step;
                        temprecord.boundary:='n';
                        temprecord.maxormin:='max';
                        temprecord.number:=countmax;
                        temprecord.left:='n';
                        temprecord.right:='n';
                        write(outfileofrecord,temprecord);
                        close(outfileofrecord);
                      end;
if x2>x1 then if x3=x2 then
                      begin
                        countmax:=countmax+1;
                        resetoutfileofrecord(filenameofrecord);
                        seek(outfileofrecord,filesize(outfileofrecord));
                        temprecord.value:=x2;
                        temprecord.position:=no_of_step;
                        temprecord.boundary:='n';
                        temprecord.maxormin:='max';
                        temprecord.number:=countmax;
                        temprecord.left:='n';
                        temprecord.right:='n';
                        write(outfileofrecord,temprecord);
```

```
                    close(outfileofrecord);
                  end;
if x2<x1 then if x2<x3 then
                  begin
                    countmin:=countmin+1;
                    resetoutfileofrecord(filenameofrecord);
                    seek(outfileofrecord,filesize(outfileofrecord)
                    temprecord.value:=x2;
                    temprecord.position:=no_of_step;
                    temprecord.boundary:='n';
                    temprecord.maxormin:='min';
                    temprecord.number:=countmin;
                    temprecord.left:='n';
                    temprecord.right:='n';
                    write(outfileofrecord,temprecord);
                    close(outfileofrecord);
                  end;
if x2<x1 then if x2=x3 then
                  begin
                    countmin:=countmin+1;
                    resetoutfileofrecord(filenameofrecord);
                    seek(outfileofrecord,filesize(outfileofrecord)
                    temprecord.value:=x2;
                    temprecord.position:=no_of_step;
                    temprecord.boundary:='n';
                    temprecord.maxormin:='min';
                    temprecord.number:=countmin;
                    temprecord.left:='n';
                    temprecord.right:='n';
                    write(outfileofrecord,temprecord);
                    close(outfileofrecord);
                  end;
if x2=x1 then if x2<x3 then
                  begin
                    countmin:=countmin+1;
                    resetoutfileofrecord(filenameofrecord);
                    seek(outfileofrecord,filesize(outfileofrecord)
                    temprecord.value:=x2;
                    temprecord.position:=no_of_step;
                    temprecord.boundary:='n';
                    temprecord.maxormin:='min';
                    temprecord.number:=countmin;
                    temprecord.left:='n';
                    temprecord.right:='n';
                    write(outfileofrecord,temprecord);
                    close(outfileofrecord);
                  end;
while not EOF(infile) do
begin
  no_of_step:=no_of_step+1;
  x1:=x2;
```

```
x2:=x3;
read(infile,x3);
if x2>x1 then if x2>x3 then
                        begin
                          countmax:=countmax+1;
                          resetoutfileofrecord(filenameofrecord);
                          seek(outfileofrecord,filesize(outfileofrecord))
                          temprecord.value:=x2;
                          temprecord.position:=no_of_step;
                          temprecord.boundary:='n';
                          temprecord.maxormin:='max';
                          temprecord.number:=countmax;
                          temprecord.left:='n';
                          temprecord.right:='n';
                          write(outfileofrecord,temprecord);
                          close(outfileofrecord);
                        end;
if x2=x3 then if x3<x2 then
                        begin
                          countmax:=countmax+1;
                          resetoutfileofrecord(filenameofrecord);
                          seek(outfileofrecord,filesize(outfileofrecord))
                          temprecord.value:=x2;
                          temprecord.position:=no_of_step;
                          temprecord.boundary:='n';
                          temprecord.maxormin:='max';
                          temprecord.number:=countmax;
                          temprecord.left:='n';
                          temprecord.right:='n';
                          write(outfileofrecord,temprecord);
                          close(outfileofrecord);
                        end;
if x2>x1 then if x3=x2 then
                        begin
                          countmax:=countmax+1;
                          resetoutfileofrecord(filenameofrecord);
                          seek(outfileofrecord,filesize(outfileofrecord))
                          temprecord.value:=x2;
                          temprecord.position:=no_of_step;
                          temprecord.boundary:='n';
                          temprecord.maxormin:='max';
                          temprecord.number:=countmax;
                          temprecord.left:='n';
                          temprecord.right:='n';
                          write(outfileofrecord,temprecord);
                          close(outfileofrecord);
                        end;
if x2<x1 then if x2<x3 then
                        begin
                          countmin:=countmin+1;
                          resetoutfileofrecord(filenameofrecord);
```

```
                    seek(outfileofrecord,filesize(outfileofrecord));
                    temprecord.value:=x2;
                    temprecord.position:=no_of_step;
                    temprecord.boundary:='n';
                    temprecord.maxormin:='min';
                    temprecord.number:=countmin;
                    temprecord.left:='n';
                    temprecord.right:='n';
                    write(outfileofrecord,temprecord);
                    close(outfileofrecord);
                  end;
if x2<x1 then if x2=x3 then
                  begin
                    countmin:=countmin+1;
                    resetoutfileofrecord(filenameofrecord);
                    seek(outfileofrecord,filesize(outfileofrecord)
                    temprecord.value:=x2;
                    temprecord.position:=no_of_step;
                    temprecord.boundary:='n';
                    temprecord.maxormin:='min';
                    temprecord.number:=countmin;
                    temprecord.left:='n';
                    temprecord.right:='n';
                    write(outfileofrecord,temprecord);
                    close(outfileofrecord);
                  end;
if x2=x1 then if x2<x3 then
                  begin
                    countmin:=countmin+1;
                    resetoutfileofrecord(filenameofrecord);
                    seek(outfileofrecord,filesize(outfileofrecord)
                    temprecord.value:=x2;
                    temprecord.position:=no_of_step;
                    temprecord.boundary:='n';
                    temprecord.maxormin:='min';
                    temprecord.number:=countmin;
                    temprecord.left:='n';
                    temprecord.right:='n';
                    write(outfileofrecord,temprecord);
                    close(outfileofrecord);
                  end;

end;
if x2<x3 then
                  begin
                    countmax:=countmax+1;
                    resetoutfileofrecord(filenameofrecord);
                    seek(outfileofrecord,filesize(outfileofrecord)
                    temprecord.value:=x3;
                    temprecord.position:=no_of_step;
                    temprecord.boundary:='n';
```

```
                                temprecord.maxormin:='max';
                                temprecord.number:=countmax;
                                temprecord.left:='n';
                                temprecord.right:='n';
                                write(outfileofrecord,temprecord);
                                close(outfileofrecord);
                              end;
  if x2>x3 then
                              begin
                                countmin:=countmin+1;
                                resetoutfileofrecord(filenameofrecord);
                                seek(outfileofrecord,filesize(outfileofrecord))
                                temprecord.value:=x3;
                                temprecord.position:=no_of_step;
                                temprecord.boundary:='n';
                                temprecord.maxormin:='min';
                                temprecord.number:=countmin;
                                temprecord.left:='n';
                                temprecord.right:='n';
                                write(outfileofrecord,temprecord);
                                close(outfileofrecord);
                              end;
  close(infile);

{FIND YMINGLB}
  resetinfileofrecord(filenameofrecord);
  read(infileofrecord,temprecord);
  YMinGlb:=1;
  while not EOF(infileofrecord) do
  begin
    read(infileofrecord,temprecord);
    if temprecord.value<YMinGlb then YMinGlb:=temprecord.value;
  end;
  close(infileofrecord);

{FIND RANGE_HI_EVERAGE BY EVERAGE DIFFERENCE EVERY MAX TO EVERY MIN}
{  number:=0;
  range1:=0;
  n:=0;
  for n:=1 to countmax do
  begin
    resetinfileofrecord(filenameofrecord);
    seek(infileofrecord,1);
    while not EOF(infileofrecord) do
    begin
      read(infileofrecord,temprecord);
      if temprecord.maxormin ='max' then
         if temprecord.number = n then
                                        begin
                                          realtemp1:=temprecord.value;
                                          goto LABEL30;
```

```
                                                    end;
    end;
LABEL30:;
      close(infileofrecord);
      resetinfileofrecord(filenameofrecord);
      seek(infileofrecord,1);
      while not EOF(infileofrecord) do
      begin
        read(infileofrecord,temprecord);
        if temprecord.maxormin ='min' then realtemp2:=temprecord.value
           else goto LABEL10;
        number:=number+1;
        range1:=range1+realtemp1-realtemp2;
LABEL10:;
      end;
      close(infileofrecord);
  end;
  range_hi_everage:=range1/number;}

  range_hi_everage:=0.15;

{RANGE HIGHT EVERAGE BY DIFFERENCE EVERY MAX FROM YMinG1b}
 { number:=0;
  range2:=0;
  resetinfileofrecord(filenameofrecord);
  seek(infileofrecord,1);
  while not EOF(infileofrecord) do
  begin
    read(infileofrecord,temprecord);
    if temprecord.maxormin = 'min' then goto LABEL20;
    realtemp1:=temprecord.value;
    number:=number+1;
    range2:=range2+realtemp1-YMinG1b;
LABEL20:;
  end;
  range_hi_everage:=range2/number;
  close(infileofrecord);}

{FIND RANGE_LO_EVERAGE}

 { number:=0;
  range2:=0;
  resetinfileofrecord(filenameofrecord);
  seek(infileofrecord,1);
  while not EOF(infileofrecord) do
  begin
    read(infileofrecord,temprecord);
    if temprecord.maxormin = 'max' then goto LABEL12;
    realtemp1:=temprecord.value;
    number:=number+1;
    range2:=range2+realtemp1-YMinG1b;
```

```
LABEL12:;
  end;
  range_lo_everage:=range2/number;
  close(infileofrecord);}

  range_lo_everage:=0.1;

{SEGMENT}
  resetinfileofrecord(filenameofrecord);
  size_of_file:=filesize(infileofrecord);
  close(infileofrecord);
  writeln('countmax=',countmax);
  for nn:=1 to countmax do
  begin
    count1:=0;
    resetinfileofrecord(filenameofrecord);
    seek(infileofrecord,1);
LABEL13:;
    count1:=count1+1;
    read(infileofrecord,temprecord);
    if temprecord.maxormin = 'max' then
       if temprecord.number = nn then
          realtempmax:=temprecord.value
          else goto LABEL13
       else goto LABEL13;
  close(infileofrecord);

{SEGMENT RIGHTHAND SIZE}
    resetinfileofrecord(filenameofrecord);
    if count1+1>=size_of_file then seek(infileofrecord,count1)
      else seek(infileofrecord,count1+1);
    count2:=0;
LABEL1:;
    count2:=count2+1;
    if count1+count2>=size_of_file then goto LABEL3;
    read(infileofrecord,temprecord);
    if temprecord.maxormin = 'max' then
       if realtempmax<temprecord.value then goto LABEL3;
    goto LABEL1;
LABEL3:;
    close(infileofrecord);
    resetinfileofrecord(filenameofrecord);
    if count1+1>=size_of_file then seek(infileofrecord,count1)
       else seek(infileofrecord,count1+1);
    read(infileofrecord,temprecord);
    realtempmin:=temprecord.value;
    k:=0;
    if count2-1 <= 1 then goto LABEL4;
    for j:=1 to count2-2 do
    begin
      read(infileofrecord,temprecord);
```

```
      if temprecord.maxormin = 'max' then goto LABEL15;

      if realtempmin>temprecord.value then
                                        begin
                                          realtempmin:=temprecord.value;
                                          k:=j;
                                        end;

LABEL15:;
    end;
LABEL4:;
    close(infileofrecord);
    if realtempmax-realtempmin > range_hi_everage then
      if realtempmin < range_lo_everage then
             begin
              resetinfileofrecord(filenameofrecord);
              if count1+k+1>=size_of_file then seek(infileofrecord,count1+')
                 else seek(infileofrecord,count1+k+1);
              read(infileofrecord,temprecord);
              temprecord.boundary:='y';
              close(infileofrecord);
              resetinfileofrecord(filenameofrecord);
              if count1+k+1>=size_of_file then seek(infileofrecord,count1+')
                 else seek(infileofrecord,count1+k+1);
              write(infileofrecord,temprecord);
              close(infileofrecord);
             end;
    if realtempmax realtempmin > 0.02 then
      if realtempmin < 0.02 then
       { if realtempmax/realtempmin > 10 then}
             begin
              resetinfileofrecord(filenameofrecord);
              if count1+k+1>=size_of_file then seek(infileofrecord,count1+k)
                 else seek(infileofrecord,count1+k+1);
              read(infileofrecord,temprecord);
              if temprecord.boundary = 'y'  then goto LABEL100;
              temprecord.boundary:='1';
              close(infileofrecord);
              resetinfileofrecord(filenameofrecord);
              if count1+k+1>=size_of_file then seek(infileofrecord,count1+k)
                 else seek(infileofrecord,count1+k+1);
              write(infileofrecord,temprecord);
       LABEL100:;
              close(infileofrecord);
             end;
    if realtempmax-realtempmin > range_hi_everage then
    if realtempmin > range_lo_everage then
             begin
              resetinfileofrecord(filenameofrecord);
              if count1+k+1>=size_of_file then seek(infileofrecord,count1+')
                 else seek(infileofrecord,count1+k+1);
              read(infileofrecord,temprecord);
```

```
                if temprecord.boundary = 'y'  then goto LABEL101;
                temprecord.boundary:='2';
                close(infileofrecord);
                resetinfileofrecord(filenameofrecord);
                if count1+k+1>=size_of_file then seek(infileofrecord,count1-
                   else seek(infileofrecord,count1+k+1);
                write(infileofrecord,temprecord);
        LABEL101:;
                close(infileofrecord);
               end;

{SEGMENT LEFTHAND SIZE}
    count2:=0;
    n:=count1;
LABEL2:;
    count2:=count2+1;
    n:=n-1;
    if n<1 then goto LABEL5;
    resetinfileofrecord(filenameofrecord);
    seek(infileofrecord,n);
    read(infileofrecord,temprecord);
    close(infileofrecord);
    if temprecord.maxormin = 'max' then
       if realtempmax<temprecord.value then goto LABEL5;
    goto LABEL2;
LABEL5:;
    close(infileofrecord);
    resetinfileofrecord(filenameofrecord);
    if count1=1 then seek(infileofrecord,count1)
       else seek(infileofrecord,count1-1);
    read(infileofrecord,temprecord);
    close(infileofrecord);
    realtempmin:=temprecord.value;
    k:=1;
    if count2-1<=1 then goto LABEL6;
    for j:=1 to count2-1 do
    begin
      resetinfileofrecord(filenameofrecord);
      seek(infileofrecord,count1-j);
      read(infileofrecord,temprecord);
      close(infileofrecord);
      if temprecord.maxormin = 'max' then goto LABEL16;
      if realtempmin>temprecord.value then
                                        begin
                                          realtempmin:=temprecord.value;
                                          k:=j;
                                        end;
LABEL16:;
    end;
 LABEL6:;
    close(infileofrecord);
```

```
if realtempmax-realtempmin > range_hi_everage then
if realtempmin < range_lo_everage then
          begin
           resetinfileofrecord(filenameofrecord);
           if count1-k<1 then seek(infileofrecord,1)
              else seek(infileofrecord,count1-k);
           read(infileofrecord,temprecord);
           close(infileofrecord);
           temprecord.boundary:='y';
           resetinfileofrecord(filenameofrecord);
           if count1-k<1 then seek(infileofrecord,1)
              else seek(infileofrecord,count1-k);
           write(infileofrecord,temprecord);
           close(infileofrecord);
          end;
if realtempmax-realtempmin > 0.02
  if realtempmin < 0.02 then
    { if realtempmax/realtempmin >
          begin
           resetinfileofrecord(fi                ord);
           if count1-k<1 then se                 ecord,1)
              else seek(infile                   1-k);
           read(infileofreco
           if temprecord.bou                     then goto LABEL102;
           temprecord.boundary:=
           close(infileofrecord);
           resetinfileofrecord(filenameofrecord);
           if count1-k<1 then seek(infileofrecord,1)
              else seek(infileofrecord,count1-k);
           write(infileofrecord,temprecord);
    LABEL102:;
           close(infileofrecord);
          end;
if realtempmax-realtempmin > range_hi_everage then
if realtempmin > range_lo_everage then
          begin
           resetinfileofrecord(filenameofrecord);
           if count1-k<1 then seek(infileofrecord,1)
              else seek(infileofrecord,count1-k);
           read(infileofrecord,temprecord);
           if temprecord.boundary = 'y'  then goto LABEL103;
           temprecord.boundary:='2';
           close(infileofrecord);
           resetinfileofrecord(filenameofrecord);
           if count1-k<1 then seek(infileofrecord,1)
              else seek(infileofrecord,count1-k);
           write(infileofrecord,temprecord);
    LABEL103:;
           close(infileofrecord);
          end;
```

```
end;

{WRITE FILE YMAX AND YMIN}
{  resetinfileofrecord(filenameofrecord);
  seek(infileofrecord,1);
  number:=0;
  writeln(lst,'RANGE_HI_EVERAGE=',
          range_hi_everage,'  RANGE_LO_EVERAGE=',range_lo_everage);
  writeln(lst,'YMinGlb=',YMinGlb);
while not EOF(infileofrecord) do
 begin
   read(infileofrecord,temprecord);
   number:=number+1;
   writeln(lst,'NUMBER=',number);
   writeln(lst,'VALUE=',temprecord.value,'     POSITION=',temprecord.positi
   writeln(lst,'BOUNDARY=',temprecord.boundary,
           '     MAXORMIN=',temprecord.maxormin);
   writeln(lst);
 end;   }


{PITCH SUPPORT FOR SEGMENT}
filenameofrecord:='c:file';
resetinfileofrecord(filenameofrecord);
size_of_file:=filesize(infileofrecord);
close(infileofrecord);
for n:=1 to size_of_file-1 do
begin
  resetinfileofrecord(filenameofrecord);
  seek(infileofrecord,n);
  read(infileofrecord,temprecord);
  close(infileofrecord);
  {if temprecord.boundary = 'y' then
              begin
               position:=temprecord.position;
               resetinfile(filenametemp1);
               seek(infile,position);
               read(infile,realtemp);
               close(infile);
               if realtemp>0 then
                 begin
                  temprecord.boundary:='n';
                  resetoutfileofrecord(filenameofrecord);
                  seek(outfileofrecord,n);
                  write(outfileofrecord,temprecord);
                  close(outfileofrecord);
                 end;
              end; }

  if temprecord.boundary = '1' then
```

```
                    begin
                     position:=temprecord.position;
                     resetinfile(filenametemp1);
                     seek(infile,position);
                     read(infile,realtemp);
                     close(infile);
                     if realtemp<=0 then
                        begin
                         temprecord.boundary:='y';
                         resetoutfileofrecord(filenameofrecord);
                         seek(outfileofrecord,n);
                         write(outfileofrecord,temprecord);
                         close(outfileofrecord);
                        end;
                    end;

  end;


{REDEFINE BOUNDARY}

{FIND POSITION OF BOUNDARY}
filenameofinteger:='c:aa';
filenameofrecord:='c:file';
createfileofinteger(filenameofinteger);
resetoutfileofinteger(filenameofinteger);
resetinfileofrecord(filenameofrecord);
size_of_file:=filesize(infileofrecord);
seek(infileofrecord,1);
integertemp1:=0;
while not EOF(infileofrecord) do
begin
  read(infileofrecord,temprecord);
  integertemp1:=integertemp1+1;
  if temprecord.boundary='y' then
                              begin
                                write(outfileofinteger,integertemp1
                              end;
end;
close(infileofrecord);
close(outfileofinteger);

filenameofinteger:='c:aa';
resetinfileofinteger(filenameofinteger);
number_of_boundary:=filesize(infileofinteger);
  read(infileofinteger,integertemp1);
  integertemp2:=integertemp1-1;
LABEL52:;
  integertemp2:=integertemp2+1;
  if integertemp2>size_of_file then goto LABEL300;
```

```
  resetinfileofrecord(filenameofrecord);
  seek(infileofrecord,integertemp2);

  {RIGHT HAND}
  integertemp3:=integertemp2;
  read(infileofrecord,temprecord);
  if temprecord.maxormin='max' then
                                  begin
                                    close(infileofrecord);
                                    goto LABEL52;
                                  end;
  realtempmin:=temprecord.value;
  close(infileofrecord);
LABEL53:;
  integertemp2:=integertemp2+1;
  resetinfileofrecord(filenameofrecord);
  seek(infileofrecord,integertemp2);
  read(infileofrecord,temprecord);
  close(infileofrecord);
  if temprecord.maxormin='min' then
                                  begin
                                    close(infileofrecord);
                                    goto LABEL53;
                                  end;
  realtempmax:=temprecord.value;
  if (realtempmax/realtempmin)*100<300 then goto LABEL52;
  resetinfileofrecord(filenameofrecord);
  seek(infileofrecord,integertemp3);
  read(infileofrecord,temprecord);
  close(infileofrecord);
  temprecord.right:='y';
  resetoutfileofrecord(filenameofrecord);
  seek(outfileofrecord,integertemp3);
  write(outfileofrecord,temprecord);
  close(outfileofrecord);
LABEL300:;

for j:=1 to number_of_boundary-2 do
begin
writeln('loop1');
  read(infileofinteger,integertemp1);
  integertemp2:=integertemp1+1;
LABEL41:;
  integertemp2:=integertemp2-1;
  if integertemp2<1 then goto LABEL200;
  resetinfileofrecord(filenameofrecord);
n:=filesize(infileofrecord);
  seek(infileofrecord,integertemp2);

  {LEFT HAND}
  integertemp3:=integertemp2;
```

```
  read(infileofrecord,temprecord);
  if temprecord.maxormin='max' then
                                  begin
                                    close(infileofrecord);
                                    goto LABEL41;
                                  end;
  realtempmin:=temprecord.value;
  close(infileofrecord);
LABEL40:;
  integertemp2:=integertemp2-1;
  resetinfileofrecord(filenameofrecord);
  seek(infileofrecord,integertemp2);
  read(infileofrecord,temprecord);
  close(infileofrecord);
  if temprecord.maxormin='min' then
                                  begin
                                    close(infileofrecord);
                                    goto LABEL40;
                                  end;
  realtempmax:=temprecord.value;
  if (realtempmax/realtempmin)*100<300 then goto LABEL41;
  resetinfileofrecord(filenameofrecord);
  seek(infileofrecord,integertemp3);
  read(infileofrecord,temprecord);
  close(infileofrecord);
  temprecord.left:='y';
  resetoutfileofrecord(filenameofrecord);
  seek(outfileofrecord,integertemp3);
  write(outfileofrecord,temprecord);
  close(outfileofrecord);
LABEL200:;
  integertemp2:=integertemp1-1;
LABEL42:;
  resetinfileofrecord(filenameofrecord);
  integertemp2:=integertemp2+1;
  if integertemp2>size_of_file then goto LABEL400;
  seek(infileofrecord,integertemp2);

  {RIGHT HAND}
  integertemp3:=integertemp2;
  read(infileofrecord,temprecord);
  if temprecord.maxormin='max' then
                                  begin
                                    close(infileofrecord);
                                    goto LABEL42;
                                  end;
  realtempmin:=temprecord.value;
  close(infileofrecord);
LABEL43:;
  integertemp2:=integertemp2+1;
  resetinfileofrecord(filenameofrecord);
```

```
    seek(infileofrecord,integertemp2);
    read(infileofrecord,temprecord);
    close(infileofrecord);
    if temprecord.maxormin='min' then
                                  begin
                                    close(infileofrecord);
                                    goto LABEL43;
                                  end;
    realtempmax:=temprecord.value;
    if (realtempmax/realtempmin)*100<300 then goto LABEL42;
    resetinfileofrecord(filenameofrecord);
    seek(infileofrecord,integertemp3);
    read(infileofrecord,temprecord);
    close(infileofrecord);
    temprecord.right:='y';
    resetoutfileofrecord(filenameofrecord);
    seek(outfileofrecord,integertemp3);
    write(outfileofrecord,temprecord);
    close(outfileofrecord);
LABEL400:;
end;

    read(infileofinteger,integertemp1);
    integertemp2:=integertemp1+1;
LABEL61:;
    resetinfileofrecord(filenameofrecord);
    integertemp2:=integertemp2-1;
    if integertemp2<1 then goto LABEL500;
    seek(infileofrecord,integertemp2);

    {LEFT HAND}
    integertemp3:=integertemp2;
    read(infileofrecord,temprecord);
    if temprecord.maxormin='max' then
                                  begin
                                    close(infileofrecord);
                                    goto LABEL61;
                                  end;
    if temprecord.value=YMinGlb then goto LABEL61
       else realtempmin:=temprecord.value;
    close(infileofrecord);
LABEL60:;
    integertemp2:=integertemp2-1;
    resetinfileofrecord(filenameofrecord);
    seek(infileofrecord,integertemp2);
    read(infileofrecord,temprecord);
    close(infileofrecord);
    if temprecord.maxormin='min' then
                                  begin
                                    close(infileofrecord);
                                    goto LABEL60;
```

```
                                    end;
    realtempmax:=temprecord.value;
    if (realtempmax/realtempmin)*100<300 then goto LABEL61;
    resetinfileofrecord(filenameofrecord);
    seek(infileofrecord,integertemp3);
    read(infileofrecord,temprecord);
    close(infileofrecord);
    temprecord.left:='y';
    resetoutfileofrecord(filenameofrecord);
    seek(outfileofrecord,integertemp3);
    write(outfileofrecord,temprecord);
    close(outfileofrecord);
LABEL500:;


{DURATION SUPPORT FOR SEGMENT}
duration_value:=18;
filenameofrecord:='c:file';
resetinfileofrecord(filenameofrecord);
size_of_file:=filesize(infileofrecord);
writeln('size of file=',size_of_file);
close(infileofrecord);
n:=0;
LABEL703:;
n:=n+1;
if n>=size_of_file then goto LABEL701;
  resetinfileofrecord(filenameofrecord);
  writeln('n=',n);
  seek(infileofrecord,n);
  read(infileofrecord,temprecord);
  close(infileofrecord);
  if temprecord.right = 'y' then
      begin
        position1:=temprecord.position;
      LABEL700:;
        n:=n+1;
        resetinfileofrecord(filenameofrecord);
        seek(infileofrecord,n);
        read(infileofrecord,temprecord);
        close(infileofrecord);
        if temprecord.left='n' then goto LABEL700;
        integertemp1:=n;
        position2:=temprecord.position;
        if position2-position1<duration_value then
           begin
             temprecord.left:='n';
             resetinfileofrecord(filenameofrecord);
             seek(infileofrecord,n);
             write(infileofrecord,temprecord);
             close(infileofrecord);
             if temprecord.right='y' then
```

```
                begin
                  temprecord.right:='n';
                  resetinfileofrecord(filenameofrecord);
                  seek(infileofrecord,n);
                  write(infileofrecord,temprecord);
                  close(infileofrecord);
                  goto LABEL700;
                end;
            LABEL702:;
              n:=n+1;
              if n>=size_of_file then
                                  begin
                                    n:=integertemp1;
                                    resetinfileofrecord(filenameofrecord)
                                    seek(infileofrecord,n);
                                    read(infileofrecord,temprecord);
                                    close(infileofrecord);
                                    temprecord.right:='n';
                                    resetinfileofrecord(filenameofrecord)
                                    seek(infileofrecord,n);
                                    write(infileofrecord,temprecord);
                                    close(infileofrecord);
                                    goto LABEL701;
                                  end;
                resetinfileofrecord(filenameofrecord);
                seek(infileofrecord,n);
                read(infileofrecord,temprecord);
                close(infileofrecord);
                if temprecord.right='n' then goto LABEL702;
                temprecord.right:='n';
                resetinfileofrecord(filenameofrecord);
                seek(infileofrecord,n);
                write(infileofrecord,temprecord);
                close(infileofrecord);
                goto LABEL700;
    end;
    n:=n-1;
  end;
  goto LABEL703;
LABEL701:;



{DRAW SYLLABLE BOUNDARY}
InitGraphic;
DefineWindow(1,0,0,trunc((XMaxGlb+1)/3-1),
             trunc((YMaxGlb+1)/2-IVStepGlb));
```

```
DefineWindow(2,trunc((XMaxG1b+1)/3),0,trunc((XMaxG1b+1)/1.5-1),
             trunc((YMaxG1b+1)/2-IVStepG1b));
DefineWindow(3,trunc((XMaxG1b+1)/1.5),0,XMaxG1b,
             trunc((YMaxG1b+1)/2-IVStepG1b));
DefineWindow(4,0,trunc((YMaxG1b+1)/2-IVStepG1b+1),
             trunc((XMaxG1b+1)/3-1),YMaxG1b-IVStepG1b-1);
DefineWindow(5,trunc((XMaxG1b+1)/3),trunc((YMaxG1b+1)/2-IVStepG1b+1),
             trunc((XMaxG1b+1)/1.5-1),YMaxG1b-IVStepG1b-1);
DefineWindow(6,trunc((XMaxG1b+1)/1.5),trunc((YMaxG1b+1)/2-IVStepG1b+1),
             XMaxG1b,YMaxG1b-IVStepG1b-1);
DefineWindow(7,0,0,trunc((XMaxG1b+1)/2-1),YMaxG1b-IVStepG1b-1);
DefineWindow(8,trunc((XMaxG1b+1)/2+1),0,XMaxG1b,YMaxG1b-IVStepG1b-1);
DefineWindow(9,0,0,XMaxG1b,YMaxG1b-IVStepG1b-1);
DefineWindow(10,0,YMaxG1b-6,XMaxG1b,YMaxG1b);
DefineWindow(11,0,0,XMaxG1b,trunc((YMaxG1b+1)/2-IVStepG1b));
DefineWindow(12,0,trunc((YMaxG1b+1)/2-IVStepG1b+1),
             XMaxG1b,YMaxG1b-IVStepG1b-1);
clearscreen;
DrawBorder;
resetinfile(filename);
size_of_file:=filesize(infile);
start_no:=1;
no_to_plot:=size_of_file-1;
read(infile,step);
seek(infile,start_no);
for i:=1 to no_to_plot do
begin
  read(infile,a_plot[i,2]);
  a_plot[i,1]:=i;
end;
close(infile);
wind:=11;
selectwindow(wind);
findworld(1,a_plot,no_to_plot,1,1);
str(step:9,wrkst3);
head:=filename;
defineheader(wind,concat(head,': (STEP =',wrkst3,'MSEC)'));
head:=' ';
setheaderon;
drawborder;
setlinestyle(0);
drawaxis(-8,7,0,0,0,0,0,1,true);
drawpolygon(a_plot,1,no_to_plot,0,2,0);

{SEGMENT SYLLABLE FOR SAVE VALUE}
{filenameofrecord:='c:file';
resetinfileofrecord(filenameofrecord);
seek(infileofrecord,1);
while not EOF(infileofrecord) do
begin
  read(infileofrecord,temprecord);
```

```
  if temprecord.right='y' then
    begin
LABEL701:;
      position1:=temprecord.position;
LABEL700:;
      read(infileofrecord,temprecord);
      if temprecord.left='n' then goto LABEL700;
      position2:=temprecord.position;
      if temprecord.right='y' then goto LABEL701;
    end;
end;  }


{DRAW MARK SYLLABLE FOR GRAPHIC}
no_of_syllable:=0;
filenameofinteger:='c:boundary';
createfileofinteger(filenameofinteger);
resetoutfileofinteger(filenameofinteger);
write(outfileofinteger,integertemp1);
close(outfileofinteger);
filenameofrecord:='c:file';
resetinfileofrecord(filenameofrecord);
seek(infileofrecord,1);
position2:=1;
while not EOF(infileofrecord) do
begin
  read(infileofrecord,temprecord);
  if temprecord.right='y' then
    begin
LABEL601:;
      position1:=temprecord.position;
      if position1=position2 then position1:=position1+1;
      for i:= position2 to position1 do
      begin
        a_plot[i,1]:=i;
        a_plot[i,2]:=YMinGlb;
      end;
LABEL600:;
      read(infileofrecord,temprecord);
      if temprecord.left='n' then goto LABEL600;
      position2:=temprecord.position;
      no_of_syllable:=no_of_syllable+1;
      resetoutfileofinteger(filenameofinteger);
      seek(outfileofinteger,filesize(outfileofinteger));
      write(outfileofinteger,position1);
      write(outfileofinteger,position2);
      close(outfileofinteger);
      for i:= position1 to position2 do
      begin
        a_plot[i,1]:=i;
        a_plot[i,2]:=0.8;
```

```
        end;

        if temprecord.right='y' then goto LABEL601;
      end;
end;
if position2=no_to_plot then position2:=position2-1;
for i:=position2 to no_to_plot do
begin
  a_plot[i,1]:=i;
  a_plot[i,2]:=YMinGlb;
end;
axisglb:=true;
drawpolygon(a_plot,1,no_to_plot,0,2,0);
close(infileofrecord);

{DRAW PITCH}
resetinfile(filenametemp1);
size_of_file:=filesize(infile);
start_no:=1;
no_to_plot:=size_of_file-1;
read(infile,step);
seek(infile,start_no);
for i:=1 to no_to_plot do
begin
  read(infile,a_plot[i,2]);
  a_plot[i,1]:=i;
end;
close(infile);
wind:=12;
selectwindow(wind);
findworld(1,a_plot,no_to_plot,1,1);
str(step:9,wrkst3);
head:=filename;
defineheader(wind,concat(head,': (STEP =',wrkst3,'MSEC)'));
head:=' ';
setheaderon;
drawborder;
setlinestyle(0);
drawaxis(-8,7,0,0,0,0,0,1,true);
drawpolygon(a_plot,1,no_to_plot,0,2,0);


pompt('DO YOU WANT HARD COPY?(Y/N)');
repeat
  read(kbd,ch);
until ch in ['n','y'];
if ch='y' then
              begin
                hardcopy(false,6);
              end;
```

```
leavegraphic;

{WRITE DATA}
 { resetinfileofrecord(filenameofrecord);
  writeln(lst,'SEGMENT BY ENERGY ALPHA NO',
              ' SMOOTH SUPPORT BY PITCH AND DURATION');
  writeln(lst,'FIND RANGE HEIGHT EVERAGE BY',
              'EVERAGE DIFFERENCE EVERY MAX FROM EVERY MIN');
  writeln(lst,'SET REDEFINE BOUNDARY = 200');
  writeln(lst,'WINDOW SIZE = ',window_size,
          ' VALUE','     WINDOW FREQUENCY = ',window_freq,' VALUE');
  writeln(lst,'WINDOW SIZE OF PITCH = 250');
  writeln(lst);
  writeln(lst,'NUMBER OF SYLLABLE = ',no_of_syllable);
  writeln(lst);
  writeln(lst,'RANGE_HI_EVERAGE = ',range_hi_everage);
  writeln(lst,'RANGE_LO_EVERAGE = ',range_lo_everage);
  writeln(lst,'YMinG1b = ',YMinG1b);
  writeln(lst,'NUMBER OF MAX = ',countmax);
  writeln(lst);
while not EOF(infileofrecord) do
 begin
   read(infileofrecord,temprecord);
   if temprecord.left = 'y' then
   begin
     writeln(lst,'LEFT  ');
     writeln(lst,'VALUE = ',temprecord.value,
                 '     POSITION = ',temprecord.position);
     writeln(lst,'BOUNDARY = ',temprecord.boundary,
                 '     MAXORMIN = ',temprecord.maxormin);
   end;
   if temprecord.right = 'y' then
   begin
     writeln(lst,'RIGHT  ');
     writeln(lst,'VALUE = ',temprecord.value,
                 '     POSITION = ',temprecord.position);
     writeln(lst,'BOUNDARY = ',temprecord.boundary,
                 '     MAXORMIN = ',temprecord.maxormin);
   end;
 end;
 close(infileofrecord);}
end;

OVERLAY procedure Data(n:integer);
var J,M:integer;
    X:real;
begin
  nosound;
  CREATEFILE(datafile_name);
  RESETOUTFILE(datafile_name);
  step:=1/fs;
```

```
    write(outfile,step);
    for J:=1 to N do
    begin
      X:=-1.0;
      write(outfile,x);
    end;
    close(outfile);
  end;

  OVERLAY procedure SIFT;
  label LABEL1,LABEL2;
  var rnd,START:integer;
  begin
    nosound;
    rnd:=0;
    energyfile_name:='c:abuf';
    CREATEFILE(energyfile_name);
    STEP:=(1/fs);
    RESETOUTFILE(energyfile_name);
    seek(outfile,FileSize(outfile));
    write(outfile,step);
    close(outfile);
    filename:='c:pbuf';
    CREATEFILE(filename);
    STEP:=(1/fs);
    RESETOUTFILE(filename);
    seek(outfile,FileSize(outfile));
    write(outfile,step);
    close(outfile);
    write('HOW MUCH DOWN SAMPLING?');
    readln(down_sampl);
    write('WHAT PITCH FILE NAME?');
    readln(pitchfile_name);
    temp2file_name:=pitchfile_name;
    pitchfile_name:='c:filetem1';
    write('START AT?');
    READLN(START);
    CREATEFILE(pitchfile_name);
    STEP:=(window_freq/fs);
    RESETOUTFILE(pitchfile_name);
    seek(outfile,FileSize(outfile));
    write(outfile,step);
    close(outfile);
    for j:=1 to 3 do
    begin
      PITCH[j]:=0.0;
    end;
    RESETINFILE(datafile_name);
    size_of_file:=filesize(infile);
    seek(infile,START);
    flag1:=START;
```

```
LABEL1:;
  for J:=1 to window_size do spch[j]:=0.0;
  J:=1;
  full:=false;
  while not EOF(infile) and not full do
  begin
    read(infile,spch[j]);
    full:=true;
    if J<window_size then full:=false;
    j:=j+1;
  end;
  SIFTSTEP1(spch,pbuf);
{  RESETOUTFILE(filename);
  seek(outfile,FileSize(outfile));
  for j:=1 to 76 do
  begin
  write(outfile,pbuf[j]);
  end;
  close(outfile);  }
  rnd:=rnd+1;
  writeln('ROUND=',rnd);
  SIFTSTEP2(pbuf,pitch,pitchactual,down_sampl,fs);
  RESETOUTFILE(pitchfile_name);
  seek(outfile,FileSize(outfile));
  if pitchactual=-0.5 then pitchactual:=0.0;
  write(outfile,pitchactual);
  close(outfile);
  flag1:=flag1+window_freq;
  if flag1>=size_of_file then goto LABEL2;
  close(infile);
  resetinfile(datafile_name);
  seek(infile,flag1);
  goto LABEL1;
LABEL2:;
  close(infile);
  pitchactual:=(pitch[2]-1)*down_sampl/fs;
  RESETOUTFILE(pitchfile_name);
  seek(outfile,FileSize(outfile));
  if pitchactual=-0.5 then pitchactual:=0.0;
  write(outfile,pitchactual);
  close(outfile);
  pitchactual:=(pitch[1]-1)*down_sampl/fs;
  RESETOUTFILE(pitchfile_name);
  seek(outfile,FileSize(outfile));
  if pitchactual=-0.5 then pitchactual:=0.0;
  write(outfile,pitchactual);
  close(outfile);
  createfile(temp2file_name);
  resetinfile(pitchfile_name);
  resetoutfile(temp2file_name);
  read(infile,step);
```

```
  write(outfile,step);
  close(infile);
  resetinfile(pitchfile_name);
  seek(infile,3);
  while not EOF(infile) do
  begin
    read(infile,realtemp);
    write(outfile,realtemp);
  end;
  close(infile);
  close(outfile);
end;

OVERLAY procedure PLOT;
begin
nosound;
InitGraphic;
DefineWindow(1,0,0,trunc((XMaxGlb+1)/3-1),
             trunc((YMaxGlb+1)/2-IVStepGlb));
DefineWindow(2,trunc((XMaxGlb+1)/3),0,trunc((XMaxGlb+1)/1.5-1),
             trunc((YMaxGlb+1)/2-IVStepGlb));
DefineWindow(3,trunc((XMaxGlb+1)/1.5),0,XMaxGlb,
             trunc((YMaxGlb+1)/2-IVStepGlb));
DefineWindow(4,0,trunc((YMaxGlb+1)/2-IVStepGlb+1),
             trunc((XMaxGlb+1)/3-1),YMaxGlb-IVStepGlb-1);
DefineWindow(5,trunc((XMaxGlb+1)/3),trunc((YMaxGlb+1)/2-IVStepGlb+1),
             trunc((XMaxGlb+1)/1.5-1),YMaxGlb-IVStepGlb-1);
DefineWindow(6,trunc((XMaxGlb+1)/1.5),trunc((YMaxGlb+1)/2-IVStepGlb+1)
             XMaxGlb,YMaxGlb-IVStepGlb-1);
DefineWindow(7,0,0,trunc((XMaxGlb+1)/2-1),YMaxGlb-IVStepGlb-1);
DefineWindow(8,trunc((XMaxGlb+1)/2+1),0,XMaxGlb,YMaxGlb-IVStepGlb-1);
DefineWindow(9,0,0,XMaxGlb,YMaxGlb-IVStepGlb-1);
DefineWindow(10,0,YMaxGlb-6,XMaxGlb,YMaxGlb);
DefineWindow(11,0,0,XMaxGlb,trunc((YMaxGlb+1)/2-IVStepGlb));
DefineWindow(12,0,trunc((YMaxGlb+1)/2-IVStepGlb+1),
             XMaxGlb,YMaxGlb-IVStepGlb-1);
clearscreen;
DrawBorder;
repeat
  Pompt('1=HEADER,2=PLOT,3=CLEAR WINDOW,4=MOVE WINDOW,
        5=CLEAR SCREEN,6=HARD COPY,7=DEFINE POINT,e=END');
  read(kbd,ch);
  case ch of
    '1':DefineHeader_Test;
    '2':Plot_Test;
    '3':ClearWindow_Test;
    '4':MoveWindow_Test;
    '5':ClearScreen_Test;
    '6':HardCopy_Test;
    '7':Define_point;
  end;
```

```
until ch='e';
leavegraphic;
end;
```

```
B>type big500.asm
;
;  program interface of A to D board  version 2.30
;
stack     segment     para stack  'stack'
          db          1024 dup(0)             256 bytes of stack space
stack     ends
;
data      segment     para public  'data'
des       dw          2 dup(0), $
hex       db          16
mess1     db  0dh,0ah,"Input LABEL of word (max 20 chars) ... [ENTER]  ","$"
mess2     db  0dh,0ah,"Do you want to save these data to disk (Y/N) ?  ","$"
mess3     db  0dh,0ah, Input your file name [d:(path\)file.ext] ?"
          db          0dh,0ah," (ENTER only to cancel ...)  ","$"
mess4     db  0dh,0ah, Can not access these file name. retry again !",7
          db          0dh,0ah,"$"
mess5     db          007h
          db  0dh,0ah, Disk full. . ! change a new one and retry again !","$"
mess6     db  0dh,0ah,".....S T A R T.....D O W N L O A D.....",0DH,0AH,0AH
          db  0dh,0ah,"****** please   wait   a   minute ******",0dh,0ah,"$"
mess7     db  0dh,0ah,0ah,"..D O W N L O A D.....C O M P L E T E ..","$"
mess8     db  0dh,0ah,"Label input already ......",0dh,0ah
          db          "press any key to start input analog "
          db          "signal...","$"
new_line  db          0dh,0ah,"$"
long      dw          00
handle    dw          00
file      db          20 dup(0)
label     db          20 dup(0)
datain    db          10240 DUP(0)     ;63488 dup(0)     ;62 Kbytes
buffer    db          10240 dup(0)
pointer   dw          offset datain
int_cod   dw          0
int_off   dw          0
data      ends
;
code      segment     para public 'code'
;
start     proc        far
;   standard  program  prologue
          assume      cs:code
          push        ds          ;save psp seg addr
          mov         ax,0
          push        ax          ;save ret addr offset (psp+0)
          mov         ax,data
          mov         ds,ax       ;establish data seg addressability
          assume      ds:data
;
; set interrupt vector for IRQ2 at cs: = 00 , ip: = 28h
          cli
          mov         ah,35h
          mov         al,0ah
          int         21h
          mov         int_cod,es
          mov         int_off,bx
          push        ds
          push        cs
          pop         ds
          mov         dx,offset stod
```

```
          mov       ah,25h
          mov       al,0ah
          int       21h
          pop       ds
          mov       al,0f9h       ; set interrupt mask register
          out       21h,al
          sti
; input label of data
inp:      mov       dx,offset mess1
          call      disphex
          mov       di,offset label
input     call      read_kbd   ;read from key board
          mov       dl,al
          call      disp_chr   ;echo character
          cmp       al,0dh     ;Is it cr-return ?
          je        begin
          cmp       al,08h     ;check for back space key
          jne       noback
          dec       di         ;if yes decrement pointer
          cmp       di,offset label ;pointer below offset of label ?
          jnl       input           ;if not goto input again
          jmp       inp             ;if yes start again
noback:   mov       [di],al    ;move label to memory
          inc       di
          cmp       di,offset datain
          jnl       begin
          jmp       input
; command to start a to d conversion.
begin:    mov       dx,offset mess8
          call      disphex
          call      read_kbd
          mov       dx,offset new_line
          call      disphex
          xor       ax,ax
          mov       dx,0780h
          out       dx,al
;
forever   cli
          mov       ax,offset buffer
          cmp       ax,[pointer]
          jle       stop       ;if not enough memory go out of loop
;          mov       ah,00
;          int       16h        ;check kbd have been strucked ?
;          cmp       ah,0
;          jnz       stop       ;if yes out of loop
          sti
          jmp       forever
;
stop:     nop
          mov       al,04h
          out       21h,al     ; disable interrupt from IRQ2
          push      ds
          mov       dx,int_off
          mov       ax,int_cod
          mov       ds,ax
          mov       ah,25h
          mov       al,0ah
          int       21h
          pop       ds

;          MOV       AH,4CH
;          INT       21H
;=========================================

          mov       bx,offset label
          mov       cx,[pointer]
```

```
          sub         cx,bx
          mov         long,cx
          xor         ah,ah
;loop:     mov         al,[bx]      ;loop to convert data to hex
;                                   ;and display them
;          push        bx
;          push        di
;          jmp         start1
;table     db          0123456789abcdef
;start1    xor         ah,ah
;          xor         bx,bx
;          div         hex
;          mov         bl,al
;          mov         al,cs:table[bx]
;          mov         di,offset des
;          mov         byte ptr [di],al
;          mov         bl,ah
;          mov         al,cs:table[bx]
;          mov         byte ptr [di+1],al
;          pop         di
;          pop         bx
;          mov         dx,offset des
;          call        disphex
;          inc         bx
;          dec         cx
;          cmp         cx,0
;          ja          loop
;
again:    mov         dx,offset mess2  ;loop to ask for download or not
          call        disphex
          call        read_kbd
          mov         dl,al
          call        disp_chr
          cmp         al,'y'
          je          save
          cmp         al,'Y'
          je          save
          cmp         al,'n'
          je          termin
          cmp         al,'N'
          je          termin
          jmp         again
save:     mov         dx,offset mess3
          call        disphex
          mov         si,offset file
          xor         cx,cx        ;set cx is 0
con:      call        read_kbd
          mov         dl,al
          call        disp_chr     ;echo character from keyboard
          cmp         al,08h       ;check for back space
          jne         nobacks
          dec         si           ;if yes decrement pointer and
          dec         cx           ;decrement counter
          cmp         cx,0         ;back space under offset file
          jnl         con          ;if no continue read kbd
          jmp         save         ;if yes start again
nobacks:  cmp         al,0dh       ;check cr-return
          jne         next_chr
          cmp         cx,00        ;if cr-return only and no input file name
          je          termin       ;  then terminate program
          mov         al,00
          mov         [si],al      ;else make it to ASCIIZ form
          jmp         file_ok      ;  then file name input already
;
next_chr: mov         [si],al
          inc         cx
```

```
          inc       si
          cmp       cx,20       ;check file name not exceed 20 chars
          ja        save        ;if exeed go to input again.
          jmp       con                                          216
termin:   jmp       termin1
file_ok:  mov       dx,offset mess6
          call      disphex
;   create a file.....................
          mov       dx,offset file
          mov       cx,20h
          mov       ah,3ch
          int       21h
          jc        fail        ;can not create a file
;   open this file....................
          mov       dx,offset file
          mov       al,2
          mov       ah,3dh
          int       21h
          jc        fail        ;can not open the file
          mov       handle,ax
;   write this file...................
          mov       dx,offset label
          mov       cx,long
          mov       bx,handle
          mov       ah,40h
          int       21h
          jc        fail        ;can not write
          cmp       long,ax
          jne       dsk_ful     ;disk full
;   close this file...................
          mov       bx,handle
          mov       ah,3eh
          int       21h
          mov       dx,offset mess7
          call      disphex
          jmp       termin1
;
dsk_ful:  mov       dx,offset mess5
          call      disphex
          jmp       again
;
fail:     mov       dx,offset mess4
          call      disphex
          jmp       save
termin1:  mov       ah,4ch      ;terminate program
          int       21h
;
read_kbd  proc      near
          mov       ah,08h
          int       21h
          ret
read_kbd  endp
;
disp_chr  proc      near
          mov       ah,02h
          int       21h
          ret
disp_chr  endp
;
disphex   proc      near
          mov       ah,09h
          int       21h
          ret
disphex   endp
;
dist      proc      near
```

```
        push    bx
        mov     bx,0
        mov     ah,14
        int     10h
        pop     bx
        ret
disp    endp

atod    proc    far
        push    ax
        mov     dx,0780h    ; recieve data
        out     dx,al       ; start conversion again
        in      al,dx
;       call    disp
        mov     di,[pointer]
        mov     [di],al

        inc     [pointer]
        mov     al,20h      ; signal End Of Interrupt command
        out     20h,al
        pop     ax
        iret
atod    endp

start   endp
code    ends
        end     start
```

ประวัติผู้เขียน

เกิดวันที่ 26 มกราคม พ.ศ.2502 ที่อำเภอ บางระจัน จังหวัด สิงห์บุรี เป็นบุตรคนที่สองของนาย บุญส่ง และ นาง กิมเซี้ยะ ประทุมทาน ประวัติการศึกษาโดยย่อมีดังนี้

-พ.ศ.2512 จบการศึกษาระดับประถมศึกษาชั้นปีที่ 4 จากโรงเรียนวัดพระปรางค์

-พ.ศ.2515 จบการศึกษาระดับประถมศึกษาชั้นปีที่ 7 จากโรงเรียนกัลยานุเคราะห์

-พ.ศ.2520 จบการศึกษาระดับมัธยมศึกษาชั้นปีที่ 5 จากโรงเรียนสวนกุหลาบวิทยาลัย

-พ.ศ.2525 จบการศึกษาระดับปริญญาตรี คณะวิศวกรรมศาสตร์ จากสถาบันเทคโนโลยี พระจอมเกล้า บางมด

-พ.ศ.2526 จบการศึกษาระดับปริญญาตรี คณะเศรษฐศาสตร์ จากมหาวิทยาลัย รามคำแหง