เอกสารอ้างอิง

ภาษาไทย

ขวัญชัย ลีเผ่าพันธุ์, การแต่งแร่2, หน้า 1-44, ภาควิชาวิศวกรรมเหมืองแร่และธรณีวิทยาเหมืองแร่ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย, กรุงเทพมหานคร, 2530.

_____, " การใช้ระบบผู้เชี่ยวชาญ(EXPERT SYSTEM)ในงานเกี่ยวกับแร่, " ประชุมวิชาการด้านเหมืองแร่ ครั้งที่2, หน้า 437-443, ภาควิชาวิศวกรรมเหมืองแร่และโลหะวิทยา คณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์, สงขลา, 2529.

ฐิติศักดิ์ บุญปราโมทย์, ขวัญชัย ลีเผ่าพันธุ์, " การเลียนแบบวงจรการบดแร่," ประชุมวิชาการด้านเหมืองแร่ ครั้งที่2, หน้า 327-339, ภาควิชาวิศวกรรมเหมืองแร่และโลหะวิทยา คณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์, สงขลา, 2529.

ทรัพยากรธรณี, กรม, คู่มือการแต่งแร่, หน้า 132-155, โรงพิมพ์ชุมนุมสหกรณ์การเกษตรแห่งประเทศไทย จำกัด, กรุงเทพมหานคร, พิมพ์ครั้งที่2, 2522.

___, แร่, หน้า 123-125, กองเศรษฐกิจและเผยแพร่, กรุงเทพมหานคร, พิมพ์ครั้งที่3, 2526.

มานัส วีระบุรุษ, " ฟลูออไรท์," ข่าวสารการธรณี, 19(1), 22-38, มกราคม 2517.

มานุส มณีนุษย์, "ประสิทธิภาพวงจรการบดแร่ด้วยหม้อบดเซมิ-ออโตจิเนียสและไฮโดรไซโคลน," วิทยานิพนธ์ปริญญามหาบัณฑิต ภาควิชาวิศวกรรมเหมืองแร่และธรณีวิทยาเหมืองแร่ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย, 2533.

สุรชัย ปัทมศรีรัตนา, ขวัญชัย ลีเผ่าพันธุ์, " การเตรียมแร่ป้อนสำหรับการลอยแร่โดยใช้เทคนิคโปรแกรมเชิงเส้น," ประชุมวิชาการด้านเหมืองแร่ ครั้งที่2, หน้า 495-501, ภาควิชาวิศวกรรมเหมืองแร่และโลหะวิทยา คณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์, สงขลา, 2529.

สง่า ตั้งชวาล, แร่วิทยาสำหรับวิศวกร, หน้า 250-251, ภาควิชาวิศวกรรมเหมืองแร่และธรณีวิทยาเหมืองแร่ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย, กรุงเทพมหานคร

ศิริจันทร์ ทองประเสริฐ, การจำลองแบบปัญหา, หน้า 1-11, ภาควิชาวิศวกรรมอุตสาหการ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย, กรุงเทพมหานคร, 2529.

ภาษาอังกฤษ

Arbiter, N.and Harris, C.C., 1962. Flotation kinetics in Froth Flotation - 50th Anniversary Volume (ed. D.W. Fuerstenau), pp. 215-246, (Rocky Mountain Fund Series).

_____, 1951. Flotation rates and flotation efficiency, Trans.A.I.M.E., 190, pp. 791-796.

Ball, B. and Fuerstenau, D.W., 1974.On the determination of rate constants from semi-batch flotation tests, Quart. Col. School of Mines, 69, pp. 27-40.

Bisshop, J.P.and White,M.E., 1976. Study of particle entrainment in flotation froths, Proc.I.M.M., 85, pp. 191-194.

Booth, R.B. and Freyberger, W.L., 1962. Froths and frothing agents, in Froth Flotation - 50th Anniversary Volume, (ed.D.W.Fuerstenau), pp. 258-276, (Rocky Mountain Fund Series).

Brown, D.J. and Smith, H.G., 1954. The flotation of coal as a rate process, Trans.I.M.E., London, 113, pp. 1001-1017.

Bull, W.R., 1965. Flotation kinetics and its application to the interpretation of plant performance and the design of treatment circuits, 8th British Commonwealth Min.Met.Congress, Melbourne, General Proc. (ed. J.T.Woodcock, R.T. Madigan and R.G. Thomas), pp. 1113-1124.

_____, 1966. The rates of flotation of mineral particles in sulphide ores, Proc.Aus.I.M.M., No. 220, Dec., pp. 69-78.

Bushell, C.G.H., 1962. Kinetics of flotation, Trans.A.I.M.E., 223,pp. 266-278.

Cooper, H.R., 1966. Feedback process control of mineral flotation. Part 1. Development of a model for froth flotation, Trans. A.I.M.E., 235, pp. 439-446.

Cutting, G.W.and Devenish, M., 1975. A steady-state model of froth flotations structures, A.I.M.E. Annual Meeting, N.Y.Feb., Preprint 75-B56.

Danckwerts, P.V., 1953. Continuous flow systems: distribution of residence times, Chem. Eng. Sci., 2 : pp. 1-13.

Davis, W.J.N., 1964. The development of a mathematical model of the lead flotation circuit at the Zinc Corporation Limited, Proc. Aus. I.M.M., No. 212, pp. 61-89.

De Bruyn, P.L. and Modi, H.J., 1956. Particle size and flotation of quartz, Trans.A.I.M.E., 202, pp. 415-419.

Dinsdale,J.D. and Berube, Y., 1972. A characterization of hydrodynamics in a 700 cu.ft. Maxwell flotation cell, Can. Metall. Q., 11, pp. 507-513. Engelbrecht, J.A. and Woodburn, E.T., 1975. The effects of froth height, aeration rate, and precipitation on flotation, J.Sth.Afr.Inst.Min.Metall., 76, pp. 125-132.

Dzieniewiez, J. and Pryor, E,. j., 1950. Trans.IMM, 59.

Faulkner, B.P., 1966. Computer control improves metallurgy at Tennessee Copper's flotation plant, Mining Engineering, 18, No. 11, pp. 53-57.

Flint, L.R., 1973. Factors influencing the design of flotation equipment, Miner. Sci. Engng., 5, No.3, July, pp. 232-241.

_____, 1974. A mechanistic approach to flotation kinetics, Trans.I.M.M., 83, c90-95.

Frew, J.A. and Restarick, C.J., 1978. Relationship between flotation rates and pulp density in Zinc cleaning circuits at Broken Hill, Aus.I.M.M., N.W. Qld. Branch, Mill Operators Conference.

Harris, C.C. and Rimmer, H.W.,1966. Study of a two-phase model of the flotation process, Trans.I.M.M., 75, pp. C153-C162.

_____, Rimmer, H.W. and Angiulli, J., 1966. Unpublished work, Columbia University, New York.

_____ and Chakravarti, A., 1970. Semi-batch froth flotation kinetics; species distribution analysis, Trans.A.I.M.E., 247, pp. 162-172.

_____, Chakravarti, A. and Degaleesan, S.N., 1975. A recycle flow flotation machine model, Int.J.Min. Proc., 2, No.1, pp. 39-58. Imaizumi, T. and Inoue, T., 1963. Kinetic considerations of froth flotation, Proc. 6th Int.Min.Proc.Congress(Cannes), (ed.A. Roberts), pp. 581-593.

Johnson, N.W., 1972. The flotation behaviour of some chalcopyrite ores, Ph.D. Thesis(unpublished), University of Queensland.

_____, Mckee, D.J. and Lynch, A.J., 1972. Flotation rates of non sulphide minerals in chalcopyrite flotation processes, A.I.M.E. Annual Meeting, (San Francisco, California : Feb. 20-24, 1972). (Published Trans.A.I.M.E. 256, 1974).

Jowett, A., 1961. Investigation of residence time of fluid in froth flotation cells. Brit. Chem. Eng., 6, pp. 254-259.

_____ and Safvi,S.M.M.,1959. Refinements in methods of determining rates, Trans.A.I.M.E., 217, pp. 351-357.

_____, 1966. Gangue mineral contamination of froth, Brit.Chem.Eng., 11, pp. 330-333.

Kapur, P.C. and Mehrotra, S.P., 1974. Estimation of the flotation rate distributions by numerical inversion of the Laplace transform, Chem. Eng. Sci., 29, pp. 411-415.

Kay, K.R. and Patterson, J.G., 1975. Computer control at the Clarabelle Mill, Bull. C.I.M., 68, pp. 65-70.

Kelsall, D.F., 1961. Application of probability in the assessment of flotation systems, Trans.Inst.Min.Met., 70, pp. 191-204.

_____ and Stewart, P.S.B., 1971. A critical review of applications of models of grinding and flotation, Automatic Control Systems in Mineral Processing Plants, (Symposium Aus. I.M.M.: Brisbane), pp. 213-232.

_____, _____ and Trahar, W.J., 1974. Diagnostic metallurgy, a systematic method of plant optimisation, A.M.I.R.A. Meeting on Optimisation and Control, Brisbane.

King, R.P., 1973. Computer controlled flotation plants in Canada and Finland, Report No. 1517, National Institute for Metallurgy, South Africa.

_____, 1974. The simulation of flotation plants, S.M.E. Preprint 74 B-25, A.I.M.E. Annual Meeting, Dallas, Texas.

_____ and Buchalter, E.M., 1974. A pilot plant investigation of a flotation model, Report No. 1573, National Institute for Metallurgy, South Africa.

_____, 1976. The use of simulation in the design and modification of flotaion plants, Flotation - A.M. Gaudin Memorial Volume, ed. M.C. Fuerstenau, pp. 937-962.

Lanczos, C., 1956. Applied Analysis, Prentice Hall, Englewood Cliffs, N.J.

Levenspiel, O., 1962. Chemical Reaction Engineering, John Wiley and Sons, New York and London, pp. 246-252.

Lee, A.F., 1978. A new bubble pick-up technique as a rapid flotation test method, CIM Bulletin, No.71, 119.

Livshits, A.K. and Bezrodnaya, R.M., 1961. Rate of water and solids passage into flotation froth products, Tsvet Met., No. 11, pp. 14-16.

Lynch, A.J., Johnson, N.W., McKee, D.J. and Thorne, G.C., 1974. The behaviour of minerals in sulphide flotation processes, with reference to simulation and control, J.Sth.Afr.Inst.Min.Met., 74, No. 9, pp.349-361.

Mika, T.S., 1967. Discussion of paper by H.R. Cooper, Trans.A.I.M.E., 238, pp. 479-483.

MOrris, T.M., 1952. Discussion of 'Flotation Rates and Flotation Efficiency' by N. Arbiter, Mining Eng., 4, No. 8, pp. 794-798.

Mular, A.L., 1972. Empirical modelling and optimisation of mineral processes, Mineral Science and Engineering, 4, No.3, pp.30-42.

Niemi,A.,1966a. A study of dynamic and control properties of industrial flotation processes, Acta Polytech. Scand., Chem. Met. Ser., No. 48.

_____, 1966b. On the dynamics of a pneumatic flotation cell, Acta Poly-tech. Scand. Chem. Met. Ser., No. 49.

Pitt, J.C., 1968. The development of systems for continuous optimal control of flotation plants by computer. Systems Dynamics and Automatic Control in Basic Industries, (I.F.A.C. Symposium, Sydney), pp. 165-171.

Pogorelyi, A.D., 1962. Limits of the use of the kinetic equation proposed by K.F. Beloglazov, Invest. Uysshikh Ucheb. Zavendenii, Tsvet. Met., No. 5, pp. 33-40. Chem. Abstr., 57, 8256.

Rimmer, H.W., 1971. Private communication.

Schuhmann, R., 1942. Methods for steady state study of flotation problems, J. Phys. Chem., 46, pp. 891-902.

Smith, H.W. and Lewis, C.L., 1969. Computer control experiments at Lake Dufault, Can. I.M.M. Bulletin, 62, No. 682, pp. 109-115.

Steiner, H.J., 1972. Kinetic aspects of the flotation behaviour of locked particles. 10th Int. Min. Proc. Congress, London, pp. 653-666.

Taggart, A.F., 1944. Handbook of Mineral Dressing, John Wiley and Sons, New York, pp. 12-52 to 12-55.

Thatcher, C.M., 1962. Fundamentals of Chemical Engineering, Merrill Books Inc., Columbus, Ohio.

Thorne, G.C., 1975. The flotation behaviour of lead/zinc ores in industrial flotation circuits with reference to simulation and control, Ph.D. Thesis (unpublished), University of Queensland.

_____, Manlapig, E.V., Hall, J.S. and Lynch, A.J., 1976. Modelling of industrial sulphide flotation circuits, Flotation - A.M. Gaudin Memorial Volume, ed. M.C. Fuerstenau, pp. 725-752.

Tomlinson, H.S. and Fleming, M.G., 1965. Flotation rate studies, Mineral Processing, (ed. A. Roberts), 6th Int. Min. Proc. Congress, Cannes. pp. 563-579.

Trambouze, P., 1972. Single phase and liquid-liquid reactors; mixing and residence time distribution, Proceedings of the 5th European/2nd International Symposium on Chemical Reaction Engineering, Amsterdam, 2-4 May.

Volkova, Z.O., 1946. On the law governing the process of separation of solids of different floatability, Acta Phys. Chem., 21, p.1105.

Waksmundzki, A., Neczaj-Hruzewicz, J. and Planik, M., 1972. Mechanism of carryover of gangue slimes during flotation of sulphur ore, Trans.I.M.M., 81, pp. 249-251.

Watson, D. and Grainger-Allen, T.J.N., 1974. Study of froth flotation using a steady-state technique, Trans.A.I.M.E., 356,pp.242-247.

Weigel, R.L., 1976. Integrated size reduction-mineral liberation model, Trans.A.I.M.E., 260, June, pp. 147-152.

Whiten, W.J., 1971. The use of multi-dimensional cubic spline functions for regression and smoothing. The Aust. Computer JOurnal, 3, pp. 81-87.

_____, 1971. Model building techniques applied to mineral treatment processes. Aus.I.M.M. Symposium on Automatic Control Systems in Mineral Processing Plants, University of Queensland, Brisbane.

_____, 1972. Simulation and model building for mineral processing, Ph.D. Thesis (unpublished) University of Queensland, pp.208-213.

Will, B.A., 1985. 'Flotation,' Mineral Processing Technology, Chap. 12, pp. 316-387, Pergamon Press Oxford, London, 3rd ed.

Woodburn, E.T. and Loveday, B.K., 1965. Effect of variable residence time on the performance of a flotation system, J.S.Afr. Inst. Min. Met., 65, pp. 612-628.

_____, 1970. Mathematical modelling of flotation processes, Minerals Science and Engineering, 2, No., 2, pp.3-17.

_____, Kropholler, H.W., Greene, J.C.A. and Cramer, L.A., 1976. The utility and limitations of mathematical modelling in the prediction of the properties of flotation networks, Flotation - A.M. Gaudin Memorial Volume, ed. M.C. Fuerstenau, pp. 638-674.

Zuniga, H.G., 1935. The efficiency obtained by flotation is an exponential function of time, Bol. Minero. Soc. Nacl. Mineria (Chile), 47, pp. 83-86.

ภาคผนวก

ภาคผนวก ก.

โปรแกรมหลักการจำลองแบบวงจรลอยแร่ บ. เหมืองแร่ตาดดาวจำกัด

ซึ่งเขียนด้วยภาษาฟอร์แทรน 77

```
1   :     THE MEAN OF VARIABLE FOR SMAIN.FOR
2   :     N       = THE NUMBER OF COMPONENT
3   :     COMP    = NAME OF COMPONENT IN FLOTATION CELL
4   :     COMP(1) = WATER COMPONENT
5   :     COMP(2) = FAST FLOATING COMPONENT
6   :     COMP(3) = SLOW FLOATING COMPONENT
7   :     COMP(4) = GANGUE COMPONENT
8   :     MTCOMP  = NAME OF FLOATING ORE
9   :     ASSAY(2)= ASSAY OF FAST FLOATING COMPONENT
10  :     ASSAY(3)= ASSAY OF SLOW FLOATING COMPONENT
11  :     ASSAY(4)= ASSAY OF GANGUE COMPONENT
12  :     PAR(1)  = CELL'VOLUME (CU.M)
13  :     PAR(2)  = BETA CONSTANT
14  :     SOLIDS  = FEED RATE OF ORE (T/Hr.)
15  :     FEED(1) = FEED RATE OF WATER (T/Hr.)
16  :     FEED(2) = FEED RATE OF FAST FLOATING COMPONENT (% OF ORE FEED RATE)
17  :     FEED(3) = FEED RATE OF SLOW FLOATING COMPONENT (% OF ORE FEED RATE)
18  :     FEED(4) = FEED RATE OF GANGUE (% OF ORE FEED RATE)
19  :     SG(2)   = SPECIFIC GRAVITY OF FAST FLOATING COMPONENT
20  :     SG(3)   = SPECIFIC GRAVITY OF SLOW FLOATING COMPONENT
21  :     SG(4)   = SPECIFIC GRAVITY OF GANGUE
22  :     RATEF(2)= RATE CONSTANT OF FAST FLOATING COMPONENT
23  :     RATEF(3)= RATE CONSTANT OF SLOW FLOATING COMPONENT
24  :     RATEN(1)= RATE CONSTANT OF NON FLOATING COMPONENT (WATER)
25  :     RATEN(4)= RATE CONSTANT OF NON FLOATING COMPONENT (GANGUE)
26  :     NCELL1  = THE NUMBER OF ROUGHER CELLS
27  :     NCELL2  = THE NUMBER OF SCAVENGER CELLS
28  :     NCELL3  = THE NUMBER OF 1st CLEANER CELLS
29  :     NCELL4  = THE NUMBER OF 2nd CLEANER CELLS
30  :     NCELL5  = THE NUMBER OF 3rd CLEANER CELLS
31  :     CONC1   = CONCENTRATE FLOWRATE OF ROUGHER BANK (CU.M/Hr.)
32  :     CONC2   = CONCENTRATE FLOWRATE OF SCAVENGER BANK (CU.M/Hr.)
33  :     CONC3   = CONCENTRATE FLOWRATE OF 1st CLEANER BANK (CU.M/Hr.)
34  :     CONC4   = CONCENTRATE FLOWRATE OF 2nd CLEANER BANK (CU.M/Hr.)
35  :     CONC5   = CONCENTRATE FLOWRATE OF 3rd CLEANER BANK (CU.M/Hr.)
36  :     TAIL1   = TAILING FLOWRATE OF ROUGHER BANK (CU.M/Hr.)
37  :     TAIL    = TAILING FLOWRATE OF SCAVENGER BANK (CU.M/Hr.)
38  :     TAIL3   = TAILING FLOWRATE OF 1st CLEANER BANK (CU.M/Hr.)
39  :     TAIL4   = TAILING FLOWRATE OF 2nd CLEANER BANK (CU.M/Hr.)
40  :     TAIL5   = TAILING FLOWRATE OF 3rd CLEANER BANK (CU.M/Hr.)
41  :     CL      = FEED FLOWRATE OF ROUGHER BANK (CU.M/Hr.)
42  :     TAIL1   = FEED FLOWRATE OF SCAVENGER BANK (CU.M/Hr.)
43  :     CLL     = FEED FLOWRATE OF 1st CLEANER BANK (CU.M/Hr.)
44  :     LCL     = FEED FLOWRATE OF 2nd CLEANER BANK (CU.M/Hr.)
45  :     CONC4   = FEED FLOWRATE OF 3rd CLEANER BANK (CU.M/Hr.)
46  :     BCONC1  = CONCENTRATE FLOWRATE OF EACH COMPONENT OF EACH ROUGHER CELL
47  :     BCONC2  = CONCENTRATE FLOWRATE OF EACH COMPONENT OF EACH SCAVENGER CELL
48  :     BCONC3  = CONCENTRATE FLOWRATE OF EACH COMPONENT OF EACH 1st CLEANER CELL
49  :     BCONC4  = CONCENTRATE FLOWRATE OF EACH COMPONENT OF EACH 2nd CLEANER CELL
50  :     BCONC5  = CONCENTRATE FLOWRATE OF EACH COMPONENT OF EACH 3rd CLEANER CELL
```

```
51   :     SOLID1  =  CONCENTRATE SOLID FLOWRATE OF ROUGHER BANK (CU.M/Hr.)
52   :     SOLID2  =  CONCENTRATE SOLID FLOWRATE OF SCAVENGER BANK (CU.M/Hr.)
53   :     SOLID3  =  CONCENTRATE SOLID FLOWRATE OF 1st CLEANER BANK (CU.M/Hr.)
54   :     SOLID4  =  CONCENTRATE SOLID FLOWRATE OF 2nd CLEANER BANK (CU.M/Hr.)
55   :     SOLID5  =  CONCENTRATE SOLID FLOWRATE OF 3rd CLEANER BANK (CU.M/Hr.)
56   :     REC1    =  WEIGHT RECOVERY OF ROUGHER BANK
57   :     REC2    =  WEIGHT RECOVERY OF SCAVENGER BANK
58   :     REC3    =  WEIGHT RECOVERY OF 1st CLEANER BANK
59   :     REC4    =  WEIGHT RECOVERY OF 2nd CLEANER BANK
60   :     REC5    =  WEIGHT RECOVERY OF 3rd CLEANER BANK
61   :     WREC1   =  WATER RECOVERY OF ROUGHER BANK
62   :     WREC2   =  WATER RECOVERY OF SCAVENGER BANK
63   :     WREC3   =  WATER RECOVERY OF 1st CLEANER BANK
64   :     WREC4   =  WATER RECOVERY OF 2nd CLEANER BANK
65   :     WREC5   =  WATER RECOVERY OF 3rd CLEANER BANK
66   :     GRADE   =  PERCENT ASSAY OF FINAL CONCENTRATE
67   :     PCSLD   =  PERCENT SOLID
```

SMAIN4.FOR Tue 1 JAN 1980 01:25:59

```
C     MAIN PROGRAM FOR FLOTATION SIMULATION          RGSCAV.FOR
C     21 MARCH 1987
C     SOMSAK SAISINCHAI
C     ..................................................................
C     THE PROGRAM SIMULATES A FLOTATION CIRCUIT CONSISTING OF
C     ROUGHER BANK,SCAVENGER BANK,1ST CLEANER,2ND CLEANER,3 RD CLEANER
C     CONCENTRATE FROM SCAVENGER BANK IS CIRCULATED TO THE ROUGHERS.
C     ..................................................................
      CHARACTER COMP(10)*8,MTCOMP*8
      CHARACTER NAME*8,NNAME*20,SNAME*10
      REAL FEED(10),CONC(10),TAIL(10),CL(10),CONC2(10),TAIL1(10)
      REAL CONCN(10),CONCF(10),SAVE(10),FEED1(10),TEMP(100)
      REAL TCONC(10),TTAIL(10),CONC1(10),BCONC1(10,100)
      REAL BCONC2(10,100),SG(10),ASSAY(10),MASS(10),FMASS(10)
      REAL CMASS(10),TEMPC(10),TEMPF(10),MFEED,MCONC,TMETAL(10)
      REAL CLL(10),LCL(10),CONC3(10),TAIL3(10),CONC4(10),TAIL4(10)
      real ccl(10),ccc(10)
      REAL CONC5(10),TAIL5(10),BCONC3(10,100),BCONC4(10,100)
      REAL BCONC5(10,100),bconc6(10,100),conc6(10),tail6(10)
      COMMON/RATEC/PAR(2),RATEF(10),RATEN(10),RATEFR(10),RATENR(10)
      OPEN(1,FILE='FSIM1.DAT',STATUS='OLD')
      OPEN(2,FILE='OUT.DAT',STATUS='NEW',ACCESS='SEQUENTIAL',FORM=
     1'FORMATTED')
      CALL RDCOMP(N,COMP)
      CALL PTCOMP(N,COMP)
      CALL RDPULP(N,FEED,SG)
      CALL PTPULP('ROUGHER ',N,FEED)
      CALL RDASSA(N,MTCOMP,ASSAY)
      CALL PTASSA(N,MTCOMP,ASSAY)
      CALL RDCELL(N,PAR,RATEF,RATEN)
      CALL RDCELL(N,PAR,RATEFR,RATENR)
      CALL PTCELL(N,PAR,RATEF,RATEN)
      READ(1,*)NCELL1,NCELL2,NCELL3,NCELL4,NCELL5,ncell6
      DO 5 I=1,N+1
          CONC2(I)=0
   5  CONTINUE
      DO 40 K=1,30
           IND=0
           DO 10 I=1,N+1
                CL(I)=FEED(I)+CONC2(I)+TAIL3(I)
  10       CONTINUE
  20       CALL BANK(N,NCELL1,PAR,RATEFR,RATENR,CL,BCONC1,CONC1,TAIL1,
     1              SOLID1,REC1,WREC1)
C          TAILING FROM THE FIRST BANK BECOMES FEED TO THE NEXT
           CALL BANK(N,NCELL2,PAR,RATEF,RATEN,TAIL1,BCONC2,CONC2,TAIL,
     1              SOLID2,REC2,WREC2)
C          CONCENTRATE FROM ROUGHER BANK AND TAILING FROM 2ND CLEANER
C          BANK BECOMES FEED TO 1ST CLEANER
           DO 30 I=1,N+1
                CLL(I)=CONC1(I)+TAIL4(I)
  30       CONTINUE
           CALL BANK(N,NCELL3,PAR,RATEF,RATEN,CLL,BCONC3,CONC3,TAIL3,
     1              SOLID3,REC3,WREC3)
C          CONCENTRETE FROM 1ST CLEANER AND TAILING FROM 3RD CLEANER
C          BANK BECOMES FEED TO 2ND CLEANER
```

```
      DO 32 I=1,N+1
           LCL(I)=CONC3(I)+TAIL5(I)
 32   CONTINUE
      CALL BANK(N,NCELL4,PAR,RATEF,RATEN,LCL,BCONC4,CONC4,TAIL4,
     1          SOLID4,REC4,WREC4)
      do 33 i=1,n+1
           ccl(i)=conc4(i)+tail6(i)
 33   continue
C     CONCENTRATE FROM 2ND CLEANER BANK BECOMES FEED TO 3RD CLEANER
      CALL BANK(N,NCELL5,PAR,RATEF,RATEN,ccl,BCONC5,CONC5,TAIL5
     1          ,SOLID5,REC5,WREC5)
      call bank(n,ncell6,par,ratef,raten,conc5,bconc6,conc6,tail6,
     1          solid6,rec6,wrec6)
      CALL CONV(N,CONC2,SAVE,IND)
      IF(IND.EQ.0)GOTO 100
 40   CONTINUE
      WRITE(2,80)
 80   FORMAT('          ### WARNING DID NOT CONVERGE ###     ')
100   WRITE(2,110)K
110   FORMAT(//'###    END BANK CALCULATION    ###'/,
     1'NO.OF ITERATIONS'/,13X,I3)
      CALL PTBANK('ROUGHER   ',NCELL1)
      CALL PPBANK(N,NCELL1,COMP,BCONC1)
      CALL PTBANK('SCAVENG   ',NCELL2)
      CALL PPBANK(N,NCELL2,COMP,BCONC2)
      CALL PTBANK('1 CLEANER ',NCELL3)
      CALL PPBANK(N,NCELL3,COMP,BCONC3)
      CALL PTBANK('2 CLEANER ',NCELL4)
      CALL PPBANK(N,NCELL4,COMP,BCONC4)
      CALL PTBANK('3 CLEANER ',NCELL5)
      CALL PPBANK(N,NCELL5,COMP,BCONC5)
      call ptbank('4 CLEANER ',ncell6)
      call ppbank(n,ncell6,comp,bconc6)
      WRITE(2,170)(CONC6(I),I=1,N+1)
170   FORMAT(/'CONC.FLOW RATE FROM THE CIRCUIT'/,14X,F8.3,3X,9F8.3)
      WRITE(2,180)(TAIL(I),I=1,N+1)
180   FORMAT(/'TAILING FROM THE LAST BANK'/,14X,F8.3,3X,9F8.3)
      WRITE(2,190)(CL(I),I=1,N+1)
190   FORMAT(/'TOTAL FEED TO THE FIRST BANK'/,14X,F8.3,3X,9F8.3)
      CALL RCBANK(N,FEED,CONC6,SG)
      CALL GRAD(N,CONC6,ASSAY,SG,GRADR,PCSLD)
      CALL PTGRAD('FINAL   CONC.       ',GRADR,PCSLD)
      CALL MTREC(N,FEED,CONC6,SG,ASSAY)
      CALL GRAD(N,CONC1,ASSAY,SG,GRADER,PCSLDR)
      CALL PTGRAD('ROUGHER CONC.       ',GRADER,PCSLDR)
      CALL MTREC(N,CL,CONC1,SG,ASSAY)
      END
```

ภาคผนวก ข.

โปรแกรมหลักการจำลองแบบวงจรลอยแร่ บ.กระบี่ฟลูออไรท์จำกัด แบบที่ 2

ซึ่งเขียนด้วยภาษาฟอร์แทรน 77

```
1   *     THE MEAN OF VARIABLE FOR MAIN2.FOR
2   *     N       =  THE NUMBER OF COMPONENT
3   *     COMP    =  NAME OF COMPONENT IN FLOTATION CELL
4   *     COMP(1) =  WATER COMPONENT
5   *     COMP(2) =  FAST FLOATING COMPONENT
6   *     COMP(3) =  SLOW FLOATING COMPONENT
7   *     COMP(4) =  GANGUE COMPONENT
8   *     MTCOMP  =  NAME OF FLOATING ORE
9   *     ASSAY(2)=  ASSAY OF FAST FLOATING COMPONENT
10  *     ASSAY(3)=  ASSAY OF SLOW FLOATING COMPONENT
11  *     ASSAY(4)=  ASSAY OF GANGUE COMPONENT
12  *     PAR(1)  =  CELL'VOLUME (CU.M)
13  *     PAR(2)  =  BETA CONSTANT
14  *     SOLIDS  =  FEED RATE OF ORE (T/Hr.)
15  *     FEED(1) =  FEED RATE OF WATER (T/Hr.)
16  *     FEED(2) =  FEED RATE OF FAST FLOATING COMPONENT (% OF ORE FEED RATE)
17  *     FEED(3) =  FEED RATE OF SLOW FLOATING COMPONENT (% OF ORE FEED RATE)
18  *     FEED(4) =  FEED RATE OF GANGUE (% OF ORE FEED RATE)
19  *     SG(2)   =  SPECIFIC GRAVITY OF FAST FLOATING COMPONENT
20  *     SG(3)   =  SPECIFIC GRAVITY OF SLOW FLOATING COMPONENT
21  *     SG(4)   =  SPECIFIC GRAVITY OF GANGUE
22  *     RATEF(2)=  RATE CONSTANT OF FAST FLOATING COMPONENT
23  *     RATEF(3)=  RATE CONSTANT OF SLOW FLOATING COMPONENT
24  *     RATEN(1)=  RATE CONSTANT OF NON FLOATING COMPONENT (WATER)
25  *     RATEN(4)=  RATE CONSTANT OF NON FLOATING COMPONENT (GANGUE)
26  *     NCELL1  =  THE NUMBER OF ROUGHER CELLS
27  *     NCELL2  =  THE NUMBER OF SCAVENGER CELLS
28  *     NCELL3  =  THE NUMBER OF 1st CLEANER CELLS
29  *     NCELL4  =  THE NUMBER OF 2nd CLEANER CELLS
30  *     NCELL5  =  THE NUMBER OF 3rd CLEANER CELLS
31  *     NCELL6  =  THE NUMBER OF 4th CLEANER CELLS
32  *     CONC1   =  CONCENTRATE FLOWRATE OF ROUGHER BANK (CU.M/Hr.)
33  *     CONC2   =  CONCENTRATE FLOWRATE OF SCAVENGER BANK (CU.M/Hr.)
34  *     CONC3   =  CONCENTRATE FLOWRATE OF 1st CLEANER BANK (CU.M/Hr.)
35  *     CONC4   =  CONCENTRATE FLOWRATE OF 2nd CLEANER BANK (CU.M/Hr.)
36  *     CONC5   =  CONCENTRATE FLOWRATE OF 3rd CLEANER BANK (CU.M/Hr.)
37  *     CONC6   =  CONCENTRATE FLOWRATE OF 4th CLEANER BANK (CU.M/Hr.)
38  *     TAIL1   =  TAILING FLOWRATE OF ROUGHER BANK (CU.M/Hr.)
39  *     TAIL    =  TAILING FLOWRATE OF SCAVENGER BANK (CU.M/Hr.)
40  *     TAIL3   =  TAILING FLOWRATE OF 1st CLEANER BANK (CU.M/Hr.)
41  *     TAIL4   =  TAILING FLOWRATE OF 2nd CLEANER BANK (CU.M/Hr.)
42  *     TAIL5   =  TAILING FLOWRATE OF 3rd CLEANER BANK (CU.M/Hr.)
43  *     TAIL6   =  TAILING FLOWRATE OF 4th CLEANER BANK (CU.M/Hr.)
44  *     CL      =  FEED FLOWRATE OF ROUGHER BANK (CU.M/Hr.)
45  *     CCC     =  FEED FLOWRATE OF SCAVENGER BANK (CU.M/Hr.)
46  *     CLL     =  FEED FLOWRATE OF 1st CLEANER BANK (CU.M/Hr.)
47  *     LCL     =  FEED FLOWRATE OF 2nd CLEANER BANK (CU.M/Hr.)
48  *     CCL     =  FEED FLOWRATE OF 3rd CLEANER BANK (CU.M/Hr.)
49  *     CONC6   =  FEED FLOWRATE OF 4th CLEANER BANK (CU.M/Hr.)
50  *     BCONC1  =  CONCENTRATE FLOWRATE OF EACH COMPONENT OF EACH ROUGHER CELL
```

```
51  *     BCONC2  =  CONCENTRATE FLOWRATE OF EACH COMPONENT OF EACH SCAVENGER CELL
52  *     BCONC3  =  CONCENTRATE FLOWRATE OF EACH COMPONENT OF EACH 1st CLEANER CELL
53  *     BCONC4  =  CONCENTRATE FLOWRATE OF EACH COMPONENT OF EACH 2nd CLEANER CELL
54  *     BCONC5  =  CONCENTRATE FLOWRATE OF EACH COMPONENT OF EACH 3rd CLEANER CELL
55  *     BCONC6  =  CONCENTRATE FLOWRATE OF EACH COMPONENT OF EACH 4th CLEANER CELL
56  *     SOLID1  =  CONCENTRATE SOLID FLOWRATE OF ROUGHER BANK (CU.M/Hr.)
57  *     SOLID2  =  CONCENTRATE SOLID FLOWRATE OF SCAVENGER BANK (CU.M/Hr.)
58  *     SOLID3  =  CONCENTRATE SOLID FLOWRATE OF 1st CLEANER BANK (CU.M/Hr.)
59  *     SOLID4  =  CONCENTRATE SOLID FLOWRATE OF 2nd CLEANER BANK (CU.M/Hr.)
60  *     SOLID5  =  CONCENTRATE SOLID FLOWRATE OF 3rd CLEANER BANK (CU.M/Hr.)
61  *     SOLID6  =  CONCENTRATE SOLID FLOWRATE OF 4th CLEANER BANK (CU.M/Hr.)
62  *     REC1    =  WEIGHT RECOVERY OF ROUGHER BANK
63  *     REC2    =  WEIGHT RECOVERY OF SCAVENGER BANK
64  *     REC3    =  WEIGHT RECOVERY OF 1st CLEANER BANK
65  *     REC4    =  WEIGHT RECOVERY OF 2nd CLEANER BANK
66  *     REC5    =  WEIGHT RECOVERY OF 3rd CLEANER BANK
67  *     REC6    =  WEIGHT RECOVERY OF 4th CLEANER BANK
68  *     WREC1   =  WATER RECOVERY OF ROUGHER BANK
69  *     WREC2   =  WATER RECOVERY OF SCAVENGER BANK
70  *     WREC3   =  WATER RECOVERY OF 1st CLEANER BANK
71  *     WREC4   =  WATER RECOVERY OF 2nd CLEANER BANK
72  *     WREC5   =  WATER RECOVERY OF 3rd CLEANER BANK
73  *     WREC6   =  WATER RECOVERY OF 4th CLEANER BANK
74  *     GRADE   =  PERCENT ASSAY OF FINAL CONCENTRATE
75  *     PCSLD   =  PERCENT SOLID
```

```
C     MAIN PROGRAM FOR FLOTATION SIMULATION          KMAIN2.FOR
C     21 MARCH 1987
C     SOMSAK SAISINCHAI
C     ..............................................................
C     THE PROGRAM SIMULATES A FLOTATION CIRCUIT CONSISTING OF
C     ROUGHER BANK,SCAVENGER BANK,1ST,2ND,3RD CLEANER
C     1. ROUGHER'TAIL                          FEED TO SCAV.
C     2. SCAV.'CONC & 1ST CLEAN'TAIL           FEED TO ROUGHER
C     3. ROUGHER'CONC & 2ND CLEAN'TAIL         FEED TO 1ST CLEAN.
C     4. 1ST CLEAN'CONC.& 3RD CLEAN'TAIL       FEED TO 2ND CLEAN.
C     5. 2ND CLEAN'CONC.                       FEED TO 3RD CLEAN.
C     ..............................................................
      CHARACTER COMP(10)*8,MTCOMP*8
      CHARACTER NAME*8,NNAME*20,SNAME*10
      REAL FEED(10),CONC(10),TAIL(10),CL(10),CONC2(10),TAIL1(10)
      REAL CONCN(10),CONCF(10),SAVE(10),FEED1(10),TEMP(100)
      REAL TCONC(10),TTAIL(10),CONC1(10),BCONC1(10,100)
      REAL BCONC2(10,100),SG(10),ASSAY(10),MASS(10),FMASS(10)
      REAL CMASS(10),TEMPC(10),TEMPF(10),MFEED,MCONC,TMETAL(10)
      REAL CLL(10),LCL(10),CONC3(10),TAIL3(10),CONC4(10),TAIL4(10)
      REAL CONC5(10),TAIL5(10),BCONC3(10,100),BCONC4(10,100)
      REAL BCONC5(10,100)
      COMMON/RATEC/PAR(2),RATEF(10),RATEN(10),RATEFR(10),RATENR(10)
      OPEN(1,FILE='FSIM1.DAT',STATUS='OLD')
      OPEN(2,FILE='OUT.DAT',STATUS='NEW',ACCESS='SEQUENTIAL',FORM=
     1'FORMATTED')
      CALL RDCOMP(N,COMP)
      CALL PTCOMP(N,COMP)
      CALL RDPULP(N,FEED,SG)
      CALL PTPULP('ROUGHER ',N,FEED)
      CALL RDASSA(N,MTCOMP,ASSAY)
      CALL PTASSA(N,MTCOMP,ASSAY)
      CALL RDCELL(N,PAR,RATEF,RATEN)
      CALL RDCELL(N,PAR,RATEFR,RATENR)
      CALL PTCELL(N,PAR,RATEF,RATEN)
      READ(1,*)NCELL1,NCELL2,NCELL3,NCELL4,NCELL5
      DO 5 I=1,N+1
          CONC2(I)=0
    5 CONTINUE
      DO 40 K=1,30
          IND=0
          DO 10 I=1,N+1
              CL(I)=FEED(I)+CONC2(I)+TAIL3(I)
   10     CONTINUE
   20     CALL BANK(N,NCELL1,PAR,RATEFR,RATENR,CL,BCONC1,CONC1,TAIL1,
     1              SOLID1,REC1,WREC1)
C         TAILING FROM THE FIRST BANK BECOMES FEED TO THE NEXT
          CALL BANK(N,NCELL2,PAR,RATEF,RATEN,TAIL1,BCONC2,CONC2,TAIL,
     1              SOLID2,REC2,WREC2)
C         CONCENTRATE FROM ROUGHER BANK AND TAILING FROM 2ND CLEANER
C         BANK BECOMES FEED TO 1ST CLEANER
          DO 30 I=1,N+1
              CLL(I)=CONC1(I)+TAIL4(I)
   30     CONTINUE
          CALL BANK(N,NCELL3,PAR,RATEF,RATEN,CLL,BCONC3,CONC3,TAIL3,
```

KMAIN2.FOR Mon 27 FEB 1989 11:04:03

```
     1               SOLID3,REC3,WREC3)
C          CONCENTRETE FROM 1ST CLEANER AND TAILING FROM 3RD CLEANER
C          BANK BECOMES FEED TO 2ND CLEANER
           DO 32 I=1,N+1
                LCL(I)=CONC3(I)+TAIL5(I)
32         CONTINUE
           CALL BANK(N,NCELL4,PAR,RATEF,RATEN,LCL.BCONC4,CONC4,TAIL4,
     1               SOLID4,REC4,WREC4)
C          CONCENTRATE FROM 2ND CLEANER BANK BECOMES FEED TO 3RD CLEANER
           CALL BANK(N,NCELL5,PAR,RATEF,RATEN,CONC4,BCONC5,CONC5,TAIL5
     1               ,SOLID5,REC5,WREC5)
           CALL CONV(N,CONC2,SAVE,IND)
           IF(IND.EQ.0)GOTO 100
40    CONTINUE
      WRITE(2,80)
80    FORMAT('          ***  WARNING DID NOT CONVERGE  ***     ')
100   WRITE(2,110)K
110   FORMAT(///'***    END BANK CALCULATION    ***'/,
     1'NO.OF ITERATIONS'/,13X,I3)
      CALL PTBANK('ROUGHER   ',NCELL1)
      CALL PPBANK(N,NCELL1,COMP,BCONC1)
      CALL PTBANK('SCAVENG   ',NCELL2)
      CALL PPBANK(N,NCELL2,COMP,BCONC2)
      CALL PTBANK('1 CLEANER ',NCELL3)
      CALL PPBANK(N,NCELL3,COMP,BCONC3)
      CALL PTBANK('2 CLEANER ',NCELL4)
      CALL PPBANK(N,NCELL4,COMP,BCONC4)
      CALL PTBANK('3 CLEANER ',NCELL5)
      CALL PPBANK(N,NCELL5,COMP,BCONC5)
      WRITE(2,170)(CONC5(I),I=1,N+1)
170   FORMAT(/'CONC.FLOW RATE FROM THE CIRCUIT'/,14X,F8.3,3X,9F8.3)
      WRITE(2,180)(TAIL(I),I=1,N+1)
180   FORMAT(/'TAILING FROM THE LAST BANK'/,14X,F8.3,3X,9F8.3)
      WRITE(2,190)(CL(I),I=1,N+1)
190   FORMAT(/'TOTAL FEED TO THE FIRST BANK'/,14X,F8.3,3X,9F8.3)
      CALL RCBANK(N,FEED,CONC5,SG)
      CALL GRAD(N,CONC5,ASSAY,SG,GRADE,PCSLD)
      CALL PTGRAD('FINAL   CONC.       ',GRADE,PCSLD)
      CALL MTREC(N,FEED,CONC5,SG,ASSAY)
      CALL GRAD(N,CONC1,ASSAY,SG,GRADER,PCSLDR)
      CALL PTGRAD('ROUGHER CONC.       ',GRADER,PCSLDR)
      CALL MTREC(N,CL,CONC1,SG,ASSAY)
      END
```

ภาคผนวก ค.

โปรแกรมหลักการจำลองแบบวงจรลอยแร่ บ.กระบี่ฟลูออไรท์จำกัด แบบที่ 3

ซึ่งเขียนด้วยภาษาฟอร์แทรน 77

```
C     THE MEAN OF VARIABLE FOR KMAIN3.FOR
C     N       = THE NUMBER OF COMPONENT
C     COMP    = NAME OF COMPONENT IN FLOTATION CELL
C     COMP(1) = WATER COMPONENT
C     COMP(2) = FAST FLOATING COMPONENT
C     COMP(3) = SLOW FLOATING COMPONENT
C     COMP(4) = GANGUE COMPONENT
C     MTCOMP  = NAME OF FLOATING ORE
C     ASSAY(2)= ASSAY OF FAST FLOATING COMPONENT
C     ASSAY(3)= ASSAY OF SLOW FLOATING COMPONENT
C     ASSAY(4)= ASSAY OF GANGUE COMPONENT
C     PAR(1)  = CELL'VOLUME (CU.M)
C     PAR(2)  = BETA CONSTANT
C     SOLIDS  = FEED RATE OF ORE (T/Hr.)
C     FEED(1) = FEED RATE OF WATER (T/Hr.)
C     FEED(2) = FEED RATE OF FAST FLOATING COMPONENT (% OF ORE FEED RATE)
C     FEED(3) = FEED RATE OF SLOW FLOATING COMPONENT (% OF ORE FEED RATE)
C     FEED(4) = FEED RATE OF GANGUE (% OF ORE FEED RATE)
C     SG(2)   = SPECIFIC GRAVITY OF FAST FLOATING COMPONENT
C     SG(3)   = SPECIFIC GRAVITY OF SLOW FLOATING COMPONENT
C     SG(4)   = SPECIFIC GRAVITY OF GANGUE
C     RATEF(2)= RATE CONSTANT OF FAST FLOATING COMPONENT
C     RATEF(3)= RATE CONSTANT OF SLOW FLOATING COMPONENT
C     RATEN(1)= RATE CONSTANT OF NON FLOATING COMPONENT (WATER)
C     RATEN(4)= RATE CONSTANT OF NON FLOATING COMPONENT (GANGUE)
C     NCELL1  = THE NUMBER OF ROUGHER CELLS
C     NCELL2  = THE NUMBER OF SCAVENGER CELLS
C     NCELL3  = THE NUMBER OF 1st CLEANER CELLS
C     NCELL4  = THE NUMBER OF 2nd CLEANER CELLS
C     NCELL5  = THE NUMBER OF 3rd CLEANER CELLS
C     NCELL6  = THE NUMBER OF 4th CLEANER CELLS
C     CONC1   = CONCENTRATE FLOWRATE OF ROUGHER BANK (CU.M/Hr.)
C     CONC2   = CONCENTRATE FLOWRATE OF SCAVENGER BANK (CU.M/Hr.)
C     CONC3   = CONCENTRATE FLOWRATE OF 1st CLEANER BANK (CU.M/Hr.)
C     CONC4   = CONCENTRATE FLOWRATE OF 2nd CLEANER BANK (CU.M/Hr.)
C     CONC5   = CONCENTRATE FLOWRATE OF 3rd CLEANER BANK (CU.M/Hr.)
C     CONC6   = CONCENTRATE FLOWRATE OF 4th CLEANER BANK (CU.M/Hr.)
C     TAIL1   = TAILING FLOWRATE OF ROUGHER BANK (CU.M/Hr.)
C     TAIL    = TAILING FLOWRATE OF SCAVENGER BANK (CU.M/Hr.)
C     TAIL3   = TAILING FLOWRATE OF 1st CLEANER BANK (CU.M/Hr.)
C     TAIL4   = TAILING FLOWRATE OF 2nd CLEANER BANK (CU.M/Hr.)
C     TAIL5   = TAILING FLOWRATE OF 3rd CLEANER BANK (CU.M/Hr.)
C     TAIL6   = TAILING FLOWRATE OF 4th CLEANER BANK (CU.M/Hr.)
C     CL      = FEED FLOWRATE OF ROUGHER BANK (CU.M/Hr.)
C     CCC     = FEED FLOWRATE OF SCAVENGER BANK (CU.M/Hr.)
C     CLL     = FEED FLOWRATE OF 1st CLEANER BANK (CU.M/Hr.)
C     LCL     = FEED FLOWRATE OF 2nd CLEANER BANK (CU.M/Hr.)
C     CCL     = FEED FLOWRATE OF 3rd CLEANER BANK (CU.M/Hr.)
C     CONC5   = FEED FLOWRATE OF 4th CLEANER BANK (CU.M/Hr.)
C     BCONC1  = CONCENTRATE FLOWRATE OF EACH COMPONENT OF EACH ROUGHER CELL
```

```
51  {     BCONC2  =  CONCENTRATE FLOWRATE OF EACH COMPONENT OF EACH SCAVENGER CELL
52  {     BCONC3  =  CONCENTRATE FLOWRATE OF EACH COMPONENT OF EACH 1st CLEANER CELL
53  {     BCONC4  =  CONCENTRATE FLOWRATE OF EACH COMPONENT OF EACH 2nd CLEANER CELL
54  {     BCONC5  =  CONCENTRATE FLOWRATE OF EACH COMPONENT OF EACH 3rd CLEANER CELL
55  {     BCONC6  =  CONCENTRATE FLOWRATE OF EACH COMPONENT OF EACH 4th CLEANER CELL
56  {     SOLID1  =  CONCENTRATE SOLID FLOWRATE OF ROUGHER BANK (CU.M/Hr.)
57  {     SOLID2  =  CONCENTRATE SOLID FLOWRATE OF SCAVENGER BANK (CU.M/Hr.)
58  {     SOLID3  =  CONCENTRATE SOLID FLOWRATE OF 1st CLEANER BANK (CU.M/Hr.)
59  {     SOLID4  =  CONCENTRATE SOLID FLOWRATE OF 2nd CLEANER BANK (CU.M/Hr.)
60  {     SOLID5  =  CONCENTRATE SOLID FLOWRATE OF 3rd CLEANER BANK (CU.M/Hr.)
61  {     SOLID6  =  CONCENTRATE SOLID FLOWRATE OF 4th CLEANER BANK (CU.M/Hr.)
62  {     REC1    =  WEIGHT RECOVERY OF ROUGHER BANK
63  {     REC2    =  WEIGHT RECOVERY OF SCAVENGER BANK
64  {     REC3    =  WEIGHT RECOVERY OF 1st CLEANER BANK
65  {     REC4    =  WEIGHT RECOVERY OF 2nd CLEANER BANK
66  {     REC5    =  WEIGHT RECOVERY OF 3rd CLEANER BANK
67  {     REC6    =  WEIGHT RECOVERY OF 4th CLEANER BANK
68  {     WREC1   =  WATER RECOVERY OF ROUGHER BANK
69  {     WREC2   =  WATER RECOVERY OF SCAVENGER BANK
70  {     WREC3   =  WATER RECOVERY OF 1st CLEANER BANK
71  {     WREC4   =  WATER RECOVERY OF 2nd CLEANER BANK
72  {     WREC5   =  WATER RECOVERY OF 3rd CLEANER BANK
73  {     WREC6   =  WATER RECOVERY OF 4th CLEANER BANK
74  {     GRADE   =  PERCENT ASSAY OF FINAL CONCENTRATE
75  {     PCSLD   =  PERCENT SOLID
```

```
C     MAIN PROGRAM FOR FLOTATION SIMULATION          KMAIN3.FOR
C     21 MARCH 1987
C     SOMSAK SAISINCHAI
C     ..............................................................
C     THR PROGRAM SIMULATES A FLOTATION CIRCUIT CONSISTING OF
C     ROUGHER BANK,SCAVENGER BANK,1ST,2ND,3RD,4TH CLEANER BANK
C     1. 1ST CLEAN'TAIL                          FEED TO SCAV.
C     2. ROUGHER&SCAV.'CONC. AND 2ND CLEAN'TAIL FEED TO 1ST CLEAN.
C     3. 1ST CLEAN'CONC.& 3RD CLEAN'TAIL         FEED TO 2ND CLEAN.
C     4. 2ND CLEAN'CONC.& 4TH CLEAN'TAIL         FEED TO 3RD CLEAN.
C     5. 3RD CLEAN'CONC.                         FEED TO 4TH CLEAN.
C     ..............................................................
      CHARACTER COMP(10)*8,MTCOMP*8
      CHARACTER NAME*8,NNAME*20,SNAME*10
      REAL FEED(10),CONC(10),TAIL(10),CL(10),CONC2(10),TAIL1(10)
      REAL CONCN(10),CONCF(10),SAVE(10),FEED1(10),TEMP(100)
      REAL TCONC(10),TTAIL(10),CONC1(10),BCONC1(10,100)
      REAL BCONC2(10,100),SG(10),ASSAY(10),MASS(10),FMASS(10)
      REAL CMASS(10),TEMPC(10),TEMPF(10),MFEED,MCONC,TMETAL(10)
      REAL CLL(10),LCL(10),CONC3(10),TAIL3(10),CONC4(10),TAIL4(10)
      REAL CCL(10),CCC(10)
      REAL CONC5(10),TAIL5(10),BCONC3(10,100),BCONC4(10,100),TAIL6(10)
      REAL BCONC5(10,100),BCONC6(10,100),CONC6(10)
      COMMON/RATEC/PAR(2),RATEF(10),RATEN(10),RATEFR(10),RATENR(10)
      OPEN(1,FILE='FSIM1.DAT',STATUS='OLD')
      OPEN(2,FILE='OUT.DAT',STATUS='NEW',ACCESS='SEQUENTIAL',FORM=
     1'FORMATTED')
      CALL RDCOMP(N,COMP)
      CALL PTCOMP(N,COMP)
      CALL RDPULP(N,FEED,SG)
      CALL PTPULP('ROUGHER ',N,FEED)
      CALL RDASSA(N,MTCOMP,ASSAY)
      CALL PTASSA(N,MTCOMP,ASSAY)
      CALL RDCELL(N,PAR,RATEF,RATEN)
      CALL RDCELL(N,PAR,RATEFR,RATENR)
      CALL PTCELL(N,PAR,RATEF,RATEN)
      READ(1,*)NCELL1,NCELL2,NCELL3,NCELL4,NCELL5,NCELL6
      DO 5 I=1,N+1
           CONC2(I)=0
           TAIL4(I)=0
           TAIL5(I)=0
           TAIL6(I)=0
 5    CONTINUE
      DO 40 K=1,30
           write(*,6)k
 6         format(/' Iteration on  ',i3)
           IND=0
           DO 10 I=1,N+1
                CL(I)=FEED(I)
 10        CONTINUE
 20        CALL BANK(N,NCELL1,PAR,RATEFR,RATENR,CL,BCONC1,CONC1,TAIL1,
     1               SOLID1,REC1,WREC1)
           write(*,1111)
1111        format(/' rougher ok')
C          CONCENTRATE FROM ROUGHER & SCAVENGER BANK AND TAILING FROM
```

```
      C           2ND CLEANER BANK BECOMES FEED TO 1ST CLEANER
                  DO 30 I=1,N+1
                       CLL(I)=CONC1(I)+TAIL4(I)+CONC2(I)
       30         CONTINUE
                  CALL BANK(N,NCELL3,PAR,RATEF,RATEN,CLL,BCONC3,CONC3,TAIL3,
     1                    SOLID3,REC3,WREC3)
                  write(*,1112)
      1112        format(/' 1 st ok')
      C           TAILING FROM  1ST BANK BECOMES FEED TO SCA.
                  DO 25 I=1,N+1
                       CCC(I)=TAIL3(I)
       25         CONTINUE
                  CALL BANK(N,NCELL2,PAR,RATEF,RATEN,CCC,BCONC2,CONC2,TAIL,
     1                    SOLID2,REC2,WREC2)
                  write(*,1113)
      1113        format(/' scavenger ok')
      C           CONCENTRETE FROM 1ST CLEANER AND TAILING FROM 3RD CLEANER
      C           BANK BECOMES FEED TO 2ND CLEANER
                  DO 32 I=1,N+1
                       LCL(I)=CONC3(I)+TAIL5(I)
       32         CONTINUE
                  CALL BANK(N,NCELL4,PAR,RATEF,RATEN,LCL,BCONC4,CONC4,TAIL4,
     1                    SOLID4,REC4,WREC4)
                  write(*,1114)
      1114        format(/' 2nd ok')
      C           CONC. FROM 2ND & TAIL FROM 4TH BANK BECOMES FEED TO 3RD CLEANER
                  DO 38 I=1,N+1
                       CCL(I)=CONC4(I)+TAIL6(I)
       38         CONTINUE
                  CALL BANK(N,NCELL5,PAR,RATEF,RATEN,CCL,BCONC5,CONC5,TAIL5
     1                   ,SOLID5,REC5,WREC5)
                  write(*,1115)
      1115        format(/' 3rd ok')
      C           CONC.FROM 3RD BANK BECOMES FEED TO 4TH CLEANER
                  CALL BANK(N,NCELL6,PAR,RATEF,RATEN,CONC5,BCONC6,CONC6,TAIL6
     1                   ,SOLID6,REC6,WREC6)
                  write(*,1116)
      1116        format(/' 4th ok')
                  CALL CONV(N,CONC3,SAVE,IND)
                  IF(IND.EQ.0)GOTO 100
       40    CONTINUE
             WRITE(2,80)
       80    FORMAT('         *** WARNING DID NOT CONVERGE ***     ')
      100    WRITE(2,110)K
      110    FORMAT(//'***    END BANK CALCULATION    ***'/,
     1      1'NO.OF ITERATIONS'/,13X,I3)
             CALL PTBANK('ROUGHER   ',NCELL1)
             CALL PPBANK(N,NCELL1,COMP,BCONC1)
             CALL PTBANK('SCAVENGER ',NCELL2)
             CALL PPBANK(N,NCELL2,COMP,BCONC2)
             CALL PTBANK('1 CLEANER ',NCELL3)
             CALL PPBANK(N,NCELL3,COMP,BCONC3)
             CALL PTBANK('2 CLEANER ',NCELL4)
             CALL PPBANK(N,NCELL4,COMP,BCONC4)
             CALL PTBANK('3 CLEANER ',NCELL5)
```

KMAIN3.FOR Mon 13 FEB 1989 15:27:40

```
      CALL PPBANK(N,NCELL5,COMP,BCONC5)
      CALL PTBANK('4 CLEANER ',NCELL6)
      CALL PPBANK(N,NCELL6,COMP,BCONC6)
      WRITE(2,170)(CONC6(I),I=1,N+1)
170   FORMAT(/'CONC.FLOW RATE FROM THE CIRCUIT'/,14X,F8.3,3X,9F8.3)
      WRITE(2,180)(TAIL(I),I=1,N+1)
180   FORMAT(/'TAILING FROM THE LAST BANK-SCAVENGER'/,14X,F8.3,3X,9F8.3)
      WRITE(2,185)(TAIL1(I),I=1,N+1)
185   FORMAT(/'TAILING FROM THE LAST BANK-ROUGHER'/,14X,F8.3,3X,9F8.3)
      WRITE(2,190)(CL(I),I=1,N+1)
190   FORMAT(/'TOTAL FEED TO THE FIRST BANK'/,14X,F8.3,3X,9F8.3)
      CALL RCBANK(N,FEED,CONC6,SG)
      CALL GRAD(N,CONC6,ASSAY,SG,GRADE,PCSLD)
      CALL PTGRAD('FINAL   CONC.       ',GRADE,PCSLD)
      CALL MTREC(N,FEED,CONC6,SG,ASSAY)
      CALL GRAD(N,CONC1,ASSAY,SG,GRADER,PCSLDR)
      CALL PTGRAD('ROUGHER CONC.       ',GRADER,PCSLDR)
      CALL MTREC(N,CL,CONC1,SG,ASSAY)
      END
```

ภาคผนวก ง.

โปรแกรมหลักการจำลองแบบวงจรลอยแร่ บ.กระบี่ฟลูออไรท์จำกัด แบบที่ 4

ซึ่งเขียนด้วยภาษาฟอร์แทรน 77

```
1  :    THE MEAN OF VARIABLE FOR KMAIN4.FOR
2  :    N       =  THE NUMBER OF COMPONENT
3  :    COMP    =  NAME OF COMPONENT IN FLOTATION CELL
4  :    COMP(1) =  WATER COMPONENT
5  :    COMP(2) =  FAST FLOATING COMPONENT
6  :    COMP(3) =  SLOW FLOATING COMPONENT
7  :    COMP(4) =  GANGUE COMPONENT
8  :    MTCOMP  =  NAME OF FLOATING ORE
9  :    ASSAY(2)=  ASSAY OF FAST FLOATING COMPONENT
10 :    ASSAY(3)=  ASSAY OF SLOW FLOATING COMPONENT
11 :    ASSAY(4)=  ASSAY OF GANGUE COMPONENT
12 :    PAR(1)  =  CELL'VOLUME (CU.M)
13 :    PAR(2)  =  BETA CONSTANT
14 :    SOLIDS  =  FEED RATE OF ORE (T/Hr.)
15 :    FEED(1) =  FEED RATE OF WATER (T/Hr.)
16 :    FEED(2) =  FEED RATE OF FAST FLOATING COMPONENT (% OF ORE FEED RATE)
17 :    FEED(3) =  FEED RATE OF SLOW FLOATING COMPONENT (% OF ORE FEED RATE)
18 :    FEED(4) =  FEED RATE OF GANGUE (% OF ORE FEED RATE)
19 :    SG(2)   =  SPECIFIC GRAVITY OF FAST FLOATING COMPONENT
20 :    SG(3)   =  SPECIFIC GRAVITY OF SLOW FLOATING COMPONENT
21 :    SG(4)   =  SPECIFIC GRAVITY OF GANGUE
22 :    RATEF(2)=  RATE CONSTANT OF FAST FLOATING COMPONENT
23 :    RATEF(3)=  RATE CONSTANT OF SLOW FLOATING COMPONENT
24 :    RATEN(1)=  RATE CONSTANT OF NON FLOATING COMPONENT (WATER)
25 :    RATEN(4)=  RATE CONSTANT OF NON FLOATING COMPONENT (GANGUE)
26 :    NCELL1  =  THE NUMBER OF ROUGHER CELLS
27 :    NCELL2  =  THE NUMBER OF SCAVENGER CELLS
28 :    NCELL3  =  THE NUMBER OF 1st CLEANER CELLS
29 :    NCELL4  =  THE NUMBER OF 2nd CLEANER CELLS
30 :    NCELL5  =  THE NUMBER OF 3rd CLEANER CELLS
31 :    NCELL6  =  THE NUMBER OF 4th CLEANER CELLS
32 :    CONC1   =  CONCENTRATE FLOWRATE OF ROUGHER BANK (CU.M/Hr.)
33 :    CONC2   =  CONCENTRATE FLOWRATE OF SCAVENGER BANK (CU.M/Hr.)
34 :    CONC3   =  CONCENTRATE FLOWRATE OF 1st CLEANER BANK (CU.M/Hr.)
35 :    CONC4   =  CONCENTRATE FLOWRATE OF 2nd CLEANER BANK (CU.M/Hr.)
36 :    CONC5   =  CONCENTRATE FLOWRATE OF 3rd CLEANER BANK (CU.M/Hr.)
37 :    CONC6   =  CONCENTRATE FLOWRATE OF 4th CLEANER BANK (CU.M/Hr.)
38 :    TAIL1   =  TAILING FLOWRATE OF ROUGHER BANK (CU.M/Hr.)
39 :    TAIL    =  TAILING FLOWRATE OF SCAVENGER BANK (CU.M/Hr.)
40 :    TAIL3   =  TAILING FLOWRATE OF 1st CLEANER BANK (CU.M/Hr.)
41 :    TAIL4   =  TAILING FLOWRATE OF 2nd CLEANER BANK (CU.M/Hr.)
42 :    TAIL5   =  TAILING FLOWRATE OF 3rd CLEANER BANK (CU.M/Hr.)
43 :    TAIL6   =  TAILING FLOWRATE OF 4th CLEANER BANK (CU.M/Hr.)
44 :    CL      =  FEED FLOWRATE OF ROUGHER BANK (CU.M/Hr.)
45 :    CCC     =  FEED FLOWRATE OF SCAVENGER BANK (CU.M/Hr.)
46 :    CLL     =  FEED FLOWRATE OF 1st CLEANER BANK (CU.M/Hr.)
47 :    LCL     =  FEED FLOWRATE OF 2nd CLEANER BANK (CU.M/Hr.)
48 :    CCL     =  FEED FLOWRATE OF 3rd CLEANER BANK (CU.M/Hr.)
49 :    CONC5   =  FEED FLOWRATE OF 4th CLEANER BANK (CU.M/Hr.)
50 :    BCONC1  =  CONCENTRATE FLOWRATE OF EACH COMPONENT OF EACH ROUGHER CELL
```

```
51  {     BCONC2  =  CONCENTRATE FLOWRATE OF EACH COMPONENT OF EACH SCAVENGER CELL
52  {     BCONC3  =  CONCENTRATE FLOWRATE OF EACH COMPONENT OF EACH 1st CLEANER CELL
53  {     BCONC4  =  CONCENTRATE FLOWRATE OF EACH COMPONENT OF EACH 2nd CLEANER CELL
54  {     BCONC5  =  CONCENTRATE FLOWRATE OF EACH COMPONENT OF EACH 3rd CLEANER CELL
55  {     BCONC6  =  CONCENTRATE FLOWRATE OF EACH COMPONENT OF EACH 4th CLEANER CELL
56  {     SOLID1  =  CONCENTRATE SOLID FLOWRATE OF ROUGHER BANK (CU.M/Hr.)
57  {     SOLID2  =  CONCENTRATE SOLID FLOWRATE OF SCAVENGER BANK (CU.M/Hr.)
58  {     SOLID3  =  CONCENTRATE SOLID FLOWRATE OF 1st CLEANER BANK (CU.M/Hr.)
59  {     SOLID4  =  CONCENTRATE SOLID FLOWRATE OF 2nd CLEANER BANK (CU.M/Hr.)
60  {     SOLID5  =  CONCENTRATE SOLID FLOWRATE OF 3rd CLEANER BANK (CU.M/Hr.)
61  {     SOLID6  =  CONCENTRATE SOLID FLOWRATE OF 4th CLEANER BANK (CU.M/Hr.)
62  {     REC1    =  WEIGHT RECOVERY OF ROUGHER BANK
63  {     REC2    =  WEIGHT RECOVERY OF SCAVENGER BANK
64  {     REC3    =  WEIGHT RECOVERY OF 1st CLEANER BANK
65  {     REC4    =  WEIGHT RECOVERY OF 2nd CLEANER BANK
66  {     REC5    =  WEIGHT RECOVERY OF 3rd CLEANER BANK
67  {     REC6    =  WEIGHT RECOVERY OF 4th CLEANER BANK
68  {     WREC1   =  WATER RECOVERY OF ROUGHER BANK
69  {     WREC2   =  WATER RECOVERY OF SCAVENGER BANK
70  {     WREC3   =  WATER RECOVERY OF 1st CLEANER BANK
71  {     WREC4   =  WATER RECOVERY OF 2nd CLEANER BANK
72  {     WREC5   =  WATER RECOVERY OF 3rd CLEANER BANK
73  {     WREC6   =  WATER RECOVERY OF 4th CLEANER BANK
74  {     GRADE   =  PERCENT ASSAY OF FINAL CONCENTRATE
75  {     PCSLD   =  PERCENT SOLID
```

KMAIN4.FOR Mon 27 FEB 1989 11:15:58

```
C     MAIN PROGRAM FOR FLOTATION SIMULATION          KMAIN4.FOR
C     21 MARCH 1987
C     SOMSAK SAISINCHAI
C     ..............................................................
C     THE PROGRAM SIMULATES A FLOTATION CIRCUIT CONSISTING OF
C     ROUGHER BANK,SCAVENGER BANK,1ST,2ND,3RD,4TH CLEANER BANK
C     1. ROUGHER'TAIL & 1ST CLEAN'TAIL      FEED TO SCAV.
C     2. SCAV.'CONC. & NEW FEED             FEED TO ROUGHER
C     3. ROUGHER'CONC.& 2ND CLEAN'TAIL      FEED TO 1ST CLEAN.
C     4. 1ST CLEAN'CONC.& 3RD CLEAN'TAIL    FEED TO 2ND CLEAN.
C     5. 2ND CLEAN'CONC.& 4TH CLEAN'TAIL    FEED TO 3RD CLEAN.
C     6. 3RD CLEAN'CONC.                    FEED TO 4TH CLEAN.
C     ..............................................................
      CHARACTER COMP(10)*8,MTCOMP*8
      CHARACTER NAME*8,NNAME*20,SNAME*10
      REAL FEED(10),CONC(10),TAIL(10),CL(10),CONC2(10),TAIL1(10)
      REAL CONCN(10),CONCF(10),SAVE(10),FEED1(10),TEMP(100)
      REAL TCONC(10),TTAIL(10),CONC1(10),BCONC1(10,100)
      REAL BCONC2(10,100),SG(10),ASSAY(10),MASS(10),FMASS(10)
      REAL CMASS(10),TEMPC(10),TEMPF(10),MFEED,MCONC,TMETAL(10)
      REAL CLL(10),LCL(10),CONC3(10),TAIL3(10),CONC4(10),TAIL4(10)
      REAL CCL(10),CCC(10)
      REAL CONC5(10),TAIL5(10),BCONC3(10,100),BCONC4(10,100),TAIL6(10)
      REAL BCONC5(10,100),BCONC6(10,100),CONC6(10),
      COMMON/RATEC/PAR(2),RATEF(10),RATEN(10),RATEFR(10),RATENR(10)
      OPEN(1,FILE='FSIM1.DAT',STATUS='OLD')
      OPEN(2,FILE='OUT.DAT',STATUS='NEW',ACCESS='SEQUENTIAL',FORM=
     1'FORMATTED')
      CALL RDCOMP(N,COMP)
      CALL PTCOMP(N,COMP)
      CALL RDPULP(N,FEED,SG)
      CALL PTPULP('ROUGHER ',N,FEED)
      CALL RDASSA(N,MTCOMP,ASSAY)
      CALL PTASSA(N,MTCOMP,ASSAY)
      CALL RDCELL(N,PAR,RATEF,RATEN)
      CALL RDCELL(N,PAR,RATEFR,RATENR)
      CALL PTCELL(N,PAR,RATEF,RATEN)
      READ(1,*)NCELL1,NCELL2,NCELL3,NCELL4,NCELL5,NCELL6
      DO 5 I=1,N+1
          CONC2(I)=0
 5    CONTINUE
      DO 40 K=1,30
           IND=0
           DO 10 I=1,N+1
                CL(I)=FEED(I)
10         CONTINUE
20         CALL BANK(N,NCELL1,PAR,RATEFR,RATENR,CL,BCONC1,CONC1,TAIL1,
     1               SOLID1,REC1,WREC1)

C          TAILING FROM THE ROUGHER & 1ST BANK BECOMES FEED TO SCA.
           DO 25 I=1,N+1
                CCC(I)=TAIL1(I)+TAIL3(I)
25         CONTINUE
           CALL BANK(N,NCELL2,PAR,RATEF,RATEN,CCC,BCONC2,CONC2,TAIL,
     1               SOLID2,REC2,WREC2)
```

KMAIN4.FOR Mon 27 FEB 1989 11:15:58

```
C          CONCENTRATE FROM ROUGHER BANK AND TAILING FROM 2ND CLEANER
C          BANK BECOMES FEED TO 1ST CLEANER
           DO 30 I=1,N+1
                CLL(I)=CONC1(I)+TAIL4(I)+conc2(i)
 30        CONTINUE
           CALL BANK(N,NCELL3,PAR,RATEF,RATEN,CLL,BCONC3,CONC3,TAIL3,
     1               SOLID3,REC3,WREC3)

C          CONCENTRETE FROM 1ST CLEANER AND TAILING FROM 3RD CLEANER
C          BANK BECOMES FEED TO 2ND CLEANER
           DO 32 I=1,N+1
                LCL(I)=CONC3(I)+TAIL5(I)
 32        CONTINUE
           CALL BANK(N,NCELL4,PAR,RATEF,RATEN,LCL,BCONC4,CONC4,TAIL4,
     1               SOLID4,REC4,WREC4)

C          CONC. FROM 2ND & TAIL FROM 4TH BANK BECOMES FEED TO 3RD CLEANER
           DO 38 I=1,N+1
                CCL(I)=CONC4(I)+TAIL6(I)
 38        CONTINUE
           CALL BANK(N,NCELL5,PAR,RATEF,RATEN,CCL,BCONC5,CONC5,TAIL5
     1               ,SOLID5,REC5,WREC5)

C          CONC.FROM 3RD BANK BECOMES FEED TO 4TH CLEANER
           CALL BANK(N,NCELL6,PAR,RATEF,RATEN,CONC5,BCONC6,CONC6,TAIL6
     1               ,SOLID6,REC6,WREC6)
           CALL CONV(N,CONC2,SAVE,IND)
           IF(IND.EQ.0)GOTO 100
 40    CONTINUE
       WRITE(2,80)
 80    FORMAT('          ### WARNING DID NOT CONVERGE ###     ')
100    WRITE(2,110)K
110    FORMAT(//'###    END BANK CALCULATION    ###'/,
      1'NO.OF ITERATIONS'/,13X,I3)
       CALL PTBANK('ROUGHER  ',NCELL1)
       CALL PPBANK(N,NCELL1,COMP,BCONC1)
       CALL PTBANK('SCAVENG ',NCELL2)
       CALL PPBANK(N,NCELL2,COMP,BCONC2)
       CALL PTBANK('1 CLEANER',NCELL3)
       CALL PPBANK(N,NCELL3,COMP,BCONC3)
       CALL PTBANK('2 CLEANER',NCELL4)
       CALL PPBANK(N,NCELL4,COMP,BCONC4)
       CALL PTBANK('3 CLEANER',NCELL5)
       CALL PPBANK(N,NCELL5,COMP,BCONC5)
       CALL PTBANK('4 CLEANER',NCELL6)
       CALL PPBANK(N,NCELL6,COMP,BCONC6)
       WRITE(2,170)(CONC6(I),I=1,N+1)
170    FORMAT(/'CONC.FLOW RATE FROM THE CIRCUIT'/,14X,F8.3,3X,9F8.3)
       WRITE(2,180)(TAIL(I),I=1,N+1)
180    FORMAT(/'TAILING FROM THE LAST BANK'/,14X,F8.3,3X,9F8.3)
       WRITE(2,190)(CL(I),I=1,N+1)
190    FORMAT(/'TOTAL FEED TO THE FIRST BANK'/,14X,F8.3,3X,9F8.3)
       CALL RCBANK(N,FEED,CONC6,SG)
       CALL GRAD(N,CONC6,ASSAY,SG,GRADE,PCSLD)
```

KMAIN4.FOR Mon 27 FEB 1989 11:15:58

```
111       CALL PTGRAD('FINAL   CONC.       ',GRADE,PCSLD)
112       CALL MTREC(N,FEED,CONC6,SG,ASSAY)
113       CALL GRAD(N,CONC1,ASSAY,SG,GRADER,PCSLDR)
114       CALL PTGRAD('ROUGHER CONC.       ',GRADER,PCSLDR)
115       CALL MTREC(N,CL,CONC1,SG,ASSAY)
116       END
117
118
```

ภาคผนวก จ.

โปรแกรมย่อยการจำลองแบบวงจรลอยแร่

ซึ่งเขียนด้วยภาษาฟอร์แทรน 77

```
1 ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
2 :       FLOTATION SIMULATION                          FSIM.FOR              :
3 :       3 MARCH 1986                                                        :
4 :       SUBROUTINE RDCOMP(N,COMP)                                           :
5 :       SUBROUTINE PTCOMP(N,COMP)                                           :
6 :       SUBROUTINE RDPULP(N,FEED,SG)                                        :
7 :       SUBROUTINE PTPULP(NAME,N,PULP)                                      :
8 :       SUBROUTINE RDCELL(N,PAR,RATEF,RATEN)                                :
9 :       SUBROUTINE PTCELL(N,PAR,RATEF,RATEN)                                :
10 :      SUBROUTINE CELL(N,PAR,RATEF,RATEN,FEED,CONC,TAIL)                   :
11 :      SUBROUTINE BANK(N,NCELL,PAR,RATEF,RATEN,FEED,RCONC,CONC,TAIL,       :
12 :      SOLIDS,REC,WREC)                                                    :
13 :      SUBROUTINE CONV(N,PULP,SAVE,IND)                                    :
14 :      SUBROUTINE PTBANK(NAME,NCELL)                                       :
15 :      SUBROUTINE PPBANK(N,NCELL,COMP,PULP)                                :
16 :      SUBROUTINE RCBANK(N,FEED,CONC,SG)                                   :
17 :      SUBROUTINE RDASSA(N,MTCOMP,ASSAY)                                   :
18 :      SUBROUTINE PTASSA(N,MTCOMP,ASSAY)                                   :
19 :      SUBROUTINE GRAD(N,PULP,ASSAY,SG,GRADE,PCSLD)                        :
20 :      SUBROUTINE PTGRAD(NAME,GRADE,PCSLD)                                 :
21 :      SUBROUTINE MTREC(N,FEED,CONC,SG,ASSAY)                              :
22 :      SUBROUTINE RDSIZE(N,NSIZE,FEED)                                     :
23 ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
24 ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
25        SUBROUTINE RDCOMP(N,COMP)
26 C      23 MARCH 1987
27 C      S.SAISINCHAI
28 C
29 C      READ COMPONENT TYPE NAME FOR FLOTATION
30 C
31 C      N=NO. OF COMPONENT EXCLUDING WATER
32 C
33        CHARACTER COMP(10)*8
34        READ(1,*)N
35        READ(1,*)(COMP(I),I=2,N+1)
36        COMP(1)='WATER'
37        RETURN
38        END
39 ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
40
41        SUBROUTINE PTCOMP(N,COMP)
42 C      23 MARCH 1987
43 C      S.SAISINCHAI
44 C
45 C      PRINT COMPONENT TYPE NAME FOR FLOTATION
46 C
47        CHARACTER COMP(10)*8
48 C
49        WRITE(2,20)(COMP(I),I=1,N+1)
50  20    FORMAT('         SOLIDS',2X,A8,3X,9A8)
51        RETURN
52        END
53 ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
54
55        SUBROUTINE RDPULP(N,FEED,SG)
```

```
      C       23 MARCH 1987
      C       S.SAISINCHAI
      C       .
      C       READ PULP DESCRIPTION FOR FLOTATION
      C       INPUT TPH. FOR WATER AND % BY WT. FOR THE COMPONENTS
      C       INPUT SG. FOR EACH COMPONENT
              REAL FEED(10),SG(10)
              N1=N+1
              READ(1,*)SOLIDS,(SG(I),I=2,N1),(FEED(I),I=1,N1)
      C       CONVERT TO FEED ARRAY,VOL. FLOW RATE(CU.M/HR)
              DO 10 I=2,N1
                   FEED(I)=(FEED(I)*(SOLIDS/SG(I)))/100.
   10         CONTINUE
      C       .
              RETURN
              END
C*****************************************************************************

              SUBROUTINE PTPULP(NAME,N,PULP)
      C       23 MARCH 1987
      C       S.SAISINCHAI
      C       .
      C       PRINT PULP DESCRIPTION FOR FLOTATION
      C       INPUT NAME OF THE STREAM IN CALLING SUBROUTINE
      C       E.G CALL PTPULP('ROUGHER',N,FEED)
      C       .
              CHARACTER NAME*8
              REAL PULP(10),TEMP(10)
              SOLIDS=0.
              DO 10 I=2,N+1
   10         SOLIDS=SOLIDS+PULP(I)
              DO 20 I=2,N+1
   20         TEMP(I)=100.*PULP(I)/SOLIDS
              WRITE(2,30)NAME,SOLIDS,PULP(1),(TEMP(I),I=2,N+1)
   30         FORMAT(/A8,F6.2,F8.2,3X,9F8.2)
      C       .
              RETURN
              END
C*****************************************************************************

              SUBROUTINE RDCELL(N,PAR,RATEF,RATEN)
      C       28 MARCH 1987
      C       S.SAISINCHAI
      C       .
      C       READ CELL DESCRIPTION
      C       PAR(1)=VOL. OF THE CELL
      C       PAR(2)=BETA CONSTANT
      C       RATEF=RATE CONSTANTS OF FLOATING COMPONENTS
      C       RATEN=RATE CONSTANT OF NON FLOATING COMPONENTS
      C       .
              REAL PAR(2),RATEF(10),RATEN(10)
              READ(1,*)PAR(1),PAR(2)
              READ(1,*)(RATEF(I),I=1,N+1),(RATEN(I),I=1,N+1)
              RETURN
              END
```

```
111 ********************************************************************************
112
113         SUBROUTINE PTCELL(N,PAR,RATEF,RATEN)
114 C       28 MARCH 1987
115 C       S.SAISINCHAI
116 C       .
117 C       PRINT CELL DESCRIPTION
118         REAL PAR(2),RATEF(10),RATEN(10)
119         WRITE(2,20)(RATEF(I),I=1,N+1),(RATEN(I),I=1,N+1),PAR(1),PAR(2)
120  20     FORMAT(/'FLOTATION RATE CONSTANTS',//'FLOATING      ',F8.2,3X,
121      1  3F8.2,/'NON FLOATING  ',F8.2,3X,3F8.2,//,'VOL.OF CELL.',5X,
122      1  'BETA',/F12.2,5X,F5.2)
123         RETURN
124         END
125 ********************************************************************************
126
127         SUBROUTINE CELL(N,PAR,RATEF,RATEN,FEED,CONC,TAIL)
128 C       21 MARCH 1987
129 C       S.SAISINCHAI
130 C       .
131 C       FLOTATION SIMULATION ALLOWING OVERLOADING CONDITION
132 C       TWO SETS OF RATE CONSTANTS FOR FLOATING AND NON FLOATING COMPO.
133 C       .
134 C       ITERATION CALCULATION PROCEDURE
135         REAL PAR(2),RATEF(10),RATEN(10),FEED(10),CONC(10),TAIL(10)
136         REAL CONCF(10),CONCN(10)
137 C       INTEGER CELLNO
138 C       PAR(1)=VOL.OF CELL
139 C       PAR(2)=BETA
140 C       CONCF=CONCENTRATE OF FLOATING COMPONENTS
141 C       COCCN=CONCENTRATE OF NON FLOATING COMPO.
142         H=1.
143         G=0.
144         WKSTAR=0.
145 C       K=1
146 C       .
147  25     SUMF=0.
148         DO 30 I=1,N+1
149              SUMF=SUMF+FEED(I)
150  30     CONTINUE
151         DO 100 M=1,50
152               ALPHA=(SUMF-G)/PAR(1)
153               SUMCF=0.
154               DO 40 I=1,N+1
155                    CONCF(I)=(FEED(I)*RATEF(I))/(1.*RATEF(I)+(H*ALPHA))
156                    SUMCF=SUMCF+CONCF(I)
157  40           CONTINUE
158               HNEW=1.+(H*PAR(2)*(SUMCF/PAR(1)))
159 C             .
160               SUMCN=0.
161               CONCN(1)=WKSTAR+RATEN(1)*SUMCF
162               DO 50 I=2,N+1
163                    CONCN(I)=(FEED(I)*(RATEN(I)*CONCN(1)))/
164      1             (RATEN(I)*CONCN(1)+ALPHA*PAR(1))
165                    SUMCN=SUMCN+CONCN(I)
```

```
 166  50          CONTINUE
 167              SUMC=SUMCF+SUMCN+CONCN(1)
 168              G=SUMC
 169 C            .
 170 C            .
 171 C            SUMC=SUM OF ALL COMPONENTS
 172              IF(ABS(H-HNEW)/HNEW .LT. 1.E-5)GOTO 200
 173              H=HNEW
 174  100    CONTINUE
 175 C       .
 176 C       ITERATION CALCULATION FOR ONE CELL
 177 C       CONVERT TAIL FROM THE FIRST CELL TO THE FEED TO THE NEXT CELL
 178 C       RELOCATE VALUES INTO CONC(I)
 179 C       .
 180  200    DO 250 I=1,N+1
 181              CONC(I)=CONCF(I)
 182              IF(CONC(I) .EQ. 0.)CONC(I)=CONCN(I)
 183  250    CONTINUE
 184         DO 400 I=1,N+1
 185  400         TAIL(I)=FEED(I)-CONC(I)
 186 C       K=K+1
 187         RETURN
 188         END
 189 ************************************************************************
 190
 191         SUBROUTINE BANK(N,NCELL,PAR,RATEF,RATEN,FEED,BCONC,CONC,TAIL,
 192        1 SOLIDS,REC,WREC)
 193 C       21 MARCH 1987,13 APRIL 1988
 194 C       S.SAISINCHAI
 195 C       .
 196 C       CALCULATE CONCENTRATE FLOW RATE OF N COMPONENTS
 197 C       IN A BANK OF NCELL CELLS
 198         REAL FEED(10),CONC(10),TAIL(10),TCONC(10),TEMP(100),TTAIL(10)
 199         REAL BCONC(10,100),PAR(2),RATEF(10),RATEN(10)
 200 C       INTEGER NCELL
 201 C       .
 202 C       COPY FEED(I) INTO TEMP(I) FOR I=1 TO N+1
 203 C       .
 204         DO 10 I=1,N+1
 205              TEMP(I)=FEED(I)
 206  10     CONTINUE
 207 C       .
 208         DO 25 I=1,N+1
 209              TTAIL(I)=0.
 210              TCONC(I)=0.
 211  25     CONTINUE
 212         TSOLID=0.
 213         K=1
 214  30     SOLIDS=0.
 215         CALL CELL(N,PAR,RATEF,RATEN,TEMP,CONC,TAIL)
 216 C       .
 217 C       CALCULATE SOLIDS VOL FLOW RATE IN EACH CELL
 218 C       .
 219         DO 40 I=2,N+1
 220              SOLIDS=SOLIDS+CONC(I)
```

```
 40     CONTINUE
C       .
C       CALCULATE SOLIDS VOL FLOW RATE OF EACH COMPONENT
C       .
        DO 45 I=1,N+1
              BCONC(I,K)=CONC(I)
              TCONC(I)=TCONC(I)+BCONC(I,K)
 45     CONTINUE
C       .
C       COPY SOLIDS FLOW RATE FOR EACH CELL INTO TEMP(I>N+1)
C       .
        TEMP(N+1+K)=SOLIDS
C       .
        IF(K .GE. NCELL)GOTO 60
        K=K+1
        DO 50 I=1,N+1
              TEMP(I)=TAIL(I)
 50     CONTINUE
        GOTO 30
 60     DO 70 I=1,N+1
              CONC(I)=TCONC(I)
              TAIL(I)=FEED(I)-CONC(I)
 70     CONTINUE
 100    DO 110 I=1,K
               TSOLID=TSOLID+TEMP(N+1+I)
 110    CONTINUE
        SOLIDS=TSOLIDS
C       .
        SUMCI=0.
        SUMFI=0.
        DO 120 I=2,N+1
               SUMCI=SUMCI+CONC(I)
               SUMFI=SUMFI+FEED(I)
 120    CONTINUE
        IF(SUMCI .GE. 0.00005)GOTO 121
        REC=0.
        GOTO 122
 121    REC=(SUMCI/SUMFI)*100.
 122    IF(FEED(1) .GE. 0.00005)GOTO 123
        WREC=0.
        GOTO124
 123    WREC=(CONC(1)/FEED(1))*100.
 124    RETURN
        END
 ***********************************************************************

        SUBROUTINE CONV(N,PULP,SAVE,IND)
C       21 MARCH 1987
C       S.SAISINCHAI
C       .
C       CALCULATE CONVERGENCE FOR CLOSED CIRCUIT CIRCULATING LOAD
C       GIVE FRACTIONAL DIFFERENCE BETWEEN NEW ESTIMATE AND CONCENTRATE
C       SPEED UP ITERATION BY FACTOR F
C       .
        REAL PULP(10),SAVE(10)
```

```
      DATA F/0.20/
      DO 10 I=1,N+1
           C=PULP(I)-SAVE(I)
           PULP(I)=PULP(I)+F*C
           SAVE(I)=PULP(I)
C          .
C          COMPARENEW ESTIMATE(PULP) AND PREVIOUS(SAVE)
C          IF DIFFERENCE IS GREATER,SET INDICATOR TO 1
C          .
           IF(ABS(C)/(PULP(I)+.0001) .GT. 1.E-5)IND=1
10    CONTINUE
C     .
C     RETURN VALUE OF INDICATOR TO THE MAIN PROGRAM
      RETURN
      END
**********************************************************************

      SUBROUTINE PTBANK(SNAME,NCELL)
C     21 MARCH 1987,13 APRIL 1988
C     S.SAISINCHAI
C     .
C     PRINT BANK DESCRIPTION NAME,NO. OF CELLS
C     .
      CHARACTER SNAME*10
      WRITE(2,20)SNAME,NCELL
20    FORMAT(//A10,8X,'BANK',/'NO.OF CELLS',5X,I4,3X,'CELLS')
      RETURN
      END
**********************************************************************

      SUBROUTINE PPBANK(N,NCELL,COMP,PULP)
C     21 MARCH 1987
C     S.SAISINCHAI
C     .
C     PRINT PULP FLOW RATE FROM EACH CELL IN THE BANK
C     OF NCELL CELLS AT STEADY STATE
      CHARACTER COMP(10)*8
      REAL PULP(10,100),TEMP(100),TCONC(10)
      WRITE(2,10)
10    FORMAT(//'CONC.FLOW RATE',/'CELL.NO')
      CALL PTCOMP(N,COMP)
      TSOLID=0.
      DO 20 I=1,N+1
20    TCONC(I)=0.
      DO 25 K=1,NCELL
25    TEMP(K)=0.
      DO 60 K=1,NCELL
           DO 30 I=2,N+1
                TEMP(K)=TEMP(K)+PULP(I,K)
30         CONTINUE
           TSOLID=TSOLID+TEMP(K)
C          .
C          PRINT CONCENTRATE FLOW RATE OF EACH CELL
C          .
           WRITE(2,40)K,TEMP(K),(PULP(I,K),I=1,N+1)
```

```
 40          FORMAT(I3,5X,F6.2,F8.3,3X,9F8.3)
C            .
C            CALCULATE TOTAL FLOW RATE OF EACH COMPO. FROM THE BANK
C            .
             DO 50 I=1,N+1
                  TCONC(I)=TCONC(I)+PULP(I,K)
 50          CONTINUE
 60     CONTINUE
C       .
C       PRINT TOTAL FLOW RATE FROM THE BANK
C       .
        WRITE(2,100)TSOLID,(TCONC(I),I=1,N+1)
 100    FORMAT(/'TOTAL FLOW RATE  (CU.M./HR.)'//,8X,F6.2,F8.3,3X,
     1  9F8.3)
        RETURN
        END
************************************************************************

        SUBROUTINE RCBANK(N,FEED,CONC,SG)
C       28 MARCH 1987,13 APRIL 1988
C       S.SAISINCHAI
C       .
C       CALCULATE RECOVERY OF SOLIDS AND WATER
C       FROM THE BANK OF NCELL CELLS
C       GANGUE MINERALS SPECIES IS THE LAST SPECIES(N+1)
C       .
        REAL FEED(10),CONC(10),SG(10)
        SUMF=0.
        DO 20 I=2,N+1
             SUMF=SUMF+FEED(I)*SG(I)
             SUMC=SUMC+CONC(I)*SG(I)
 20     CONTINUE
        TEMPR=(SUMC/SUMF)*100.
        TEMPWR=(CONC(1)/FEED(1))*100.
        WRITE(2,30)TEMPR,TEMPWR
 30     FORMAT(/'RECOVERY %',10X,F10.2,/'WATER RECOVERY %',4X,F10.2)
        RETURN
        END
************************************************************************

        SUBROUTINE RDASSA(N,MTCOMP,ASSAY)
C       23 MARCH 1987
C       S.SAISINCHAI
C       .
C       READ ASSAY VALUES OF EACH COMPONENT OR SPECIES(i.e % metal)
C       .
        CHARACTER MTCOMP*8
        REAL ASSAY(10)
        READ(1,*)MTCOMP,(ASSAY(I),I=2,N+1)
        RETURN
        END
************************************************************************

        SUBROUTINE PTASSA(N,MTCOMP,ASSAY)
C       23 MARCH 1987
```

```
386 C       S.SAISINCHAI
387 C       .
388 C       PRINT METAL COMPONENTAND PERCENT ASSAY IN EACH COMPONENT
389 C       .
390         CHARACTER MTCOMP*8
391         REAL ASSAY(10)
392         WRITE(2,20)MTCOMP,(ASSAY(I),I=2,N+1)
393  20     FORMAT(A8,17X,10F8.2)
394         RETURN
395         END
396 ************************************************************************
397
398         SUBROUTINE GRAD(N,PULP,ASSAY,SG,GRADE,PCSLD)
399 C       23 MARCH 1987
400 C       S.SAISINCHAI
401 C       .
402 C       CALCULATE GRADE AND PERCENT SOLIDS OF STREAM PULPS
403 C       .
404         REAL MASS(10),PULP(10),SG(10),TMETAL(10),ASSAY(10),METAL
405         TMASS=0.
406         METAL=0.
407         DO 10 I=2,N+1
408              MASS(I)=PULP(I)*SG(I)
409              TMASS=TMASS+MASS(I)
410              TMETAL(I)=MASS(I)*ASSAY(I)/100.
411              METAL=METAL+TMETAL(I)
412  10     CONTINUE
413 C       .
414 C       CALCULATE GRADE OF METAL %
415 C       .
416         GRADE=(METAL/TMASS)*100.
417         PCSLD=(TMASS/(TMASS+PULP(1)))*100.
418         RETURN
419         END
420 ************************************************************************
421
422         SUBROUTINE PTGRAD(NNAME,GRADE,PCSLD)
423 C       23 MARCH 1987
424 C       S.SAISINCHAI
425 C       .
426 C       PRINT GRADE AND PERCENT SOLIDS OF PULP STREAMS FROM FLOTATION
427 C       .
428         CHARACTER NNAME*20
429         WRITE(2,20)NNAME,GRADE,PCSLD
430  20     FORMAT(/A20,/'GRADE %METAL',13X,F6.3,4X,'%SOLIDS',5X,F6.2)
431         RETURN
432         END
433 ************************************************************************
434
435         SUBROUTINE MTREC(N,FEED,CONC,SG,ASSAY)
436 C       21 MARCH 1987
437 C       S.SAISINCHAI
438 C       .
439 C       CALCULATE RECOVERY OF METAL
440 C       .
```

```
      REAL FEED(10),CONC(10),FMASS(10),SG(10),ASSAY(10),CMASS(10)
      REAL TEMPF(10),TEMPC(10),MFEED,MCONC
C     .
C     CALCULATE METAL IN FEED SOLIDS
C     .
      MFEED=0.
      MCONC=0.
      DO 20 I=2,N+1
          FMASS(I)=FEED(I)*SG(I)
          TEMPF(I)=FMASS(I)*ASSAY(I)/100.
          MFEED=MFEED+TEMPF(I)
C         .
C         CALCULATE METAL IN CONCENTRATE
C         .
          CMASS(I)=CONC(I)*SG(I)
          TEMPC(I)=CMASS(I)*ASSAY(I)/100.
          MCONC=MCONC+TEMPC(I)
 20   CONTINUE
      RMET=(MCONC/MFEED)*100.
      WRITE(2,30)RMET
 30   FORMAT('RECOVERY (METAL)%',7X,F6.2)
      RETURN
      END
************************************************************************
C
      SUBROUTINE RDSIZE(N,NSIZE,FEED)
C     5 MARCH 1987
C     S.SAISINCHAI
C     .
C     READ PULP OF SEVERAL COMPONENTS AND DIFF. SIZE FRACTIONS
C     FOR FLOTATION THEN CALCULATE VOL. FLOW RATE FOR EACH COMPO.
C     AT EACH SIZE FRACTION
C     FEED ARRAY INPUT CONTAINS A MATRIX OF SIZE FRACTIONS AND COMPONENTS
C     AND FEED TPH
C     .
      REAL TEMPF(10,10),FCOMP(10),FEED(10,10),SG(10),VFCOMP(10)
      READ(1,*)NSIZE
      READ(1,*)SOLIDS,(SG(I),I=2,N+1)
      READ(1,*)((TEMPF(I,J),I=2,N+1),J=1,NSIZE)
C     .
      DO 30 I=2,N+1
      DO 20 J=1,NSIZE
      FCOMP(I)=FCOMP(I)+TEMPF(I,J)
 20   CONTINUE
 30   CONTINUE
C     .
C     CALCULATE TOTAL VOL. FLOW RATE OF EACH COMPO.
C     .
      DO 40 I=2,N+1
 40   VFCOMP(I)=((FCOMP(I)*SOLIDS)/SG(I))/100.
C     .
C     CALCULATE VOL. FLOW RATE OF EACH COMPONENT AT EACH SIZE FRACTION
C     .
      DO 60 J=1,NSIZE
      DO 50 I=2,N+1
```

```
      FEED(I,J)=VFCOMP(I)*TEMPF(I,J)/FCOMP(I)
50    CONTINUE
60    CONTINUE
      RETURN
      END
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

โปรแกรมหลักการจำลองแบบวงจรลอยแร่ บ. เหมืองแร่ตาดดาวจำกัด

ซึ่งเขียนด้วยภาษาเทอร์โบปาสคาล 4.0

MENUSIM.PAS

```
 1 Program
 2 uses
 3   bined,
 4   crt,                        {Basic video operations - standard unit}
 5   Dos,                        {DOS calls - standard unit}
 6   printer,turbo3,gdriver,gkernel,gshell,gwindow;
 7 {$I Editor.lib}
 8 {$I dos.lib}
 9 {$I tools.inc}
10 {$I simula.inc}
11 {$i flotpar.inc}
12 {$i file_ser.lib}
13
14
15 procedure compute;   {scompute.pas}
16 label 100;
17 begin
18           CLRSCR;
19           WRITESTR(35,12,'START COMPUTE !');
20
21           FOR I:=1 TO N+1 DO
22               begin
23                 SAVE[I]:=0;
24                 CONC2[I]:=0;
25                 tail3[i]:=0;
26                 tail4[i]:=0;
27                 tail5[i]:=0;
28               end;
29           FOR K:=1 TO 30 DO
30               BEGIN
31                    IND:=0;
32                    FOR I:=1 TO N+1 DO  CL[I]:=FEED[I]+CONC2[I]+TAIL3[I];
33                    BANK(N,NCELL[1],PAR,RATEFR,RATENR,CL,CONC1,TAIL1,BCONC1,SOLID1,REC1,WREC1);
34                    {tailing from the first bank becomes feed to the next(scavenger)}
35                    BANK(N,NCELL[2],PAR,RATEF,RATEN,TAIL1,CONC2,TAIL,BCONC2,SOLID2,REC2,WREC2);
36                    {concentrate from rougher bank and tailing from 2nd cleaner bank becomes feed to 1st cleaner
37                    FOR I:=1 TO N+1 DO  CLL[I]:=CONC1[I]+TAIL4[I];
38                    BANK(N,NCELL[3],PAR,RATEF,RATEN,CLL,CONC3,TAIL3,BCONC3,SOLID3,REC3,WREC3);
39                    {concentrate from 1st cleaner and tailing from 3rd cleaner
40                     bank becomes feed to 2nd cleaner}
41                    FOR I:=1 TO N+1 DO LCL[I]:=CONC3[I]+TAIL5[I];
42                    BANK(N,NCELL[4],PAR,RATEF,RATEN,LCL,CONC4,TAIL4,BCONC4,SOLID4,REC4,WREC4);
43                    {concentrate from 2nd cleaner bank becomes feed to 3rd cleaner}
44                    BANK(N,NCELL[5],PAR,RATEF,RATEN,CONC4,CONC5,TAIL5,BCONC5,SOLID5,REC5,WREC5);
45                    CONV(N,CONC2,SAVE,IND);
46                    IF(IND=0) THEN GOTO 100;
47               END;
48               WRITELN('     ### WARNING DID NOT CONVERGE ###      ');
49 100:          WRITELN('     ### END BANK CALULATION      ###      ');
50               CLRSCR;
51               WRITESTR(35,1,'FINISHED COMPUTE !');
52 end;
53 procedure title;
54 begin
55      initgraphic;
```

```
 56      defineWorld(1,0,0,1000,1000);
 57      defineWindow(1,0,0,89,349);
 58      defineWindow(2,30,95,57,155);
 59      selectWorld(1);
 60      selectWindow(2);
 61      drawborder;
 62      setcolorblack;
 63      setbackGround(85);
 64      drawtextw(145,460,6,'FSIM');
 65      selectWindow(1);
 66      setcolorwhite;
 67      drawtextw(150,140,3,'FLOTATION SIMULATION PROGRAM');
 68      drawtextw(473,600,2,'BY');
 69      drawtextw(300,700,2,'   SOMSAK  SAISINCHAI    ');
 70      drawtextw(300,750,2,'MINING  ENGINEERING  DEPT');
 71      drawtextw(300,800,2,'CHULALONGKORN  UNIVERSITY');
 72      drawtextw(300,850,2,'BANGKOK, THAILAND,  1988.');
 73      drawborder;
 74      read(kbd,re);
 75      leavegraphic;
 76 end;
 77
 78 procedure outsim;
 79 var ans :char;
 80 begin
 81      clrscr;
 82      writeln('!You can see output data after runing program!');
 83      write('Continue press [Y],  Return Menu press [N]');
 84      repeat read(kbd,ans);
 85            ans:=upcase(ans);
 86      until ans in ['Y','N'];
 87      case ans of
 88          'Y':begin outs_direct;end;
 89          'N':begin clrscr; end;
 90      end;
 91 end;
 92
 93 procedure outsimp;
 94 var ans :char;
 95 begin
 96      clrscr;
 97      writeln('!You can print output data after runing program!');
 98      writeln('!The printer must be ready before printing!');
 99      write('Continue press [Y],  Return Menu press [N]');
100      repeat read(kbd,ans);
101            ans:=upcase(ans);
102      until ans in ['Y','N'];
103      case ans of
104          'Y':begin out_filep;end;
105          'N':begin clrscr; end;
106      end;
107 end;
108
109 procedure outkp;
110 var ans :char;
```

```
111 begin
112      clrscr;
113      writeln('!You can print output data after runing K program!');
114      writeln('!The printer must be ready before printing!');
115      write('Continue press [Y],  Return Menu press [N]');
116      repeat read(kbd,ans);
117           ans:=upcase(ans);
118      until ans in ['Y','N'];
119      case ans of
120          'Y':begin kout_datafilep;end;
121          'N':begin clrscr; end;
122      end;
123 end;
124 procedure main_menu;
125 begin
126      repeat
127      cursorOn;
128      clrscr;
129      border1(1,2,80,22);gotoxy(34,2);write('           ');
130      gotoxy(7,2); write('                      ');
131      gotoxy(53,2);write('                       ');
132      border_solid(34,1,46,3);border2(6,1,28,3);border2(52,1,74,3);
133      gotoxy(8,2);write('DATE : ');get_date;
134      gotoxy(56,2);write('TIME : ');
135      gotoxy(36,2);write('MAIN MENU');
136      gotoxy(1,12);write(chr(195));gotoxy(80,12);write(chr(180));
137      for i:=2 to 79 do begin gotoxy(i,12);write(chr(196));end;
138      for i:=4 to 23 do begin gotoxy(40,i);write(chr(179));end;
139      gotoxy(40,12);write(chr(197));
140      textcolor(black);textbackground(white);
141      gotoxy(10,5);write(' SIMULATION ');
142      gotoxy(55,5);write(' EDIT/CREATE ');
143      gotoxy(8,14);write(' K RATE CONSTANT ');gotoxy(54,14);write(' MISCELLANEOUS ');
144      textcolor(white);
145      gotoxy(7,7);  write('A.  Run Program by file');
146      gotoxy(7,8);  write('B.  Run Program by Interective');
147      gotoxy(7,9);  write('C.  Display Output');
148      gotoxy(7,10); write('D.  Print Output');
149      gotoxy(47,7); write('E.  Create/ Edit K Data File');
150      gotoxy(47,8); write('F.  Create/ Edit Other Data File');
151      gotoxy(47,9); write('G.  Create/ Edit Input Data File');
152      gotoxy(7,17); write('H.  Compute');
153      gotoxy(7,18); write('I.  Display Input File');
154      gotoxy(7,19); write('J.  Display Output File');
155      gotoxy(7,20); write('K.  Print Output');
156      gotoxy(47,17);write('L.  Information');
157      gotoxy(47,18);write('S.  File Service');
158      gotoxy(47,19);write('X.  Exit to Operating System');
159      border1(1,22,80,24);
160      gotoxy(1,22);write(chr(195));gotoxy(80,22);write(chr(180));
161      gotoxy(40,22);write(chr(193));
162      gotoxy(34,23);write('Please Select... ');
163      textcolor(20);write('_');textcolor(2);
164      repeat
165      repeat gotoxy(63,2);cursorOff;get_time;until keypressed;
```

```
166      cursorOn;
167      re:=readkey;re:=upcase(re);
168      until re in ['A','B','C','D','E','F','G','H','I','J','K','L','S','X'];
169      case re of
170               'A':begin in_file;compute;out_file;end;
171               'B':begin clrscr;in_direct;compute;out_file;end;
172               'C':begin outsim;end;
173               'D':begin outsimp;end;
174               'E':begin f_name:='ink.dat';edit;end;
175               'F':begin change_file;f_name:=data_file;edit;end;
176               'G':begin f_name:='input.dat';edit;end;
177               'H':begin flotpar;end;
178               'I':begin kget_data;kdisplay;end;
179               'J':begin kout_data; end;
180               'K':begin outkp;end;
181               'L':begin writeln('file_name;print_file');end;
182               'S':begin change_file;end;
183               'X':begin clrscr;halt;end;
184     end;
185 cursorOn;
186 until re = 'XXX';
187 end;
188
189 begin
190      clrscr;
191      checkBreak:=false;
192      data_file:='NONAME.DAT';title_file:='NONAME.TLE';output_file:='NONAME.OUT';
193      set_screen;lowvideo;
194      getDir(0,strDir);drvName := strDir[1];
195      strDir:=copy(strDir,3,30);
196      title;
197      clrscr;
198      main_menu;
199 end.
```

ภาคผนวก ช.

โปรแกรมย่อยการจำลองแบบวงจรลอยแร่

ซึ่งเขียนด้วยภาษาเทอร์โบปาสคาล 4.0

EDITOR.LIB

```
 1 {EDITOR.LIB}
 2
 3 const
 4   {Commands other than ^K^D to exit editor}
 5   ExitCommands : array[0..18] of Char =
 6   (#2,^K,^Q,#2,#0,#68,#2,^K,^S,#2,#0,#61,#2,#0,#60,#2,#0,#62,#0);
 7   MakeBackup = False;
 8
 9 const
10   {Initial Coordinates of the editor window}
11   Windx1 = 2;
12   Windy1 = 2;
13   Windx2 = 79;
14   Windy2 = 22;
15
16 var
17   EdData : EdCB;              {Editor control block}
18   ExitCode : Word;            {Status code set by binary editor functions}
19   ExitCommand : Word;         {Code for command used to leave editor}
20   Fname,F_name : String;            {Input name of file being edited}
21
22 type
23   BorderElements = (topleft, topright, botleft, botright, horiz, vert);
24   BorderChars = array[BorderElements] of Char;
25 const
26   Border :   BorderChars = '+++++-|';
27   NoBorder : BorderChars = '      ';
28
29 { demo.inc}
30
31   procedure DrawBox(Border : BorderChars; x1, y1, x2, y2 : byte);
32     {-Draw a box around an editor window}
33   var
34     i : Word;
35     bar : String;
36     barlen : Byte absolute bar;
37
38   begin                         {DrawBox}
39
40     {Build horizontal bar}
41     barlen := 3+X2-X1;
42     FillChar(bar[1], barlen, Border[horiz]);
43
44     {Draw top bar}
45     bar[1] := Border[topleft];
46     bar[barlen] := Border[topright];
47     CRTputFast(X1, Y1, bar);
48
49     {Draw botton bar}
50     bar[1] := Border[botleft];
51     bar[barlen] := Border[botright];
52     CRTputFast(X1, Y2+2, bar);
53
54     {Vertical bars}
55     for i := Succ(Y1) to Succ(Y2) do begin
```

```
56      CRTputFast(X1, i, Border[vert]);
57      CRTputFast(X2+2, i, Border[vert]);
58    end;
59  end;                        {DrawBox}
60
61  procedure WriteStatus(msg : String);
62    {-Write a status message to the bottom line of the screen}
63  var
64    msglen : Byte absolute msg;
65
66  begin                       {WriteStatus}
67    FillChar(msg[Succ(msglen)], 80-msglen, #32);
68    msglen := 80;
69    CRTputFast(1, 25, CAerr+msg);
70  end;                        {WriteStatus}
71
72  procedure CheckInitBinary(ExitCode : Word);
73    {-Check the results of the editor load operation}
74
75  begin                       {CheckInitBinary}
76    if ExitCode <> 0 then begin
77      {Couldn't initialize editor}
78      GoToXY(1, 25);
79      case ExitCode of
80        1 : WriteLn('Insufficient heap space for text buffer');
81      else
82        WriteLn('Unknown load error');
83      end;
84      delay(3000);
85      Halt(1);
86    end;
87  end;                        {CheckInitBinary}
88
89  procedure CheckReadFile(ExitCode : Word; Fname : String);
90    {-Check the results of the file read}
91  var
92    f : file;
93
94  begin                       {CheckReadFile}
95    if ExitCode <> 0 then begin
96      {Couldn't read file}
97      case ExitCode of
98        1 : begin
99              {New file, assure valid file name}
100             {$I-}
101             Assign(f, Fname);
102             Rewrite(f);
103             if IOResult <> 0 then begin
104               Close(f);
105               WriteStatus('Illegal file name '+Fname);
106             end else begin
107               Close(f);
108               Erase(f);
109               Write('New File');
110               Delay(1000);
```

```
111                Write(^M);
112                ClrEol;
113                GoToXY(1, 1);
114                ClrEol;
115                Exit;
116              end;
117              {$I+}
118            end;
119          2 : WriteStatus('Insufficient text buffer size');
120        else
121          WriteStatus('Unknown read error');
122        end;
123        GoToXY(1, 25);
124        delay(3000);
125        Halt(1);
126      end;
127      GoToXY(1, 1);
128      ClrEol;
129    end;                          {CheckReadFile}
130
131    procedure CheckSaveFile(ExitCode : Word; Fname : String);
132      {-Check the results of a file save}
133
134    begin                         {CheckSaveFile}
135      if ExitCode <> 0 then begin
136        {Couldn't save file}
137        case ExitCode of
138          1 : WriteStatus('Unable to create output file '+Fname);
139          2 : WriteStatus('Error while writing output to '+Fname);
140          3 : WriteStatus('Unable to close output file '+Fname);
141        else
142          WriteStatus('Unknown write error');
143        end;
144        GoToXY(1, 25);
145        delay(3000);
146        Halt(1);
147      end;
148    end;                          {CheckSaveFile}
149
150    procedure WriteKeyboardToggles(info : Word);
151      {-Write the status of the keyboard toggles}
152    var
153      s : String;
154
155    begin                         {WriteKeyboardToggles}
156      s := CAerr;
157      if (info and $40) <> 0 then
158        s := s+'CL'
159      else
160        s := s+'  ';
161      if (info and $20) <> 0 then
162        s := s+' NL'
163      else
164        s := s+'   ';
165      if (info and $10) <> 0 then
```

```
166       s := s+' SL'
167     else
168       s := s+'   ';
169     CRTputFast(72, 25, s);
170   end;                          {WriteKeyboardToggles}
171
172 type
173   string20 = string[20];
174 var
175   TickCount : Word;         {Counter used to support on-screen clock}
176   TickMax : Word;           {Count when on-screen clock is updated}
177   LastTime : String20;      {Current time showing on screen}
178
179 {Note the user event handler must have a FAR attribute}
180 {$F+}
181
182   procedure UserEventCheck(EventNo, KbdFlagInfo : Word);
183     {-User hook for a background process called at every keypressed check}
184   var
185     NewTime : String20;
186
187     function Time : String20;
188       {-Return a string holding the current time}
189     type
190       string2 = string[2];
191     var
192       hours, mins : string2;
193       hiclock, loclock : Word;
194       regs : registers;
195
196       function ZeroPad(s : string2) : string2;
197         {-Left pad a numeral with a zero}
198
199       begin                     {ZeroPad}
200         if s[0] = #1 then
201           s := '0'+s;
202         ZeroPad := s;
203       end;                      {ZeroPad}
204
205     begin                       {Time}
206
207       {Get the time from DOS}
208       regs.ah := $2C;
209       intr($21, regs);
210       hiclock := regs.cx;
211       loclock := regs.dx;
212
213       {Convert to string}
214       Str(Hi(hiclock), hours);
215       Str(Lo(hiclock), mins);
216       Time := ' '+ZeroPad(hours)+':'+ZeroPad(mins)+' ';
217     end;                        {Time}
218
219   begin                         {UserEventCheck}
220
```

```
221     {Update on-screen clock once a minute}
222     if TickCount > TickMax then begin
223       TickCount := 0;
224       NewTime := Time;
225       if NewTime <> LastTime then begin
226         CRTputFast(65, 25, CAerr+NewTime);
227         LastTime := NewTime;
228       end;
229     end else
230       TickCount := Succ(TickCount);
231
232     if eventno = EventKBflag then
233       {Update keyboard toggles whenever changed}
234       WriteKeyboardToggles(kbdflaginfo);
235
236   end;                          {UserEventCheck}
237
238 {$R-}
239
240   { End demo.inc }
241   function GetFileName : String;
242     {-Return a file name either from the command line or a prompt}
243   var
244     Fname : String;
245
246   begin                         {GetFileName}
247     Fname:='noname.dat';
248     GetFileName := Fname;
249   end;                          {GetFileName}
250
251   procedure InitWindow(var EdData : EdCB);
252     {-Draw a nice screen frame around the editor window}
253   var
254     MsgPos : Byte;
255     DemoMsg : String;
256
257   begin                         {InitWindow}
258     {Draw a frame around the editor window}
259     with EdData do begin
260       DrawBox(Border, x1, y1, x2, y2);
261     end;
262   end;                          {InitWindow}
263
264   procedure InitStatusLine;
265     {-Draw a status/prompt line for the editor demo}
266
267   begin                         {InitStatusLine}
268     WriteStatus(' F2-Save  F3-Load  F4-Exit  F10-Resizes Window');
269     {Initialize for the on-screen clock}
270     TickMax := 2500; {Empirical: keep clock ticking without excessive overhead}
271     TickCount := TickMax;
272     LastTime := '';
273   end;                          {InitStatusLine}
274
275   function ExitBinaryEditor(var EdData : EdCB; ExitCommand : Integer}
```

```
276     : Boolean;
277     {-Handle an editor exit - save or abandon file}
278   var
279     ExitCode : Word;
280
281     procedure ModifyWindow(var EdData : EdCB);
282       {-Move or resize editor window interactively}
283     var
284       ch : Char;
285       redraw, done, scroll : Boolean;
286       kbflag : Word absolute $0040 : $0017;
287       lastkbflag : Word;
288
289       procedure UpdateScreen(var EdData : EdCb);
290         {-Update the screen after the window is resized}
291       var
292         junk : Word;
293
294       begin                    {UpdateScreen}
295         {Redraw the window box}
296         InitWindow(EdData);
297         if not(keypressed) then begin
298           {Force the editor to update the screen and return}
299           junk := UseBinaryEditor(EdData, ^K^S);
300           redraw := false;
301         end;
302       end;                     {UpdateScreen}
303
304     begin                      {ModifyWindow}
305
306       with eddata do begin
307
308         {Show a prompt for resizing}
309         WriteStatus(^X^Y^Z^[' to move/resize window. <ScrollLock> moves. <Enter> accepts.');
310
311         lastkbflag := not(kbflag);
312         done := False;
313         redraw := False;
314
315         repeat
316
317           {Update the screen}
318           if redraw then
319             UpdateScreen(EdData);
320
321           {Keep keyboard toggles up to date}
322           repeat
323             scroll := (kbflag and $10) <> 0;
324             if kbflag <> lastkbflag then begin
325               WriteKeyboardToggles(kbflag);
326               lastkbflag := kbflag;
327             end;
328           until KeyPressed;
329
330           ch := Readkey;
```

```
331
332         if (ch = #0) and KeyPressed then begin
333
334           ch:=Readkey;
335
336           case ch of
337
338             #75 :           {left arrow}
339               if scroll then begin
340                 if X1 > 1 then begin
341                   DrawBox(NoBorder, x1, y1, x2, y2);
342                   X1 := Pred(X1);
343                   X2 := Pred(X2);
344                   redraw := true;
345                 end;
346               end else if X2 > X1+30 then begin
347                 DrawBox(NoBorder, x1, y1, x2, y2);
348                 X2 := Pred(X2);
349                 redraw := true;
350               end;
351
352             #77 :           {right arrow}
353               if scroll then begin
354                 if X2 < 78 then begin
355                   DrawBox(NoBorder, x1, y1, x2, y2);
356                   X1 := Succ(X1);
357                   X2 := Succ(X2);
358                   redraw := true;
359                 end;
360               end else if X2 < 78 then begin
361                 DrawBox(NoBorder, x1, y1, x2, y2);
362                 X2 := Succ(X2);
363                 redraw := true;
364               end;
365
366             #72 :           {up arrow}
367               if scroll then begin
368                 if Y1 > 1 then begin
369                   DrawBox(NoBorder, x1, y1, x2, y2);
370                   Y1 := Pred(Y1);
371                   Y2 := Pred(Y2);
372                   redraw := true;
373                 end;
374               end else if Y2 > Y1+3 then begin
375                 DrawBox(NoBorder, x1, y1, x2, y2);
376                 Y2 := Pred(Y2);
377                 redraw := true;
378               end;
379
380             #80 :           {down arrow}
381               if scroll then begin
382                 if Y2 < 22 then begin
383                   DrawBox(NoBorder, x1, y1, x2, y2);
384                   Y1 := Succ(Y1);
385                   Y2 := Succ(Y2);
```

```
386                   redraw := true;
387                 end;
388               end else if Y2 < 22 then begin
389                 DrawBox(NoBorder, x1, y1, x2, y2);
390                 Y2 := Succ(Y2);
391                 redraw := true;
392               end;
393
394           end;
395         end else if ch = #13 then
396           done := True;
397
398       until done;
399
400       InitStatusLine;
401
402     end;
403   end;                        {ModifyWindow}
404
405   function YesAnswer(prompt : String) : Boolean;
406     {-Return true for a yes answer to the prompt}
407   var
408     ch : Char;
409
410   begin                       {YesAnswer}
411     WriteStatus(prompt);
412     repeat
413       ch:=UpCase(Readkey);
414     until ch in ['Y', 'N'];
415     YesAnswer := (ch = 'Y');
416   end;                        {YesAnswer}
417
418 begin                         {ExitBinaryEditor}
419   case ExitCommand of
420     -1,4 :                     {^K^D}  {F2}
421       begin
422         ExitCode := SaveFileBinaryEditor(EdData, MakeBackup);
423         CheckSaveFile(ExitCode, FilenameBinaryEditor(eddata));
424         ExitBinaryEditor := True;
425         GoToXY(1, 25);
426         releaseBinaryEditorHeap(Eddata);
427       end;
428
429     0,5 :                      {^K^Q}  {F4}
430       begin
431         if ModifiedFileBinaryEditor(EdData) then
432           if YesAnswer('File modified. Save it? (Y/N) ') then begin
433             ExitCode := SaveFileBinaryEditor(EdData, MakeBackup);
434             CheckSaveFile(ExitCode, FilenameBinaryEditor(eddata));
435           end;
436         ExitBinaryEditor := True;
437         GoToXY(1, 25);
438         releaseBinaryEditorHeap(Eddata);
439       end;
440
```

```
441       1, 2 :                  {F10 or ^K^S}
442         begin
443           {Allow moving or resizing the window}
444           ModifyWindow(EdData);
445           ExitBinaryEditor := False;
446         end;
447       3    :              {F3}
448             begin
449             gotoxy(1,24);clrEol;
450             write('Enter filename : ');
451             readln(Fname);
452             gotoxy(1,24);clrEol;
453             ExitCode := ReadFileBinaryEditor(EdData, Fname);
454             CheckReadFile(ExitCode, FilenameBinaryEditor(eddata));
455
456            {Reset the editor for the new file}
457             ResetBinaryEditor(EdData);
458
459             {Write a status and prompt line}
460             InitStatusLine;
461
462             ExitBinaryEditor:=False;
463             end;
464     end;
465   end;                        {ExitBinaryEditor}
466
467 procedure edit;
468
469 begin
470
471   {Get a file name}
472   Fname := F_name;
473
474   ClrScr;
475
476   {Initialize a window for the file}
477   ExitCode :=
478   InitBinaryEditor(
479   EdData,                     {Editor control block, initialized by InitBinaryEditor}
480   MaxFileSize,                {Size of data area to reserve for binary editor text buffer, $FFE0 max}
481   Windx1,                     {Coordinates of editor window, upper left 1..80}
482   Windy1,                     {Coordinates of editor window, upper left 1..25}
483   Windx2,                     {Coordinates of editor window, lower right}
484   Windy2,                     {Coordinates of editor window, lower right}
485   True,                       {True to wait for retrace on color cards}
486   EdOptInsert+EdOptIndent,    {Initial editor toggles}
487   '',                         {Default file extension}
488   ExitCommands,               {Commands to exit editor}
489   Addr(UserEventCheck));      {Address of user event handler}
490
491   CheckInitBinary(ExitCode);
492
493   {Read the file}
494   ExitCode := ReadFileBinaryEditor(EdData, Fname);
495   CheckReadFile(ExitCode, FilenameBinaryEditor(eddata));
```

```
496
497   {Reset the editor for the new file}
498   ResetBinaryEditor(EdData);
499
500   {Write a status and prompt line}
501   InitStatusLine;
502
503   repeat
504     {Set up the window border and title}
505     InitWindow(EdData);
506
507     {Edit the file}
508     ExitCommand := UseBinaryEditor(EdData, '');
509
510     {Handle the exit by saving the file or whatever}
511   until ExitBinaryEditor(EdData, ExitCommand);
512
513   ClrScr;
514
515 end;
516
```

```
 1
 2 {DOS.LIB}
 3 Var
 4         Regs        : Registers;
 5         DrvName     : Char;
 6         StrDir      : String[30];
 7         check       : Boolean;
 8
 9 Procedure Get_dir;
10
11 begin
12         Getdir(0,strDir);
13         strDir:=copy(strDir,3,30)+'\';
14 end;
15
16 procedure set_screen;
17 const
18       VideoBW80x25A           =   2; {Mode 80x25 B/W, Alpha}
19       VideoColor80x25A        =   3; {Mode color 80x25,Alpha}
20
21       begin
22           Regs.AX := VideoBW80x25A;
23           Regs.BX := 15;Intr($10,Regs);
24 end;
25
26 procedure cursor_off;   { Set cursor off temporary }
27 begin
28       Regs.AX := $200;          {Set cursor position}
29       Regs.DX := $1A01;
30       Intr($10,Regs);           {Set cursor to Row 26 and Column 1}
31 end;
32
33 procedure cursorOff;
34 begin
35    regs.AH := $01;
36    regs.CH :=  14;
37    regs.CL :=  14;
38    intr($10,regs);
39 end;
40
41 procedure cursorBig;
42 begin
43    regs.AH := $01;
44    regs.CH :=   0;
45    regs.CL :=  13;
46    intr($10,regs);
47 end;
48
49 procedure cursorOn;
50 begin
51    regs.AH := $01;
52    regs.CH :=  12;
53    regs.CL :=  13;
54    intr($10,regs);
55 end;
```

```
 56
 57 procedure DosVer;
 58 function DosVer:Real;
 59 var AL,AH:byte;
 60 begin
 61      Regs.AX:=$3000;
 62      MSDos(Regs);
 63      AL:=Regs.AX and $FF;
 64      AH:=(Regs.AX and $FF00) shr 8;
 65      DosVer:= AL+AH/100;
 66 end;{Function DosVer}
 67 begin {Main body of DosVer}
 68       write(' MS DOS ver. ',DosVer:4:2,' ');
 69 end;
 70
 71 procedure get_date;
 72 var year,month,day,dayfweek:word;
 73     dayofweek,Sun,Mon,Tue,Wed,Thu,Fri,Sat:string[3];
 74     zero,zero1:string[1];
 75 begin
 76      getdate(year,month,day,dayfweek);
 77      case dayfweek of
 78      0:dayofweek:='Sun';
 79      1:dayofweek:='Mon';
 80      2:dayofweek:='Tue';
 81      3:dayofweek:='Wed';
 82      4:dayofweek:='Thu';
 83      5:dayofweek:='Fri';
 84      6:dayofweek:='Sat';
 85      end;
 86      if month < 10 then zero:='0' else zero:='';
 87      if day < 10 then zero1:='0'  else zero1:='';
 88      year := year-1900;
 89      write(dayofweek,' ',zero1,day:1,'-',zero,month:1,'-',year);
 90 end;
 91
 92 procedure get_time;
 93 var hour,minute,second,sec100:word;
 94 begin
 95     getTime(hour,minute,second,sec100);
 96     write(hour:2,':',minute:2,':',second:2);
 97 end;
 98
 99 procedure border1(x1,y1,x2,y2 : integer);
100 var i,j:integer;
101       begin
102           if y2 > 24 then y2:=24;
103           if x2 > 80 then x2:=80;
104           if x1 = 0 then x1:=1;
105           if y1 = 0 then y1:=1;
106           gotoxy(x1,y1);write(chr(218));
107           for i:=1 to x2-x1-1 do write(chr(196));write(chr(191));
108           for j:=1 to y2-y1-1 do
109                begin
110                     gotoxy(x1,y1+j);write(chr(179));
```

```
111                     gotoxy(x2,y1+j);write(chr(179))
112                 end;
113             gotoxy(x1,y2);write(chr(192));
114             for i:=1 to x2-x1-1 do write(chr(196));write(chr(217));
115 end;
116
117 procedure border2(x1,y1,x2,y2 : integer);
118 var i,j:integer;
119       begin
120             if y2 > 24 then y2:=24;
121             if x2 > 80 then x2:=80;
122             if x1 = 0 then x1:=1;
123             if y1 = 0 then y1:=1;
124             gotoxy(x1,y1);write(chr(201));
125             for i:=1 to x2-x1-1 do write(chr(205));write(chr(187));
126             for j:=1 to y2-y1-1 do
127                 begin
128                     gotoxy(x1,y1+j);write(chr(186));
129                     gotoxy(x2,y1+j);write(chr(186))
130                 end;
131             gotoxy(x1,y2);write(chr(200));
132             for i:=1 to x2-x1-1 do write(chr(205));write(chr(188));
133 end;
134
135 procedure border_solid(x1,y1,x2,y2:integer);
136 begin
137      textcolor(0);textbackground(15);
138      border2(x1,y1,x2,y2);
139      textcolor(2);
140 end;
141
142 procedure reverse;
143    begin
144         textcolor(0);textbackground(15);
145 end;
146 procedure normal;
147    begin
148         textcolor(2);textbackground(0);
149 end;
150
151
152
153
```

```
 1
 2 type
 3     str80=string[80];
 4 const
 5     lovideo=7;
 6     hivideo=15;
 7     blink=16;
 8     underln=-6;
 9     inverse=8;
10     norm=$0b0c;
11     none=$1000;
12     full=$010c;
13     half=$060c;
14
15 procedure play(f,d:integer);
16 begin
17   sound(f);
18   delay(d);
19   Nosound;
20 end;
21
22 procedure screenattr(i:byte);
23 begin
24   textcolor(black);
25   textbackground(white);
26   textcolor(i);
27 end;
28
29 function upcasestr(instr:str80):str80;
30 var
31   outstr : str80;
32   i      : integer;
33 begin
34   read(instr);
35   for i := 0 tO length (instr) do
36   outstr[i]:=upcase(instr[i]);
37   upcasestr := outstr;
38 end;
39
40 procedure writestr1(s:str80;c,r,l,p,attr:integer);
41 begin
42   Screenattr(attr);
43   gotoxy(c,r);
44   if length(s) > L then
45     write(s)
46   else
47     case p of
48       1: write(s,' ':L-Length(s));
49       2: write(s:(L-Length(s)) div 2 + length(s));
50       3: write(s:L);
51     end;
52   normvideo;
53 end;
54
55
```

```
 56 procedure writestr(col,row:integer;S:STR80);
 57 begin
 58  gotoxy(col,row);
 59  write(S);
 60 end;
 61 procedure writecentr(ln,row:integer;s:str80);
 62 begin
 63   writestr((ln div 2)-length(s) div 2 ,row,s);
 64 end;
 65
 66 function yes(ln,row:integer;s:str80):boolean;
 67 var
 68   ch:char;
 69 begin
 70   writecentr(ln,row,s);
 71   CH:=readkey;
 72 {  read(KBD,CH);}
 73 { write(ch);}
 74   Yes:= UpCase(ch)='Y';
 75   gotoxy(1,row);
 76   clreol;
 77 end;
 78
 79 function  getdirection(var x,y: integer): boolean;
 80 var
 81   c:char;
 82 begin
 83   repeat
 84     c:=readkey;
 85    { read(kbd,c);}
 86     if upcase(c) in ['N','E','S','W'] then write(' ');
 87       case upcase(c) of
 88         'N': begin  Y:=Y-1;  if y<1 then y:=1;  end;
 89         'E': begin  X:=X+1;  if x>80 then x:=80;  end;
 90         'S': begin  Y:=Y+1;  if y>25 then y:=25;  end;
 91         'W': begin  X:=X-1;  if x<1 then x:=1;  end;
 92         'Q': begin end;
 93       else
 94         write(^G);
 95       end;
 96   until upcase(c) in ['E','N','S','W','Q'];
 97   getdirection:=upcase(c) <>'Q';
 98 end;
 99
100
101 procedure drawhline(ch:char;lowcol,upcol:integer);
102 var
103   s:string[80];
104 begin
105   fillchar(s,lowcol-upcol,ch);
106   s[0]:=chr(lowcol-upcol-1);
107   write(s);
108 end;
109
110 procedure drawconner(ch:char;c,r:integer);
```

```
111 begin
112   gotoxy(c,r);
113   write(ch);
114 end;
115
116 procedure drawvline(ch:char;lowcol,lowrow,upcol,uprow:integer);
117 var i:integer;
118 begin
119   for i:=uprow+1 to lowrow do
120    begin
121      gotoxy(upcol,i);
122      write(ch);
123      gotoxy(lowcol,i);
124      write(ch);
125    end;
126 end;
127
128 procedure drawbox1(upcol,uprow,lowcol,lowrow:integer);
129 begin {drawbox}
130   drawconner(#201,upcol,uprow);
131   drawhline(#205,lowcol,upcol);
132   drawconner(#187,lowcol,uprow);
133   drawvline(#186,lowcol,lowrow,upcol,uprow);
134   drawconner(#200,upcol,lowrow);
135   drawhline(#205,lowcol,upcol);
136   drawconner(#188,lowcol,lowrow);
137   window(upcol+1,uprow+1,lowcol-1,lowrow-1);
138 end;
139
140 {PROGRAM YESORNO;}{HAVE OR NO UNNESSESARY}
141 FUNCTION YESORNO:BOOLEAN;
142 VAR
143   CH:CHAR;
144 BEGIN
145   GOTOXY(20,24);
146   WRITE('TO CORRECT THIS DATA AGAIN ? <YES/NO>');
147   ch:=readkey;
148  { READ(KBD,CH);}                    {AFTER KBD PUT"CH"}
149   IF UPCASE(CH)='Y' THEN     {1    CHANGE n OR N TO "N" AFTER UPCASE MUSTBE}
150      YESORNO:=FALSE          {2      "CH" NOT "STRING"}
151   ELSE                       {3      {YESORNO:=UPCASE(CH)='Y';}{=1,2,3,4}
152      YESORNO:=TRUE;          {4                                          }
153 END;
154 procedure thaibox(x1,y1,x2,y2:integer);
155 var
156   n,i:integer;
157 begin
158    for n:=x1 to x2 do
159      begin
160        gotoxy(n,y1);write(#221);
161        gotoxy(n,y2);write(#221);
162      end;
163    for n:=y1 to y2 do
164      begin
165        gotoxy(x1,n);write(#221);
```

```
166        gotoxy(x2,n);write(#221);
167      end;
168 end;
169 PROCEDURE  DrawBox11  (x1,y1,x2,y2 : Integer);
170 VAR  n : Integer;
171 BEGIN  { drawbox }
172 { ONE } FOR n := x1 TO x2 DO
173           BEGIN
174             GotoXY (n,y1); Write (CHR(205));
175             GotoXY (n,y2); Write (CHR(205));
176           END;
177 { TWO } FOR n := y1 TO y2 DO
178           BEGIN
179             GotoXY (x1,n); Write (CHR(186));
180             GotoXY (x2,n); Write (CHR(186));
181           END;
182           GotoXY (x1,y1); Write (CHR(201));
183           GotoXY (x1,y2); Write (CHR(200)) ;
184           GotoXY (x2,y1); Write (CHR(187));
185           GotoXY (x2,y2); Write (CHR(188));
186 END;  { drawbox }
187 procedure thaihline(x,y1,y2:integer);
188 var
189   n,i:integer;
190 begin
191    for n:=y1 to y2 do
192     begin
193        gotoxy(x,n);write(#221);
194     end;
195 end;
196 {
197 procedure  direction(var x,y:integer);
198 var
199   c:char;
200 begin
201   repeat
202     read(kbd,c);
203     if upcase(c) in ['2','4','6','8'] then write(' ');
204     if upcase(c)='' then
205       case upcase(c) of
206         '8': begin  Y:=Y-1;  if y<1 then y:=1;  end;
207         '6': begin  X:=X+1;  if x>80 then x:=80;  end;
208         '2': begin  Y:=Y+1;  if y>25 then y:=25;  end;
209         '4': begin  X:=X-1;  if x<1 then x:=1;  end;
210       else
211        write(c);
212       end;
213     gotoxy(x,y);
214    until upcase(c)<>^M;
215 end;
216 }
217
218 type
219   register=record
220              ax,bx,cx,dx,bp,si,di,ds,es,flags:integer;
```

```
221           end;
222 {procedure curser(r:integer);
223 var
224   reg:register;
225 begin
226   with reg do
227     begin
228       ax:=$0100;
229       cx:=r;
230       intr($10,reg);
231      end;
232 end;}
233
234
235
236
237
238
239 {function inkey(var i:integer):boolean;
240 var
241  reg:register;
242 begin
243  reg.ax := $0000;
244  intr($16,reg);
245  if lo(reg.ax) = 0 then
246     i:= hi (reg.ax)
247  else
248     i:= lo (reg.ax);
249   inkey:=lo(reg.ax)=0;
250 end;
251
252 var
253   i:integer;
254 begin
255    repeat
256      if inkey(i) then
257      writeln(i)
258      else
259        writeln(chr(i));
260    until i=13;
261 end.
262 }
```

```
 1 TYPE NAME20= STRING[20];
 2      NAME = STRING[10];
 3      AR   = ARRAY[1..10] OF REAL;
 4      AS   = ARRAY[1..10] OF NAME;
 5      A2R  = ARRAY[1..10,1..100] OF REAL;
 6      AI   = ARRAY[1..10] OF INTEGER;
 7
 8 VAR
 9      N,I,M,N1,IND,K,stage_no:INTEGER;
10      NCELL:AI;
11      SG,FEED,CONC,TAIL,PULP,SAVE,TEMP,ASSAY,PAR,RATEF,RATEN,RATEFR,RATENR:AR;
12      CL,CLL,LCL,CONC1,CONC2,CONC3,CONC4,CONC5,TAIL1,TAIL3,TAIL4,TAIL5:AR;
13      BCONC1,BCONC2,BCONC3,BCONC4,BCONC5:A2R;
14      COMP : AS;
15      SOLIDS,H,G,WESTAR,SUMF,ALPHA,HNEW,GRADE,PCSLD : REAL;
16      SOLID1,SOLID2,SOLID3,SOLID4,SOLID5:REAL;
17      REC1,REC2,REC3,REC4,REC5,WREC1,WREC2,WREC3,WREC4,WREC5:REAL;
18      NAMES,MTCOMP : NAME;
19
20
21 PROCEDURE RDCOMP (VAR N:INTEGER; VAR COMP:AS);
22 {23 MARCH 1987
23  S. SAISINCHAI
24
25  READ COMPONENT TYPE NAME FOR FLOTATION
26
27  N = NO. OF COMPONENT EXCLUDING WATER
28 }
29
30 {TYPE NAME = STRING[10];
31 VAR  N:INTEGER;
32      COMP : ARRAY[1..10] OF NAME;}
33 VAR  I:INTEGER;
34 BEGIN
35      clrscr;
36      drawbox11(20,1,60,12);
37      window(20,1,60,12);
38      screenattr(7);
39      writestr(5,2,'MR.SOMSAK  SAISINCHAI');
40      writestr(2,3,'NO. OF SOLID COMPONENT NORMAL  = 3');
41      writestr(2,4,'COMP(1) = WATER');
42      writestr(2,5,'COMP(2) = FAST FLOATING');
43      writestr(2,6,'COMP(3) = SLOW FLOATING');
44      writestr(2,7,'COMP(4) = GANGUE');
45
46      repeat
47           window(1,14,80,25);
48           clrscr;
49           gotoxy(1,1);
50           normvideo;
51           WRITE ('ENTER NO. OF SOLID COMPONENT = ');
52           READLN (N);
53           FOR I:=2 TO N+1 DO
54           BEGIN
55               WRITE('ENTER COMP(',I:1,')=');
```

```
56              READLN (COMP[I]);
57          END;
58          COMP[1] :='WATER';
59          window(1,1,80,25);
60      until yesorno;
61      clrscr;
62  END;
63
64
65 PROCEDURE PTCOMP(N:INTEGER; COMP:AS);
66 { 23 MARCH 1987
67   S. SAISINCHAI
68
69   PRINT COMPONENT TYPE NAME FOR FLOTATION
70 }
71
72 {TYPE NAME = STRING[10];
73 VAR  N:INTEGER;
74      COMP:ARRAY[1..10] OF NAME;  }
75 VAR  I:INTEGER;
76 BEGIN
77
78      WRITE ('SOLIDS':17);
79      FOR I:=1 TO N+1 DO
80          WRITE(COMP[I]:10);
81      WRITELN;
82 END;
83
84 PROCEDURE  RDPULP(N:INTEGER; VAR FEED,SG:AR);
85 {
86   23 MARCH 1987
87   S. SAISINCHAI
88
89   READ PULP DESCRIPTION FOR FLOTATION
90   INPUT TPH. FOR WATER AND % BY WT. FOR THE COMPONENTS
91   INPUT SG. FOR EACH COMPONENT
92 }
93 {TYPE NAME = STRING[10];
94 VAR  SG   : ARRAY[1..10] OF REAL;
95      FEED : ARRAY[1..10] OF REAL;
96 VAR  N,N1 : INTEGER;
97      SOLIDS : REAL;}
98 VAR  I,N1 :INTEGER;
99      SOLIDS:REAL;
100 BEGIN
101      N1:=N+1;
102      clrscr;
103      drawbox11(10,1,70,12);
104      window(10,1,70,12);
105      screenattr(23);
106      writestr(5,2,'MR.SOMSAK  SAISINCHAI');
107      writestr(2,3,'SOLIDS= RATE OF FEED ORE (IN THAILAND) 2-20 TON/HR.');
108      writestr(2,4,'SG(1) = WATER = 1');
109      writestr(2,5,'SG(2) = FAST FLOATING = 3.2');
110      writestr(2,6,'SG(3) = SLOW FLOATING = 2.8');
```

```
111      writestr(2,7,'SG(4) = GANGUE = 2.45');
112      writestr(2,8,'FEED(1) = WEIGHT WATER FLOWRATE ABOUT 3 TIME FEED ORE' );
113      writestr(2,9,'FEED(2) = WEIGHT FAST FLOATING FLOWRATE ABOUT 50%FEED ORE');
114      writestr(2,10,'FEED(3)= WEIGHT SLOW FLOATING FLOWRATE ABOUT 15%FEED ORE');
115      writestr(2,11,'FEED(4)= WEIGHT GANG FLOWRATE ABOUT 35%FEED ORE');
116      repeat
117           window(1,14,80,25);
118           clrscr;
119           gotoxy(1,1);
120           normvideo;
121           WRITE('ENTER FEED ORE TON/HR.  SOLIDS = ');
122           READLN(SOLIDS);
123           WRITELN('ENTER SPECIFIC GRAVITY OF SOLIDS  SG(I)');
124           FOR I:=2 TO N1 DO
125               BEGIN
126                    WRITE('SG(',I,')=');
127                    READLN(SG[I]);
128               END;
129           WRITELN('ENTER WEIGHT FLOWRATE  WATER = FEED1 (ton/hr),  OTHER FEED(I)(%)');
130           FOR I:=1 TO N1 DO
131               BEGIN
132                   WRITE('FEED(',I,')=');
133                   READLN(FEED[I]);
134               END;
135           FOR I:=2 TO N1 DO
136               FEED[I]:=(FEED[I]*(SOLIDS/SG[I]))/100;
137               window(1,1,80,25);
138      until yesorno;
139      clrscr;
140 END;
141
142
143
144 PROCEDURE PTPULP(NAMES:NAME; N:INTEGER; PULP:AR);
145 {
146   23 MARCH 1987
147   S. SAISINCHAI
148
149   PRINT PULP DESCRIPTION FOR FLOTATION
150   INPUT NAME OF THE STREAM IN CALLING PROCEDURE
151   E.G. CALL PTPULP('ROUGHER',N,FEED)
152 }
153 {VAR PULP:ARRAY[1..10] OF REAL; }
154 VAR TEMP:AR;
155     I:INTEGER;
156     SOLIDS:REAL;
157 BEGIN
158      SOLIDS:=0;
159      FOR I:=2 TO N+1 DO
160          SOLIDS:=SOLIDS+PULP[I];
161      FOR I:=2 TO N+1 DO
162          TEMP[I]:=100*PULP[I]/SOLIDS;
163      WRITE(NAMES,SOLIDS:10:2,PULP[1]:10:2);
164      FOR I:=2 TO N+1 DO
165      WRITE(TEMP[I]:10:2);
```

```
166 END;
167
168
169 PROCEDURE RDASSA(N:INTEGER; VAR MTCOMP:NAME; VAR ASSAY:AR);
170 {
171   28 MARCH
172   S. SAISINCHAI
173
174   READ ASSAY VALUES OF EACH COMPONENTS OR SPECIES(i.e. % metal)
175 }
176 VAR I:INTEGER;
177 BEGIN
178      clrscr;
179      drawbox11(10,1,70,12);
180      window(10,1,70,12);
181      screenattr(23);
182      writestr(5,2,'MR.SOMSAK  SAISINCHAI');
183      writestr(2,3,'NAME OF ORE MINERAL FOR FLOATING = %CAF2');
184      writestr(2,4,'ASSAY(1) = WATER = 0');
185      writestr(2,5,'ASSAY(2) = ASSAY OF FAST FLOATING ABOUT 100% ');
186      writestr(2,6,'ASSAY(3) = ASSAY OF SLOW FLOATING ABOUT 80%');
187      writestr(2,7,'ASSAY(4) = ASSAY OF GANGUE ABOUT 0%');
188      repeat
189           window(1,14,80,25);
190           clrscr;
191           gotoxy(1,1);
192           normvideo;
193           WRITE('ENTER NAME OF ORE MINERAL FOR FLOAT = ');
194           READLN(MTCOMP);
195           WRITELN('ENTER ASSAY VALUES OF EACH COMPONENT ' );
196           FOR I:=2 TO N+1 DO
197           BEGIN
198               WRITE('ASSAY(',I,')= ');
199               READLN(ASSAY[I]);
200           END;
201           window(1,1,80,25);
202      until yesorno;
203      clrscr;
204 END;
205
206
207 PROCEDURE PTASSA(N:INTEGER;MTCOMP:NAME;ASSAY:AR);
208 {
209   28 MARCH 1987
210   S. SAISINCHAI
211
212   PRINT METAL COMPONENTS AND PERCENT ASSAY IN EACH COMPONENTS
213 }
214 VAR  I:INTEGER;
215 BEGIN
216      WRITELN;
217      WRITE(MTCOMP:10,'              ');
218      FOR I:=2 TO N+1 DO
219          WRITE(ASSAY[I]:10:2);
220 END;
```

```
221
222
223 PROCEDURE RDCELL(N:INTEGER;STAGE:NAME; VAR PAR,RATEF,RATEN:AR);
224 {
225   28 MARCH 1987
226   S. SAISINCHAI
227
228   READ CELL DESCRIPTION
229   PAR(1) = VOL. OF THE CELL
230   PAR(2) = BETA CONSTANT
231   RATEF  = RATE CONSTANT OF FLOATING COMPONENTS
232   RATEN  = RATE CONSTANT OF NON-FLOATING COMPONENTS
233 }
234 VAR I:INTEGER;
235 BEGIN
236      clrscr;
237      drawbox11(10,1,70,13);
238      window(10,1,70,13);
239      screenattr(23);
240      writestr(30,2,stage);
241      writestr(5,2,'MR.SOMSAK  SAISINCHAI');
242      writestr(2,3,'PAR(1) = VOLUME OF CELL ABOUT 0.2-1 cu.m.');
243      writestr(2,4,'PAR(2) = BETA CONSTANT ABOUT 0.2-1');
244      writestr(2,5,'RATEF(1) = DUMMY = 0');
245      writestr(2,6,'RATEF(2) = RATE OF FAST FLOATING = 10-20');
246      writestr(2,7,'RATEF(3) = RATE OF SLOW FLOATING =  5-15');
247      writestr(2,8,'RATEF(4) = DUMMY = 0');
248      writestr(2,9,'RATEN(1) = RATE OF WATER = 1-3 ');
249      writestr(2,10,'RATEN(2)= DUMMY = 0');
250      writestr(2,11,'RATEN(3)= DUMMY = 0');
251      writestr(2,12,'RATEN(4)= RATE OF GANGUE = 0.1-1.5');
252      repeat
253           window(1,14,80,25);
254           clrscr;
255           gotoxy(1,1);
256           normvideo;
257           WRITE('ENTER VOLUME OF CELL  PAR(1) = ');
258           READLN(PAR[1]);
259           WRITE('ENTER BETA CONSTANT  PAR(2) = ');
260           READLN(PAR[2]);
261           WRITELN('ENTER RATE CONSTANTS OF FLOATING COMPONENTS  "0 FOR I=1,4" ');
262           FOR I:=1 TO N+1 DO
263               BEGIN
264                   WRITE('RATEF(',I,')= ');
265                   READLN(RATEF[I]);
266               END;
267           WRITELN('ENTER RATE CONSTANTS OF NON FLOATING COMPONENT  "0 FOR I=2,3" ');
268           FOR I:=1 TO N+1 DO
269               BEGIN
270                   WRITE('RATEN(',I,')= ');
271                   READLN(RATEN[I]);
272               END;
273           window(1,1,80,25);
274      until yesorno;
275      clrscr;
```

```
276 END;
277
278
279
280 PROCEDURE PTCELL(N:INTEGER;sname:name; PAR,RATEF,RATEN:AR);
281 { .
282   28 MARCH 1987
283   S. SAINSINCHAI
284
285   PRINT CELL DESCRIPTION
286 }
287 VAR I:INTEGER;
288 BEGIN
289      WRITELN;
290      WRITELN('FLOTATION RATE CONSTANTS',sname:10);
291      WRITELN;
292      WRITE('FLOATING':20);
293      FOR I:=1 TO N+1 DO
294          WRITE(RATEF[I]:10:2);
295      WRITELN;
296      WRITE('NON FLOATING':20);
297      FOR I:=1 TO N+1 DO
298          WRITE(RATEN[I]:10:2);
299      WRITELN;
300      WRITELN('VOL. OF CELL':10,'BETA':10);
301      WRITELN(PAR[1]:10:2,PAR[2]:10:2);
302 END;
303
304 PROCEDURE RDCELLNO (VAR NCELL : AI);
305 {
306 }
307 VAR STAGE_NO :INTEGER;
308 BEGIN
309      clrscr;
310      drawbox11(10,1,70,13);
311      window(10,1,70,13);
312      screenattr(23);
313      writestr(5,2,'MR.SOMSAK  SAISINCHAI');
314      writestr(2,3,'NCELL(1) = NO. OF ROUGHER CELLS');
315      writestr(2,4,'NCELL(2) = NO. OF SCAVENGER CELLS');
316      writestr(2,5,'NCELL(3) = NO. OF 1ST CLEANER CELLS');
317      writestr(2,6,'NCELL(4) = NO. OF 2ND CLEANER CELLS');
318      writestr(2,7,'NCELL(5) = NO. OF 3 RD CEANER CELLS');
319      writestr(2,8,'NCELL(6) = NO. OF 4 TH CLEANER CELLS');
320      writestr(2,9,'NCELL(7) = NO. OF 5 TH CLEANER CELLS ');
321      repeat
322           window(1,14,80,25);
323           clrscr;
324           gotoxy(1,1);
325           normvideo;
326           WRITE('ENTER NO. OF FLOTATION STAGE = ');
327           READLN(STAGE_NO);
328           FOR I:=1 TO STAGE_NO DO
329               BEGIN
330                   WRITE('NCELL(',I,')= ');
```

```
331                 READLN(NCELL[I]);
332               END;
333           window(1,1,80,25);
334     until yesorno;
335     clrscr;
336 END;
337
338
339 PROCEDURE CELL(N:INTEGER; VAR PAR,RATEF,RATEN,FEED,CONC,TAIL:AR);
340 {
341   21 MARCH 1987
342   S. SAISINCHAI
343
344   FLOTATION SIMULATION ALLOWING OVERLOADING CONDITION
345   TWO SETS OF RATE CONSTANTS FOR FLOATING AND NON-FLOATING COMPO.
346
347   ITERATION CALCULATION PROCEDURE
348 }
349 VAR CONCF,CONCN:AR;
350     I,M,K :INTEGER;
351     H,G,WKSSTAR,SUMF,SUMCF,SUMC,SUMCN,ALPHA,HNEW:REAL;
352 LABEL 1;
353
354 BEGIN
355     H:=1;G:=0;WKSTAR:=0;K:=1;SUMF:=0;
356     FOR I:=1 TO N+1 DO SUMF:=SUMF + FEED[I];
357     FOR M:=1 TO 50 DO
358         BEGIN
359             ALPHA := (SUMF-G)/PAR[1];
360             SUMCF := 0;
361             FOR I:= 1 TO N+1 DO
362                 BEGIN
363                   CONCF[I]:=(FEED[I]*RATEF[I])/(1*RATEF[I]+(H*ALPHA));
364                   SUMCF:=SUMCF+CONCF[i];
365                 END;
366              HNEW:=1+(H*PAR[2]*(SUMCF/PAR[1]));
367              SUMCN:=0;
368              CONCN[1]:=WKSTAR+RATEN[1]*SUMCF;
369              FOR I:=2 TO N+1 DO
370                  BEGIN
371                    CONCN[I]:=(FEED[I]*RATEN[I]*CONCN[1])/
372                              (RATEN[I]*CONCN[1]+ALPHA*PAR[1]);
373                    SUMCN := SUMCN+CONCN[I];
374                  END;
375              SUMC:=SUMCF+SUMCN+CONCN[1];
376              G:=SUMC;
377              IF(ABS(H-HNEW)/HNEW<0.00001) THEN
378                GOTO 1
379               ELSE
380                H:=HNEW;
381         END;
382 1:   FOR I:=1 TO N+1 DO
383       BEGIN
384           CONC[I]:=CONCF[I];
385           IF(CONC[I]=0)THEN CONC[I]:=CONCN[I];
```

```
386        END;
387      FOR I:=1 TO N+1 DO TAIL[I]:=FEED[I]-CONC[I];
388 END;
389
390
391 PROCEDURE BANK(N,NCELL:INTEGER;PAR,RATEF,RATEN,FEED:AR;VAR CONC,TAIL:AR;
392               VAR BCONC:A2R;VAR SOLIDS,REC,WREC:REAL);
393 {
394   21 MARCH 1987
395   S. SAISINCHAI
396
397   CALCULATE CONCENTRATE FLOW RATE OF N COMPONENTS
398   IN A BANK OF NCELL CELLS
399 }
400 VAR  I,K:INTEGER; TCONC,TEMP,TTAIL:AR; TSOLID,SUMFI,SUMCI,SUMPI:REAL;
401 BEGIN
402      FOR I:= 1 TO N+1 DO TEMP[I]:=FEED[I];
403      FOR I:= 1 TO N+1 DO
404          BEGIN
405               TTAIL[I]:=0;
406               TCONC[I]:=0;
407          END;
408      TSOLID:=0;
409      FOR K:=1 TO NCELL DO
410          BEGIN
411               SOLIDS:=0;
412               CELL(N,PAR,RATEF,RATEN,TEMP,CONC,TAIL);
413               FOR I:=2 TO N+1 DO SOLIDS:=SOLIDS+CONC[I];
414               FOR I:=1 TO N+1 DO
415                   BEGIN
416                        BCONC[I,K]:=CONC[I];
417                        TCONC[I] := TCONC[I]+BCONC[I,K];
418                   END;
419               TEMP[N+1+K]:=SOLIDS;
420               FOR I:=1 TO N+1 DO TEMP[I]:=TAIL[I];
421           END;
422      FOR I:=1 TO N+1 DO
423          BEGIN
424               CONC[I]:=TCONC[I];
425               TAIL[I]:=FEED[I]-CONC[I];
426          END;
427      FOR I:=1 TO K DO TSOLID:=TSOLID+TEMP[N+1+I];
428      SOLIDS:=TSOLID;
429      SUMCI:=0;SUMFI:=0;
430      FOR I:=2 TO N+1 DO
431          BEGIN
432               SUMCI:=SUMCI+CONC[I];
433               SUMFI:=SUMFI+FEED[I];
434          END;
435      REC:=(SUMCI/SUMFI)*100;
436      WREC:=(CONC[1]/FEED[1])*100;
437 END;
438
439 PROCEDURE CONV(N:INTEGER;VAR PULP,SAVE:AR; VAR IND:INTEGER);
440 {
```

```
441   21 MARCH 1987
442   S. SAISINCHAI
443
444   CALCULATE CONVERGENCE FOR CLOSED CIRCUIT CIRCULATING LOAD
445   GIVE FRACTIONAL DIFFERENCE BETWEEN NEW ESTIMATE AND CONCENTRATE
446   SPEED UP ITERATION BY FACTOR F
447 }
448 CONST F=0.2;
449 VAR I:INTEGER; C:REAL;
450 BEGIN
451      FOR I:=1 TO N+1 DO
452          BEGIN
453               C:=PULP[I]-SAVE[I];
454               PULP[I]:=PULP[I]+F*C;
455               SAVE[I]:=PULP[I];
456               IF(ABS(C)/(PULP[I]+0.0001)>0.00001)THEN IND:=1;
457          END;
458 END;
459
460
461 PROCEDURE PTBANK(SNAME:NAME; NCELL:INTEGER);
462 {
463   21 MARCH 1987
464   S. SAISINCHAI
465
466   PRINT BANK DESCRIPTION NAME, NO. OF CELLS
467 }
468 BEGIN
469     WRITELN;WRITELN;
470     WRITELN(SNAME:10,'        ','BANK');
471     WRITELN('NO.OF CELLS',NCELL:10,'  CELLS');
472 END;
473
474
475 PROCEDURE PPBANK(N,NCELL:INTEGER;COMP:AS;PULP:A2R);
476 {
477   21 MARCH 1987
478   S. SAISINCHAI
479
480   PRINT PULP FLOW RATE FROM EACH CELL IN THE BANK
481   OF NCELL CELLS AT STEADY STATE
482 }
483 VAR  I,K:INTEGER;
484      TEMP:ARRAY[1..100] OF REAL;
485      TSOLID:REAL;
486      TCONC:AR;
487 BEGIN
488      WRITELN;
489      WRITELN('CONC.FLOW RATE');
490      WRITELN('CELL.NO');
491      PTCOMP(N,COMP);
492      TSOLID:=0;
493      FOR I:=1 TO N+1   DO TCONC[I]:=0;
494      FOR K:=1 TO NCELL DO TEMP[K]:=0;
495      FOR K:=1 TO NCELL DO
```

```
496          BEGIN
497               FOR I:=2 TO N+1 DO TEMP[K]:=TEMP[K]+PULP[I,K];
498               TSOLID:=TSOLID+TEMP[K];
499               WRITE(K:3,'      ',TEMP[K]:8:3);
500               FOR I:=1 TO N+1 DO WRITE(PULP[I,K]:10:3);
501               WRITELN;
502               FOR I:=1 TO N+1 DO TCONC[I]:=TCONC[I]+PULP[I,K];
503          END;
504      WRITELN;
505      WRITELN('TOTAL FLOWRATE (CU.M./HR)');
506      WRITE('         ',TSOLID:8:3);
507      FOR I:=1 TO N+1 DO WRITE(TCONC[I]:10:3);
508      WRITELN;
509 END;
510
511
512 PROCEDURE RCBANK(N:INTEGER;FEED,CONC,SG:AR);
513 {
514   28 MARCH 1987
515   S. SAISINCHAI
516
517   CALCULATE RECOVERY OF SOLIDS AND WATER
518   FROM THE BANK OF NCELL CELLS
519   GANGUE MINERALS SPECIES IS THE LAST SPECIES(N+1)
520 }
521 VAR I:INTEGER;
522     SUMF,SUMC,TEMPR,TEMPWR:REAL;
523 BEGIN
524      SUMF:=0;sumc:=0;
525      FOR I:=2 TO N+1 DO
526          BEGIN
527               SUMF:=SUMF+FEED[I]*SG[I];
528               SUMC:=SUMC+CONC[I]*SG[I];
529          END;
530      TEMPR:=(SUMC/SUMF)*100;
531      TEMPWR:=CONC[1]/FEED[1]*100;
532      WRITELN;
533      WRITELN('RECOVERY%',TEMPR:20:2);
534      WRITE('WATER RECOVERY%',TEMPWR:14:2);
535 END;
536
537 PROCEDURE GRAD(N:INTEGER;VAR PULP,ASSAY,SG:AR;VAR GRADE,PCSLD:REAL);
538 {
539   23 MARCH 1987
540   S. SAISINCHAI
541
542   CALCULATE GRADE AND PERCENT SOLIDS OF STREAM PULPS
543 }
544 VAR I:INTEGER;
545     MASS,TMETAL:AR;
546     TMASS,METAL:REAL;
547     BEGIN
548          TMASS:=0;METAL:=0;
549          FOR I:=2 TO N+1 DO
550               BEGIN
```

```
551                 MASS[I]:=PULP[I]*SG[I];
552                 TMASS:=TMASS+MASS[I];
553                 TMETAL[I]:=MASS[I]*ASSAY[I]/100;
554                 METAL:=METAL+TMETAL[I];
555             END;
556
557         GRADE:=METAL/TMASS*100;
558         PCSLD:=TMASS/(TMASS+PULP[1])*100;
559     END;
560
561
562 PROCEDURE PTGRAD(NNAME:NAME20;GRADE,PCSLD:REAL);
563 {
564   23 MARCH 1987
565   S. SAISINCHAI
566
567   PRINT GRADE AND PERCENT SOLIDS OF PULP STREAMS FROM FLOTATION
568 }
569 BEGIN
570     WRITELN;
571     WRITELN(NNAME);
572     WRITELN('GRADE %METAL','            ',GRADE:6:3,'    %SOLIDS     ',PCSLD:6:2);
573 END;
574
575 PROCEDURE MTREC(N:INTEGER;FEED,CONC,SG,ASSAY:AR);
576 {
577   21 MARCH 1987
578   S. SAISINCHAI
579
580   CALCULATE RECOVERY OF METAL
581 }
582 VAR I:INTEGER;
583     FMASS,CMASS,TEMPF,TEMPC:AR;
584     MFEED,MCONC,RMET:REAL;
585 {
586   CACULATE METAL IN FEED SOLIDS
587 }
588 BEGIN
589      MFEED:=0;MCONC:=0;
590      FOR I:=2 TO N+1 DO
591          BEGIN
592              FMASS[I]:=FEED[I]*SG[I];
593              TEMPF[I]:=FMASS[I]*ASSAY[I]/100;
594              MFEED:=MFEED+TEMPF[I];
595              CMASS[I]:=CONC[I]*SG[I];
596              TEMPC[I]:=CMASS[I]*ASSAY[I]/100;
597              MCONC:=MCONC+TEMPC[I];
598          END;
599      RMET:=(MCONC/MFEED)*100;
600      WRITE('RECOVERY (METAL)%       ',RMET:6:2);
601 END;
602
603
604
605
```

```
 1 { flotpar.inc}
 2 type kn80  = string[80];
 3      kn8   = string[8];
 4      kas   = array[1..10] of kn8;
 5      kar10 = array[1..10] of real;
 6      kar100= array[1..100] of real;
 7      ka2r  = array[1..10,1..100] of real;
 8 var
 9      kname1                      : kn8;
10      kname                       : kn80;
11      d_filek,o_filek             : text;
12      kncell,j                    : integer;
13      kcvol,kbeta                 : real;
14      krcomp                      : kas;
15      kfeed,ktempk,krate,kffeed   : kar10;
16      kttail,kconcf,kh            : kar100;
17      kconc,ktail,kk              : ka2r;
18
19 procedure kget_data;
20 begin
21      assign(d_filek,'ink.dat');
22      reset(d_filek);
23      readln(d_filek,kname);
24      readln(d_filek,n,kncell,kcvol,kbeta);
25      for i:=1 to n+1 do read(d_filek,krcomp[i]);
26      readln(d_filek);
27      for i:=1 to n+1 do read(d_filek,kfeed[i]);
28      readln(d_filek);
29      for j:=1 to kncell do
30          begin
31              for i:=1 to n+1 do read(d_filek,kconc[i,j]);
32              readln(d_filek);
33          end;
34      close(d_filek);
35 end;
36
37 procedure kdisplay;
38 begin
39      clrscr;
40      writeln(kname);
41      writeln(n:10,kncell:10,kcvol:10:2,kbeta:10:2);
42      for i:=1 to n+1 do write(krcomp[i]:10);
43      writeln;
44      for i:=1 to n+1 do write(kfeed[i]:10:2);
45      writeln;
46      for j:=1 to kncell do
47          begin
48              for i:=1 to n+1 do write(kconc[i,j]:10:2);
49              writeln;
50          end;
51          repeat until keypressed;
52 end;
53
54 procedure kcalc;
55 begin
```

```
56      {calculate rate constant for water}
57      for i:=1 to n+1 do kffeed[i]:=kfeed[i];
58      for j:=1 to kncell do
59      begin
60           kconcf[j]:=0;
61           for i:=2 to n do kconcf[j]:=kconcf[j]+kconc[i,j];
62           kh[j]:=1/(1-kbeta*kconcf[j]/kcvol);
63           kk[1,j]:=kconc[1,j]/kconcf[j];
64      end;
65      {calculate tail from each cell}
66      for j:=1 to kncell do
67          begin
68               kttail[j]:=0;
69               for i:=1 to n+1 do
70                   begin
71                       ktail[i,j]:=kfeed[i]-kconc[i,j];
72                       kttail[j]:=kttail[j]+ktail[i,j];
73                       kfeed[i]:=ktail[i,j];
74                   end;
75          end;
76
77      {calculate rate constants for flotation}
78      for j:=1 to kncell do
79          begin
80              for i:=2 to n do kk[i,j]:=(kconc[i,j]*kh[j]*kttail[j])/(ktail[i,j]*kcvol);
81          end;
82
83      {calculate rate constant for gangue material}
84      for j:=1 to kncell do
85          kk[n+1,j]:=(kconc[n+1,j]*(1+kttail[j]))/(kfeed[n+1]*kconc[1,j]);
86
87      {calculate average rate constant}
88      for i:= 1 to n+1 do ktempk[i]:=0;
89      for j:=1 to kncell do
90          begin
91              for i:=1 to n+1 do ktempk[i]:=ktempk[i]+kk[i,j];
92          end;
93      for i:=1 to n+1 do krate[i]:=ktempk[i]/kncell;
94  end;
95
96  procedure kout_data;
97  begin
98      clrscr;
99      writeln(kname);
100     writeln('CELL NO.');
101     write('         ');
102     for i:=1 to n+1 do write(krcomp[i]:10);
103     writeln;
104     for j:=1 to kncell do
105         begin
106              write('    ',j:2,'   ');
107              for i:=1 to n+1 do write(kk[i,j]:10:2);
108              writeln;
109         end;
110     writeln('AVERAGE RATE PARAMETERS');
```

```
111      write('          ');
112      for i:=1 to n+1 do write(krate[i]:10:3);
113      writeln;
114      writeln('FLOW RATE (CU.M/HR)');
115      writeln('FEED');
116      write('          ');
117      for i:=1 to n+1 do write(kffeed[i]:10:3);
118      writeln;
119      writeln('CONCENTRATE                                          TAILING');
120      writeln('CELL NO.');
121      for j:=1 to kncell do
122          begin
123              write('    ',j:2,'   ');
124              for i:=1 to n+1 do
125                  begin
126                      write(kconc[i,j]:10:3);
127                  end;
128                  write(kttail[j]:10:3);
129                  writeln;
130          end;
131       writeln;
132       repeat until keypressed;
133 end;
134
135
136
137 procedure kout_datafile;
138 begin
139      assign(o_filek,'outk.dat');
140      rewrite(o_filek);
141      writeln(o_filek,kname);
142      writeln(o_filek,'CELL NO.');
143      write(o_filek,'          ');
144      for i:=1 to n+1 do write(o_filek,krcomp[i]:10);
145      writeln(o_filek);
146      for j:=1 to kncell do
147          begin
148              write(o_filek,'    ',j:2,'   ');
149              for i:=1 to n+1 do write(o_filek,kk[i,j]:10:2);
150              writeln(o_filek);
151          end;
152      writeln(o_filek,'AVERAGE RATE PARAMETERS');
153      write(o_filek,'          ');
154      for i:=1 to n+1 do write(o_filek,krate[i]:10:3);
155      writeln(o_filek);
156      writeln(o_filek,'FLOW RATE (CU.M/HR)');
157      writeln(o_filek,'FEED');
158      write(o_filek,'          ');
159      for i:=1 to n+1 do write(o_filek,kffeed[i]:10:3);
160      writeln(o_filek);
161      writeln(o_filek,'CONCENTRATE                                          TAILING');
162      writeln(o_filek,'CELL NO.');
163      for j:=1 to kncell do
164          begin
165              write(o_filek,'    ',j:2,'   ');
```

```
166              for i:=1 to n+1 do
167                  begin
168                      write(o_filek,kconc[i,j]:10:3);
169                  end;
170                  write(o_filek,kttail[j]:10:3);
171                  writeln(o_filek);
172          end;
173       writeln(o_filek);
174       close(o_filek);
175 end;
176
177 procedure kout_datafilep;
178 begin
179      assign(o_filek,'lpt1');
180      rewrite(o_filek);
181      writeln(o_filek,kname);
182      writeln(o_filek,'CELL NO.');
183      write(o_filek,'         ');
184      for i:=1 to n+1 do write(o_filek,krcomp[i]:10);
185      writeln(o_filek);
186      for j:=1 to kncell do
187          begin
188              write(o_filek,'    ',j:2,'   ');
189              for i:=1 to n+1 do write(o_filek,kk[i,j]:10:2);
190              writeln(o_filek);
191          end;
192      writeln(o_filek,'AVERAGE RATE PARAMETERS');
193      write(o_filek,'         ');
194      for i:=1 to n+1 do write(o_filek,krate[i]:10:3);
195      writeln(o_filek);
196      writeln(o_filek,'FLOW RATE (CU.M/HR)');
197      writeln(o_filek,'FEED');
198      write(o_filek,'         ');
199      for i:=1 to n+1 do write(o_filek,kffeed[i]:10:3);
200      writeln(o_filek);
201      writeln(o_filek,'CONCENTRATE                                          TAILING');
202      writeln(o_filek,'CELL NO.');
203      for j:=1 to kncell do
204          begin
205              write(o_filek,'    ',j:2,'   ');
206              for i:=1 to n+1 do
207                  begin
208                      write(o_filek,kconc[i,j]:10:3);
209                  end;
210                  write(o_filek,kttail[j]:10:3);
211                  writeln(o_filek);
212          end;
213       writeln(o_filek);
214       close(o_filek);
215 end;
216
217
218
219 procedure flotpar;
220 begin
```

```
221      clrscr;
222      writestr(35,12,'!COMPUTE K-RATE!');
223      writeln;
224      kget_data;
225      {kdisplay;}
226      kcalc;
227      {kout_data;}
228      kout_datafile;
229 end;
230
231
232
233
```

```
 1 {File_Ser.lib}
 2
 3
 4
 5 var
 6
 7       t_file,d_file,o_file,d_file1      :   text;
 8       data_file,title_file,output_file  :   string[12];
 9       dummy                             :   string[12];
10       re                                :   char;
11       name1                             :   string;
12
13 procedure pline;
14 var i:integer;
15 begin for i:=1 to 78 do write('-');writeln('-');end;
16
17 procedure pline1;
18 var k:integer;
19 begin for k:=1 to 49 do write('-');writeln('-');end;
20
21 procedure pline_p;
22 var k:integer;
23 begin for k:=1 to 79 do write(lst,'-');writeln(lst,'-');end;
24
25 procedure wait;
26 begin
27      border1(58,22,74,24);
28      gotoxy(60,23);write('Press any key');cursorOff;
29      read(kbd,re);cursorOn;
30 end;
31
32 procedure wait2;
33 begin
34      border1(31,22,47,24);
35      gotoxy(33,23);write('Press any key');cursorOff;
36      read(kbd,re);cursorOn;
37 end;
38
39
40 procedure uppercase;
41 var i:integer;
42     str:string[12];
43 begin
44        str:='';
45        for i:=1 to length(dummy)  do
46        begin
47        dummy[i]:=upcase(dummy[i]);
48        str := str+dummy[i];
49        end;
50 dummy := str;
51 end;
52
53 procedure upcaseStrDir;
54 var i:integer;
55     str:string[32];
```

```
 56 begin
 57        str:='';
 58        for i:=1 to length(strDir)  do
 59        begin
 60        strDir[i]:=upcase(strDir[i]);
 61        str := str+strDir[i];
 62        end;
 63 strDir := str;
 64 end;
 65
 66 procedure Directory;
 67 var File_name : searchRec;
 68     Fname     : string[8];
 69     Ext       : string[3];
 70     list      : string[12];
 71     m         : byte;
 72 begin
 73     m:= 1;
 74     FindFirst('*.*',Archive,file_name);
 75     while DosError = 0 do
 76     begin
 77         list:=file_name.name;
 78         Fname := copy(list,1,length(list)-4);
 79         Ext   := copy(list,length(list)-2,3);
 80         if m <= 17 then begin {1}
 81             gotoxy(3,3+m);write(list);end {1}
 82          else begin {2}
 83                 if m <= 34 then begin {3}
 84                    gotoxy(16,m-14);write(list);end{3}
 85                  else begin {4}
 86                     if m <= 51 then begin {5}
 87                     gotoxy(7,23);write('Press any key to continue.');
 88            gotoxy(43,14);write('                                          ');
 89                     gotoxy(69,13);read(kbd,re);
 90            for i:= 1 to 17 do begin
 91            gotoxy(2,3+i);write('                                           ');end;
 92            gotoxy(3,23);write('                                         ');
 93            m:=0;
 94                   end;{5}
 95                end;{4} end;{2}
 96      findnext(File_name);
 97         m := m+1;
 98     end;
 99       if m-1 = 0 then
100         begin sound(800);delay(220);nosound;
101         gotoxy(5,23);write('No data file in this directory !');
102         end;
103 end;
104
105 procedure main_menu;forward;
106
107 procedure CHANGE_FILE;
108 var loop:string[4];fileName:string[32];
109 newFile,FileInput:string[12];oldFile:file;
110 label 22,33,44;
```

```
111 begin
112   upcaseStrDir;
113   22:clrscr;
114   border1(1,1,40,21);border1(1,22,40,24);
115   gotoxy(15,2);write('Directory');
116   gotoxy(2,3);write('-----------------------------------');
117   border2(51,1,67,3);gotoxy(53,2);write('File Services');
118   border1(41,5,80,11);
119   gotoxy(43,6);reverse;write('L');normal;write('ogged drive : ',DrvName);
120   gotoxy(43,7);reverse;write('A');normal;write('ctive directory : ',strDir);
121   gotoxy(43,8);reverse;write('T');normal;write('itle File  : ',title_file);
122   gotoxy(43,9); write('Data  File  : ',data_file);
123   gotoxy(43,10);reverse;write('O');normal;write('utput File : ',output_file);
124   border1(41,12,80,15);gotoxy(53,13);write('Please Select..[ ]');
125   border1(46,16,76,23);
126   gotoxy(49,17);write('Select highlight letter or');
127   gotoxy(47,18);write('---------------------------');
128   gotoxy(49,19);write('R = Rename file');
129   gotoxy(49,20);write('E = Erase file');
130   gotoxy(49,21);write('C = Continue');
131   gotoxy(49,22);write('X = Exit');
132   directory;
133   33:repeat
134   gotoxy(69,13);repeat read(kbd,re);
135   gotoxy(3,23);write('                                          ');gotoxy(69,13);
136   re:=upcase(re);until re in ['L','A','C','T','R','E','X','O'];
137   case re of    {case 1}
138       'L':begin
139           gotoxy(43,14);write('                                  ');
140           gotoxy(43,14);write('New drive : ');
141           repeat until keypressed;
142           re := readkey;re := upcase(re);
143           if (re in ['A'..'Z']) then
144               begin  drvName:=upcase(re);end
145               else  goto 33;
146           gotoxy(69,13);
147           {$I-} chDir(drvName+':'+ '\') {$I+};
148           if IOresult <> 0 then begin {2}
149           gotoxy(3,23);textcolor(20);write('* Error * ');textcolor(2);
150           write('  Drive not ready!');sound(800);delay(220);nosound;
151           gotoxy(43,14);write('                          ');
152           gotoxy(58,6);write(drvName);
153           goto 33;end;
154           gotoxy(58,6);write(drvName);
155           gotoxy(43,14);write('                          ');
156            end;
157       'A':begin {1} 44:gotoxy(43,14);write('New directory : ');
158           readln(strDir);
159           if length(strDir)>0 then
160             begin  {2}
161               if copy(strDir,1,1) = ' '  then begin
162                 gotoxy(43,14);write('                                  ');
163                 goto 44;end;
164           if copy(strDir,1,1)<>'\' then strDir:=concat('\',strDir);
165           if length(strDir) > 1 then
```

```
166              begin
167                 if copy(strDir,length(strDir),1) ='\' then
168                 strDir:=copy(strDir,length(strDir)-1,1);end;
169              end {2}
170           else  goto 44;
171           gotoxy(62,7);write('                  ');
172           gotoxy(69,13);
173           {1}{$I-} chDir(drvName+':'+ strDir) {$I+};
174           if IOresult <> 0 then begin {2}
175           gotoxy(3,23);textcolor(20);write('* Error * ');textcolor(2);
176           write('  Directoty not valid!');sound(800);delay(220);nosound;
177           gotoxy(43,14);write('                         ');
178           goto 33;end;
179           for i:= 1 to 17 do begin
180           gotoxy(2,3+i);write('                                      ');end;
181           directory;
182           upcaseStrDir;
183           gotoxy(62,7);write(strDir);
184           gotoxy(43,14);write('                                    . ');
185           end;{1}
186       'T':begin gotoxy(43,14);write('Title filename [.TLE]: ');
187           readln(dummy);
188           uppercase;
189           title_file:=dummy;data_file:=dummy;output_file:=dummy;
190           title_file:= copy(title_file,1,8)+'.TLE';
191           data_file:=  copy(data_file,1,8)+'.DAT';
192           output_file:=copy(output_file,1,8)+'.OUT';
193           gotoxy(57,8);write('                    ');
194           gotoxy(57,8);write(title_file);
195           gotoxy(57,9);write('                    ');
196           gotoxy(57,9);write(data_file);
197           gotoxy(57,10);write('                    ');
198           gotoxy(57,10);write(output_file);
199           gotoxy(43,14);write('                                   ');
200           end;
201        'R':begin gotoxy(43,14);write('File to be renamed : ');
202           readln(FileInput);
203           gotoxy(43,14);write('                                    ');
204           assign(oldFile,FileInput);
205           gotoxy(43,14);write('New file name : ');
206           readln(newFile);
207           gotoxy(43,14);write('                                    ');
208           {$I-} rename(oldFile,newFile) {$I+};
209           check := (IOresult =0);
210           if not check then begin
211           gotoxy(43,14);write('                                    ');
212           gotoxy(3,23);textcolor(20);write('* Error * ');textcolor(2);
213           write('   File not found !');sound(800);delay(220);nosound;
214           goto 33;end;
215           for i:= 1 to 17 do begin
216           gotoxy(2,3+i);write('                                     ');end;
217           directory;
218           end;
219        'E':begin gotoxy(43,14);write('File to be eraseed : ');
220           readln(FileInput);
```

```
221           gotoxy(43,14);write('                                          ');
222           assign(oldFile,FileInput);
223           {$I-} erase(oldFile) {$I+};
224           check := (IOresult =0);
225           if not check then begin
226           gotoxy(43,14);write('                                          ');
227           gotoxy(3,23);textcolor(20);write('* Error * ');textcolor(2);
228           write('   File not found !');sound(800);delay(220);nosound;
229           goto 33;end;
230           for i:= 1 to 17 do begin
231           gotoxy(2,3+i);write('                                          ');end;
232           directory;
233           end;
234       'O':begin gotoxy(43,14);write('Output filename [.OUT]: ');
235           readln(dummy);
236           uppercase;
237           output_file:= copy(dummy,1,8)+'.OUT';
238           gotoxy(57,10);write('                    ');
239           gotoxy(57,10);write(output_file);
240           gotoxy(43,14);write('                                          ');
241           end;
242        'X':begin main_menu;exit;end;
243        'C': loop:='EXIT';
244 end;  {case 1}
245 until loop = 'EXIT';
246 end;{procedure change_file}
247
248 procedure file_name;
249 begin
250       if title_file = 'NONAME.TLE' then change_file;
251 end;
252
253 procedure error;
254 begin
255       clrscr;sound(800);delay(220);nosound;
256       gotoxy(33,10);write('File not found !');
257       wait2;
258       change_file;clrscr;
259 end;
260
261
262 PROCEDURE RDCOMPf (VAR N:INTEGER; VAR COMP:AS);
263 {23 MARCH 1987
264  S. SAISINCHAI
265
266  READ COMPONENT TYPE NAME FOR FLOTATION
267
268  N = NO. OF COMPONENT EXCLUDING WATER
269 }
270
271 {TYPE NAME = STRING[10];
272 VAR  N:INTEGER;
273       COMP : ARRAY[1..10] OF NAME;}
274 VAR  I:INTEGER;
275 BEGIN
```

```
276           READLN(d_file1,N);
277           FOR I:=2 TO N+1 DO
278           BEGIN
279              READ (d_file1,COMP[I]);
280           END;
281           readln(d_file1);
282           COMP[1] :='WATER';
283  END;
284
285
286 PROCEDURE  RDPULPf(N:INTEGER; VAR FEED,SG:AR);
287 {
288   23 MARCH 1987
289   S. SAISINCHAI
290
291   READ PULP DESCRIPTION FOR FLOTATION
292   INPUT TPH. FOR WATER AND % BY WT. FOR THE COMPONENTS
293   INPUT SG. FOR EACH COMPONENT
294 }
295 {TYPE NAME = STRING[10];
296 VAR  SG   : ARRAY[1..10] OF REAL;
297      FEED : ARRAY[1..10] OF REAL;
298 VAR  N,N1 : INTEGER;
299      SOLIDS : REAL;|
300 VAR  I,N1 :INTEGER;
301      solids:real;
302 BEGIN
303      N1:=N+1;
304           READLN(d_file1,SOLIDS);
305           FOR I:=2 TO N1 DO
306               BEGIN
307                    READ(d_file1,SG[I]);
308               END;
309               readln(d_file1);
310           FOR I:=1 TO N1 DO
311               BEGIN
312                   READ(d_file1,FEED[I]);
313               END;
314               readln(d_file1);
315           FOR I:=2 TO N1 DO
316               FEED[I]:=(FEED[I]*(SOLIDS/SG[I]))/100;
317 END;
318
319 PROCEDURE  RDPULPfw(N:INTEGER; VAR FEED,SG:AR;var solids:real);
320 {
321   23 MARCH 1987
322   S. SAISINCHAI
323
324   READ PULP DESCRIPTION FOR FLOTATION
325   INPUT TPH. FOR WATER AND % BY WT. FOR THE COMPONENTS
326   INPUT SG. FOR EACH COMPONENT
327 }
328 {TYPE NAME = STRING[10];
329 VAR  SG   : ARRAY[1..10] OF REAL;
330      FEED : ARRAY[1..10] OF REAL;
```

```
331 VAR  N,N1 : INTEGER;
332      SOLIDS : REAL;|
333 VAR  I,N1 :INTEGER;
334 BEGIN
335      N1:=N+1;
336           READLN(d_file1,SOLIDS);
337           FOR I:=2 TO N1 DO
338               BEGIN
339                    READ(d_file1,SG[I]);
340               END;
341               readln(d_file1);
342           FOR I:=1 TO N1 DO
343               BEGIN
344                   READ(d_file1,FEED[I]);
345               END;
346               readln(d_file1);
347 END;
348
349
350
351 PROCEDURE RDASSAf(N:INTEGER; VAR MTCOMP:NAME; VAR ASSAY:AR);
352 {
353   28 MARCH
354   S. SAISINCHAI
355
356   READ ASSAY VALUES OF EACH COMPONENTS OR SPECIES(i.e. % metal)
357 }
358 VAR I:INTEGER;
359 BEGIN
360           READLN(d_file1,MTCOMP);
361           FOR I:=2 TO N+1 DO
362           BEGIN
363               READ(d_file1,ASSAY[I]);
364           END;
365           readln(d_file1);
366 END;
367
368
369
370 PROCEDURE RDCELLf(N:INTEGER;STAGE:NAME; VAR PAR,RATEF,RATEN:AR);
371 {
372   28 MARCH 1987
373   S. SAISINCHAI
374
375   READ CELL DESCRIPTION
376   PAR(1) = VOL. OF THE CELL
377   PAR(2) = BETA CONSTANT
378   RATEF  = RATE CONSTANT OF FLOATING COMPONENTS
379   RATEN  = RATE CONSTANT OF NON-FLOATING COMPONENTS
380 }
381 VAR I:INTEGER;
382 BEGIN
383           READLN(d_file1,PAR[1],par[2]);
384           FOR I:=1 TO N+1 DO
385               BEGIN
```

```
386                 READ(d_file1,RATBF[I]);
387             END;
388             readln(d_file1);
389         FOR I:=1 TO N+1 DO
390             BEGIN
391                 READ(d_file1,RATBN[I]);
392             END;
393             readln(d_file1);
394 END;
395
396
397 PROCEDURE RDCELLNOf (VAR NCELL : AI);
398 VAR STAGE_NO :INTEGER;
399 BEGIN
400         READLN(d_file1,STAGE_NO);
401         FOR I:=1 TO STAGE_NO DO
402             BEGIN
403                 READ(d_file1,NCELL[I]);
404             END;
405             readln(d_file1);
406 END;
407 PROCEDURE RDCELLNOfw (VAR NCELL : AI;var stage_no:integer);
408 BEGIN
409         READLN(d_file1,STAGE_NO);
410         FOR I:=1 TO STAGE_NO DO
411             BEGIN
412                 READ(d_file1,NCELL[I]);
413             END;
414             readln(d_file1);
415 END;
416 PROCEDURE PTCOMPf(N:INTEGER; COMP:AS);
417 { 23 MARCH 1987
418   S. SAISINCHAI
419
420   PRINT COMPONENT TYPE NAME FOR FLOTATION
421 }
422
423 {TYPE NAME = STRING[10];
424 VAR  N:INTEGER;
425      COMP:ARRAY[1..10] OF NAME;  }
426 VAR  I:INTEGER;
427 BEGIN
428
429      WRITE (o_file,'SOLIDS':17);
430      FOR I:=1 TO N+1 DO
431          WRITE(o_file,COMP[I]:10);
432      WRITELN(o_file);
433 END;
434
435
436
437
438
439 PROCEDURE PTPULPf(NAMES:NAME; N:INTEGER; PULP:AR);
440 {
```

```
441   23 MARCH 1987
442   S. SAISINCHAI
443
444   PRINT PULP DESCRIPTION FOR FLOTATION
445   INPUT NAME OF THE STREAM IN CALLING PROCEDURE
446   E.G. CALL PTPULP('ROUGHER',N,FEED)
447 }
448 {VAR PULP:ARRAY[1..10] OF REAL; }
449 VAR TEMP:AR;
450     I:INTEGER;
451     SOLIDS:REAL;
452 BEGIN
453      SOLIDS:=0;
454      FOR I:=2 TO N+1 DO
455          SOLIDS:=SOLIDS+PULP[I];
456      FOR I:=2 TO N+1 DO
457          TEMP[I]:=100*PULP[I]/SOLIDS;
458      WRITE(o_file,NAMES,SOLIDS:10:2,PULP[1]:10:2);
459      FOR I:=2 TO N+1 DO
460      WRITE(o_file,TEMP[I]:10:2);
461 END;
462
463
464
465
466
467 PROCEDURE PTASSAf(N:INTEGER;MTCOMP:NAME;ASSAY:AR);
468 {
469   28 MARCH 1987
470   S. SAISINCHAI
471
472   PRINT METAL COMPONENTS AND PERCENT ASSAY IN EACH COMPONENTS
473 }
474 VAR  I:INTEGER;
475 BEGIN
476      WRITELN(o_file);
477      WRITE(o_file,MTCOMP:10,'                ');
478      FOR I:=2 TO N+1 DO
479          WRITE(o_file,ASSAY[I]:10:2);
480 END;
481
482
483
484
485
486
487 PROCEDURE PTCELLf(N:INTEGER; sname:name;PAR,RATEF,RATEN:AR);
488 {
489   28 MARCH 1987
490   S. SAINSINCHAI
491
492   PRINT CELL DESCRIPTION
493 }
494 VAR I:INTEGER;
495 BEGIN
```

```
496      WRITELN(o_file);
497      WRITELN(o_file,'FLOTATION RATE CONSTANTS',sname:10);
498      WRITELN(o_file);
499      WRITE(o_file,'FLOATING':20);
500      FOR I:=1 TO N+1 DO
501         WRITE(o_file,RATEF[I]:10:2);
502      WRITELN(o_file);
503      WRITE(o_file,'NON FLOATING':20);
504      FOR I:=1 TO N+1 DO
505         WRITE(o_file,RATEN[I]:10:2);
506      WRITELN(o_file);
507      WRITELN(o_file,'VOL. OF CELL':10,'BETA':10);
508      WRITELN(o_file,PAR[1]:10:2,PAR[2]:10:2);
509 END;
510
511
512
513
514
515
516
517 PROCEDURE PTBANKf(SNAME:NAME; NCELL:INTEGER);
518 {
519   21 MARCH 1987
520   S. SAISINCHAI
521
522   PRINT BANK DESCRIPTION NAME, NO. OF CELLS
523 }
524 BEGIN
525      WRITELN(o_file);WRITELN(o_file);
526      WRITELN(o_file,SNAME:10,'          ','BANK');
527      WRITELN(o_file,'NO.OF CELLS',NCELL:10,'  CELLS');
528 END;
529
530
531 PROCEDURE PPBANKf(N,NCELL:INTEGER;COMP:AS;PULP:A2R);
532 {
533   21 MARCH 1987
534   S. SAISINCHAI
535
536   PRINT PULP FLOW RATE FROM EACH CELL IN THE BANK
537   OF NCELL CELLS AT STEADY STATE
538 }
539 VAR  I,K:INTEGER;
540      TEMP:ARRAY[1..100] OF REAL;
541      TSOLID:REAL;
542      TCONC:AR;
543 BEGIN
544      WRITELN(o_file);
545      WRITELN(o_file,'CONC.FLOW RATE');
546      WRITELN(o_file,'CELL.NO');
547      PTCOMPf(N,COMP);
548      TSOLID:=0;
549      FOR I:=1 TO N+1   DO TCONC[I]:=0;
550      FOR K:=1 TO NCELL DO TEMP[K]:=0;
```

```
551      FOR K:=1 TO NCELL DO
552          BEGIN
553               FOR I:=2 TO N+1 DO TEMP[K]:=TEMP[K]+PULP[I,K];
554               TSOLID:=TSOLID+TEMP[K];
555               WRITE(o_file,K:3,'      ',TEMP[K]:8:3);
556               FOR I:=1 TO N+1 DO WRITE(o_file,PULP[I,K]:10:3);
557               WRITELN(o_file);
558               FOR I:=1 TO N+1 DO TCONC[I]:=TCONC[I]+PULP[I,K];
559          END;
560      WRITELN(o_file);
561      WRITELN(o_file,'TOTAL FLOWRATE (CU.M./HR)');
562      WRITE(o_file,'         ',TSOLID:8:3);
563      FOR I:=1 TO N+1 DO WRITE(o_file,TCONC[I]:10:3);
564      WRITELN(o_file);
565 END;
566
567 PROCEDURE RCBANKf(N:INTEGER;FEED,CONC,SG:AR);
568 {
569   28 MARCH 1987
570   S. SAISINCHAI
571
572   CALCULATE RECOVERY OF SOLIDS AND WATER
573   FROM THE BANK OF NCELL CELLS
574   GANGUE MINERALS SPECIES IS THE LAST SPECIES(N+1)
575 }
576 VAR I:INTEGER;
577     SUMF,SUMC,TEMPR,TEMPWR:REAL;
578 BEGIN
579      SUMF:=0; sumc:=0;
580      FOR I:=2 TO N+1 DO
581          BEGIN
582               SUMF:=SUMF+FEED[I]*SG[I];
583               SUMC:=SUMC+CONC[I]*SG[I];
584          END;
585      TEMPR:=(SUMC/SUMF)*100;
586      TEMPWR:=CONC[1]/FEED[1]*100;
587      WRITELN(o_file);
588      WRITELN(o_file,'RECOVERY%',TEMPR:20:2);
589      WRITE(o_file,'WATER RECOVERY%',TEMPWR:14:2);
590 END;
591
592
593
594 PROCEDURE PTGRADf(NNAME:NAME20;GRADE,PCSLD:REAL);
595 {
596   23 MARCH 1987
597   S. SAISINCHAI
598
599   PRINT GRADE AND PERCENT SOLIDS OF PULP STREAMS FROM FLOTATION
600 }
601 BEGIN
602      WRITELN(o_file);
603      WRITELN(o_file,NNAME);
604      WRITELN(o_file,'GRADE %METAL','          ',GRADE:6:3,'    %SOLIDS     ',PCSLD:6:2);
605 END;
```

```
606
607 PROCEDURE MTRECf(N:INTEGER;FEED,CONC,SG,ASSAY:AR);
608 {
609   21 MARCH 1987
610   S. SAISINCHAI
611
612   CALCULATE RECOVERY OF METAL
613 }
614 VAR I:INTEGER;
615     FMASS,CMASS,TEMPF,TEMPC:AR;
616     MFEED,MCONC,RMET:REAL;
617 {
618   CACULATE METAL IN FEED SOLIDS
619 }
620 BEGIN
621      MFEED:=0;MCONC:=0;
622      FOR I:=2 TO N+1 DO
623          BEGIN
624               FMASS[I]:=FEED[I]*SG[I];
625               TEMPF[I]:=FMASS[I]*ASSAY[I]/100;
626               MFEED:=MFEED+TEMPF[I];
627               CMASS[I]:=CONC[I]*SG[I];
628               TEMPC[I]:=CMASS[I]*ASSAY[I]/100;
629               MCONC:=MCONC+TEMPC[I];
630           END;
631      RMET:=(MCONC/MFEED)*100;
632      WRITE(o_file,'RECOVERY (METAL)%       ',RMET:6:2);
633 END;
634
635 procedure in_file;    {insim401.pas}
636 begin
637      assign(d_file1,'input.dat');
638      reset(d_file1);
639
640      RDCOMPf(N,COMP);
641      RDPULPf(N,FEED,SG);
642      RDASSAf(N,MTCOMP,ASSAY);
643      RDCELLf(N,'CLEANER',PAR,RATEF,RATEN);
644      RDCELLf(N,'ROUGHER',PAR,RATEFR,RATENR);
645      RDCELLNOf(NCELL);
646      close(d_file1);
647 end;
648 procedure rwin_file;    {insim401.pas}
649 begin
650      assign(d_file1,'input.pas');
651      reset(d_file1);
652      RDCOMPf(N,COMP);
653      RDPULPfw(N,FEED,SG,solids);
654      RDASSAf(N,MTCOMP,ASSAY);
655      RDCELLf(N,'CLEANER',PAR,RATEF,RATEN);
656      RDCELLf(N,'ROUGHER',PAR,RATEFR,RATENR);
657      RDCELLNOfw(NCELL,stage_no);
658      close(d_file1);
659 end;
660 procedure win_file;
```

```
661 begin
662      writeln(n:2);
663      for i:=2 to n+1 do write(comp[i]:10);writeln;
664      writeln(solids:10:2);
665      for i:=2 to n+1 do write(sg[i]:10:2);writeln;
666      for i:=1 to n+1 do write(feed[i]:10:2);writeln;
667      writeln(mtcomp);
668      for i:=2 to n+1 do write(assay[i]:10:2);writeln;
669      writeln(par[1]:10:2,par[2]:10:2);
670      for i:=1 to n+1 do write(ratef[i]:10:2);writeln;
671      for i:=1 to n+1 do write(raten[i]:10:2);writeln;
672      writeln(par[1]:10:2,par[2]:10:2);
673      for i:=1 to n+1 do write(ratefr[i]:10:2);writeln;
674      for i:=1 to n+1 do write(ratenr[i]:10:2);writeln;
675      writeln(stage_no:2);
676      for i:=1 to stage_no do write(ncell[i]:10);writeln;
677      repeat until keypressed;
678 end;
679
680
681 procedure in_direct;   {indirect.pas}
682 begin
683      RDCOMP(N,COMP);
684      RDPULP(N,FEED,SG);
685      RDASSA(N,MTCOMP,ASSAY);
686      RDCELL(N,'CLEANER',PAR,RATEF,RATEN);
687      RDCELL(N,'ROUGHER',PAR,RATEFR,RATENR);
688      RDCELLNO(NCELL);
689 end;
690
691 procedure read_datafile;
692 begin
693      assign(d_file,'input.pas');
694      reset(d_file);
695      readln(d_file,n);
696      for i:=2 to n+1 do read(d_file,comp[i]);
697      readln(d_file);
698      readln(d_file,solids);
699      for i:=2 to n+1 do read(d_file,sg[i]);
700      readln(d_file);
701      for i:=1 to n+1 do read(d_file,feed[i]);
702      readln(d_file);
703      readln(d_file,mtcomp);
704      for i:=2 to n+1 do read(d_file,assay[i]);
705      readln(d_file);
706      readln(par[1],par[2]);
707      for i:=1 to n+1 do read(d_file,ratef[i]);
708      readln(d_file);
709      for i:=1 to n+1 do read(d_file,raten[i]);
710      readln(d_file);
711      readln(d_file,par[1],par[2]);
712      for i:=1 to n+1 do read(d_file,ratefr[i]);
713      readln(d_file);
714      for i:=1 to n+1 do read(d_file,ratenr[i]);
715      readln(d_file);
```

```
716      readln(d_file,stage_no);
717      for i:=1 to stage_no do read(d_file,ncell[i]);
718 end;
719
720 procedure outs_direct;    {sodirect.pas}
721 begin
722               clrscr;
723               PTCOMP(N,COMP);
724               PTPULP('ROUGHER',N,FEED);
725               PTASSA(N,MTCOMP,ASSAY);
726               PTCELL(N,'CLEANER',PAR,RATEF,RATEN);
727               PTCELL(N,'ROUGHER',PAR,RATEFR,RATENR);
728               WRITELN;
729               WRITELN('NO. OF ITERATIONS = ',K:3);
730               PTBANK('ROUGHER ',NCELL[1]);          readln;
731               PPBANK(N,NCELL[1],COMP,BCONC1);        readln;
732               PTBANK('SCAVENG ',NCELL[2]);           readln;
733               PPBANK(N,NCELL[2],COMP,BCONC2);        readln;
734               PTBANK('1 CLEANER',NCELL[3]);          readln;
735               PPBANK(N,NCELL[3],COMP,BCONC3);        readln;
736               PTBANK('2 CLEANER',NCELL[4]);          readln;
737               PPBANK(N,NCELL[4],COMP,BCONC4);        readln;
738               PTBANK('3 CLEANER',NCELL[5]);          readln;
739               PPBANK(N,NCELL[5],COMP,BCONC5);        readln;
740               WRITELN('CONC.FLOW RATE FROM THE CIRCUIT');
741               WRITE('                ');
742               FOR I:=1 TO N+1 DO WRITE(CONC5[I]:10:3);
743               WRITELN;WRITELN;
744               WRITELN('TAILING FROM THE LAST BANK');
745               WRITE('                ');
746               FOR I:=1 TO N+1 DO WRITE(TAIL[I]:10:3);
747               WRITELN;WRITELN;
748               WRITELN('TOTAL FEED TO THE FIRST BANK');
749               WRITE('                ');
750               FOR I:=1 TO N+1 DO WRITE(CL[I]:10:3);
751               WRITELN;
752               RCBANK(N,FEED,CONC5,SG);
753               WRITELN;
754               GRAD(N,CONC5,ASSAY,SG,GRADE,PCSLD);
755               PTGRAD('FINAL    CONC.      ',GRADE,PCSLD);
756               MTREC(N,FEED,CONC5,SG,ASSAY);
757               WRITELN;
758               GRAD(N,CONC1,ASSAY,SG,GRADE,PCSLD);
759               PTGRAD('ROUGHER  CONC.      ',GRADE,PCSLD);
760               MTREC(N,CL,CONC1,SG,ASSAY);
761               WRITELN;
762               repeat until keypressed;
763 end;
764
765
766
767
768 procedure out_file;   {soutfile.pas}
769 begin
770               assign(o_file,'out.dat');
```

```
771                rewrite(o_file);
772                PTCOMPf(N,COMP);
773                PTPULPf('ROUGHER',N,FEED);
774                PTASSAf(N,MTCOMP,ASSAY);
775                PTCELLf(N,'CLEANER',PAR,RATEF,RATEN);
776                PTCELLf(N,'ROUGHER',PAR,RATEFR,RATENR);
777                WRITELN(o_file);
778                WRITELN(o_file,'NO. OF ITERATIONS = ',K:3);
779                PTBANKf('ROUGHER ',NCELL[1]);
780                PPBANKf(N,NCELL[1],COMP,BCONC1);
781                PTBANKf('SCAVENG ',NCELL[2]);
782                PPBANKf(N,NCELL[2],COMP,BCONC2);
783                PTBANKf('1 CLEANER',NCELL[3]);
784                PPBANKf(N,NCELL[3],COMP,BCONC3);
785                PTBANKf('2 CLEANER',NCELL[4]);
786                PPBANKf(N,NCELL[4],COMP,BCONC4);
787                PTBANKf('3 CLEANER',NCELL[5]);
788                PPBANKf(N,NCELL[5],COMP,BCONC5);
789                writeln(o_file);
790                WRITELN(o_file,'CONC.FLOW RATE FROM THE CIRCUIT');
791                WRITE(o_file,'                ');
792                FOR I:=1 TO N+1 DO WRITE(o_file,CONC5[I]:10:3);
793                writeln(o_file);writeln(o_file);
794                WRITELN(o_file,'TAILING FROM THE LAST BANK');
795                WRITE(o_file,'                ');
796                FOR I:=1 TO N+1 DO WRITE(o_file,TAIL[I]:10:3);
797                WRITELN(o_file);WRITELN(o_file);
798                WRITELN(o_file,'TOTAL FEED TO THE FIRST BANK');
799                WRITE(o_file,'                ');
800                FOR I:=1 TO N+1 DO WRITE(o_file,CL[I]:10:3);
801                WRITELN(o_file);
802                RCBANKf(N,FEED,CONC5,SG);
803                WRITELN(o_file);
804                GRAD(N,CONC5,ASSAY,SG,GRADE,PCSLD);
805                PTGRADf('FINAL    CONC.      ',GRADE,PCSLD);
806                MTRECf(N,FEED,CONC5,SG,ASSAY);
807                WRITELN(o_file);
808                GRAD(N,CONC1,ASSAY,SG,GRADE,PCSLD);
809                PTGRADf('ROUGHER  CONC.      ',GRADE,PCSLD);
810                MTRECf(N,CL,CONC1,SG,ASSAY);
811                WRITELN(o_file);
812                close(o_file);
813
814 end;
815
816 procedure out_filep;   {soutfile.pas}
817 begin
818                assign(o_file,'lpt1');
819                rewrite(o_file);
820                PTCOMPf(N,COMP);
821                PTPULPf('ROUGHER',N,FEED);
822                PTASSAf(N,MTCOMP,ASSAY);
823                PTCELLf(N,'CLEANER',PAR,RATEF,RATEN);
824                PTCELLf(N,'ROUGHER',PAR,RATEFR,RATENR);
825                WRITELN(o_file);
```

```
826             WRITELN(o_file,'NO. OF ITERATIONS = ',K:3);
827             PTBANKf('ROUGHER ',NCELL[1]);
828             PPBANKf(N,NCELL[1],COMP,BCONC1);
829             PTBANKf('SCAVENG ',NCELL[2]);
830             PPBANKf(N,NCELL[2],COMP,BCONC2);
831             PTBANKf('1 CLEANER',NCELL[3]);
832             PPBANKf(N,NCELL[3],COMP,BCONC3);
833             PTBANKf('2 CLEANER',NCELL[4]);
834             PPBANKf(N,NCELL[4],COMP,BCONC4);
835             PTBANKf('3 CLEANER',NCELL[5]);
836             PPBANKf(N,NCELL[5],COMP,BCONC5);
837             writeln(o_file);
838             WRITELN(o_file,'CONC.FLOW RATE FROM THE CIRCUIT');
839             WRITE(o_file,'                ');
840             FOR I:=1 TO N+1 DO WRITE(o_file,CONC5[I]:10:3);
841             writeln(o_file);writeln(o_file);
842             WRITELN(o_file,'TAILING FROM THE LAST BANK');
843             WRITE(o_file,'                ');
844             FOR I:=1 TO N+1 DO WRITE(o_file,TAIL[I]:10:3);
845             WRITELN(o_file);WRITELN(o_file);
846             WRITELN(o_file,'TOTAL FEED TO THE FIRST BANK');
847             WRITE(o_file,'                ');
848             FOR I:=1 TO N+1 DO WRITE(o_file,CL[I]:10:3);
849             WRITELN(o_file);
850             RCBANKf(N,FEED,CONC5,SG);
851             WRITELN(o_file);
852             GRAD(N,CONC5,ASSAY,SG,GRADE,PCSLD);
853             PTGRADf('FINAL    CONC.     ',GRADE,PCSLD);
854             MTRECf(N,FEED,CONC5,SG,ASSAY);
855             WRITELN(o_file);
856             GRAD(N,CONC1,ASSAY,SG,GRADE,PCSLD);
857             PTGRADf('ROUGHER  CONC.     ',GRADE,PCSLD);
858             MTRECf(N,CL,CONC1,SG,ASSAY);
859             WRITELN(o_file);
860             close(o_file);
861
862 end;
```

ภาคผนวก ช

การจัดเตรียมข้อมูลสำหรับแบบจำลองการลอยแร่เพื่อคำนวณค่าคงที่อัตราการลอยแร่

บรรทัด	ข้อมูล	ตัวแปร	คำอธิบาย
1	'ROUGHER BANK'	KNAME	ชื่อของชุดเซลล์ลอยแร่
2	3,4,1,1	N,KNCELL,KCVOL,KBETA	จำนวนกลุ่มแร่, จำนวนเซลล์, ปริมาตรเซลล์, ค่าคงที่เบต้า
3	'WATER''FFAST''FSLOW''GANGUE'	(KRCOMP(I),I=1,N+1)	ชื่อของกลุ่มแร่ที่ทำการลอย
4	6.005,0.470,0.045,0.443	(KFEED(I),I=1,N+1)	อัตราป้อน(ลบม./ชม.)ของน้ำและกลุ่มแร่ที่ทำการลอย
5	1.063,0.330,0.024,0.072	(KCONC(1,J),J=1,4)	ปริมาณที่ลอยได้(ลบม./ชม.)ของน้ำและกลุ่มแร่ของเซลล์ที่1
6	0.367,0.109,0.013,0.026	(KCONC(2,J),J=1,4)	ปริมาณที่ลอยได้(ลบม./ชม.)ของน้ำและกลุ่มแร่ของเซลล์ที่2
7	0.089,0.025,0.005,0.006	(KCONC(3,J),J=1,4)	ปริมาณที่ลอยได้(ลบม./ชม.)ของน้ำและกลุ่มแร่ของเซลล์ที่3
8	0.020,0.005,0.002,0.001	(KCONC(4,J),J=1,4)	ปริมาณที่ลอยได้(ลบม./ชม.)ของน้ำและกลุ่มแร่ของเซลล์ที่4

ภาคผนวก ฌ

การจัดเตรียมข้อมูลสำหรับแบบจำลองการลอยแร่

บรรทัด	ข้อมูล	ตัวแปร	คำอธิบาย
1	3	N	จำนวนของกลุ่มแร่ที่ทำการลอย
2	'FFAST' 'FSLOW' 'GANGUE'	(COMP(I),I=2,N+1)	ชื่อของกลุ่มแร่ที่ทำการลอย
3	2.5	SOLIDS	แร่ป้อน (ตัน/ชม)
4	3.2,2.8,2.45	(SG(I),I=2,N+1)	ถ.พ. ของกลุ่มแร่ที่ทำการลอย
5	6,28.15,0,71.85	(FEED(I),I=1,N+1)	ปริมาณน้ำที่ป้อน(ตัน/ชม)และอัตราส่วนของกลุ่มแร่ที่ป้อน
6	'%CaF_2'	MTCOMP	ชนิดของแร่ที่ทำการลอย
7	100.0,80.0,0.0	(ASSAY(I),I=2,N+1)	เปอร์เซ็นต์ของกลุ่มแร่ที่ทำการลอย
8	0.2,0.2	PAR(1),PAR(2)	ปริมาตรของเซลล์ลอยแร่และค่าคงที่เบต้า (ชุดขั้นสะอาด)
9	0,4.5,0,0	(RATEF(I),I=1,N+1)	ค่าคงที่อัตราการลอยแร่ของกลุ่มแร่ที่ลอยได้ (ชุดขั้นสะอาด)
10	3,0,0,0.25	(RATEN(I),I=1,N+1)	ค่าคงที่อัตราการลอยของน้ำและมลทิน (ชุดขั้นสะอาด)
11	0.2,0.2	PAR(1),PAR(2)	ปริมาตรของเซลล์ลอยแร่และค่าคงที่เบต้า (ชุดขั้นต้น)
12	0,20,0,0	(RATEFR(I),I=1,N+1)	ค่าคงที่อัตราการลอยแร่ของกลุ่มแร่ที่ลอยได้ (ชุดขั้นต้น)
13	3,0,0,1	(RATENR(I),I=1,N+1)	ค่าคงที่อัตราการลอยของน้ำและมลทิน (ชุดขั้นต้น)
14	5	STAGENO	จำนวนชุดการลอยแร่
15	4,4,2,1,1	(NCELL(I),I=1,STAGENO)	จำนวนเซลล์ลอยแร่ของแต่ละชุด

ประวัติผู้เขียน

นายสมศักดิ์ สายสินธุ์ชัย เกิดวันที่ 2 ตุลาคม 2503 ที่อำเภอเมือง จังหวัดภูเก็ต สำเร็จการศึกษาปริญญาตรีวิศวกรรมศาสตร์บัณฑิต สาขาวิศวกรรมเหมืองแร่และโลหะวิทยา มหาวิทยาลัยสงขลานครินทร์ ในปีการศึกษา 2525 เข้าทำงานครั้งแรกในบริษัทกระบี่ฟลูออไรท์ จำกัด, ผู้จัดการโรงงานบริษัทไทยฟลูออไรท์พรอเซสซิ่งจำกัด, ผู้จัดการฝ่ายเหมืองแร่บริษัทซีทรานไมนิ่งจำกัด, ผู้จัดการฝ่ายเหมืองแร่บริษัทบีเอสเอ็มซีไมนิ่งจำกัด, ปัจจุบันเป็นผู้จัดการโรงงานบริษัทกระบี่ฟลูออไรท์จำกัด