รายการอ้างอิง

ภาษาไทย

วิจิตร ตัณฑสุทธิ์, วันชัย ริจิวนิช, ศิริจันทร์ ทองประเสริฐ. 1994. ภาควิชาวิศวกรรมอุตสาหการ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย. การวิจัยดำเนินงาน. บริษัท ซีเอ็ดยูเคชั่น จำกัด

ภาษาอังกฤษ

Andrew Kusiak. 1991. The University of Iowa, Iowa city, IA. Branching algorithms for solving the group technology problem. Journal of manufacturing systems. Vol.10 No.14.

Askin Ronald G., and Subrmanian Subramanian P. 1987. A cost-based heuristic for group technology configuration, International Journal of Production Research. Vol.25 No.1 101-113.

Atitkul W.1994. Department of Manufacturing Engineering and Operations Management, University of Nottingham. Flowshop Problem :Tabu Search Technique.

Charles S. Snead. Group technology foundation for competitive manufacturing. Van Nostrand Reinhold.

De Werra, D., and Hertz, A. 1989. Tabu Search Techniques: Authorial and an Application to Neural Networks. OR Spektrum. Vol.11:131-141.

Divakar Rajamani, N. Singh, and Y.P. Aneja. 1990. Integrated design of cellular manufacturing systems in the presence of alternative process plan. International Journal Production Research. Vol.28 No.8.

Fogarty Donald W., Cfpim, Blackstone John H., JR., Cfpim, and Hoffmann Thomas R., Cfpim.1994. Production & Inventory Management. South-Western Publishing Co.

Glover Fred. 1994. Tabu Search: Improved Solution Alternatives. Mathematical Porgramming. Braun – Brumfield, Inc. USA.

Glover Fred. 1989. Tabu Search-Part I. ORSA Journal on Computing Vol.1 No.3, Summer.

Glover Fred. 1990. Tabu Search-Part II. ORSA Journal on Computing Vol.2 No.1, Winter.

Groover Mikell P. 1994. Automation Production Systems and Computer Integrated Manufacturing:Group Technology. Prentice-Hall International Editions.

Hertz, A., and de Werra, D. 1991. The Tabu Search Metaheuristic: How We Used It. Annals of Mathematics and Artificial Intelligence. Vol.1:111-121.

Kusiak Andrew. 1991. Branching Algorithms for Solving the Group Technology Problem. Journal of Manufacturing Systems. Vol.10 No.4.

Laguna Manual, Barnes J. Wesley, and Glover F. 1991. Tabu search methods for a single machine scheduling problem. Journal of Intelligent Manufacturing. Vol.2: 63-74.

Logendran R., Ramakrishna P., and Sriskandarajah C. 1994. Tabu search-based heuristics for cellular manufacturing systems in the presence of alternative process plan. International Journal of Production Research.Vol.32 No.2 273-297.

Mikell P. Groover. 1987. Automation, Production systems, and computer integrated manufacturing. Prentice Hall International.

O.Felix Offodile, Abraham Mehrez, and John Grznar. 1994. Kent state university, Kent, Ohio. Cellular manufacturing :A taxonomic review framework, Journal of manufacturing systems. Vol.14 No.3 Nov.

Rajamani D.,Singh N., and Anega Y.P. 1996. Design of cellular manufacturing systems. International Journal Production Research. Vol.34 No.7 1917-1928.

Singh Nanua, and Rajamani Divakar. 1996. Cellular Manufacturing Systems. Chapman & Hall.

T.A. Perrego, H.C. Petersen, and W.F. Hahn. 1995. The perrego algorithm: A flexible machine-component grouping algorithm based on group technology techniques. International Journal Production Research. Vol.33 No.6.

Wemmerlov Urban, and Hyer Nancy L. 1989. Cellular manufacturing in the U.S. Industry: a survey users. International Journal of Production Research Vol. 27 No.9 1511-1530.

Wen-Chyuan Chiang, Genaro J. Gutierrez, and Panagiotis Kouvelis. 1992. Simulated Annealing and Tabu Search. Intelligent Design and Manufacturing. John Wiley & Sons, Inc.

ภาคผนวก ก
ความหมายของค่าตัวแปรต่าง ๆ

k แทน ชิ้นงาน

m แทน เครื่องจักร

p แทน แผนกระบวนการผลิต

s แทน ขั้นตอนการทำงาน

f แทน ครอบครัวชิ้นงาน

c แทน เซล

Z_m แทน จำนวนเครื่องจักรชนิด m

N_{mf} แทน จำนวนเครื่องจักรชนิด m ในครอบครัว f

N_{mc} แทน จำนวนเครื่องจักรชนิด m ในเซล c

Y_{kp} = 1 ถ้าชิ้นงาน k ถูกผลิตโดยแผนกระบวนการผลิต p
0 กรณีอื่น

$X_{ms}(kp)$ = 1 ถ้าเครื่องจักร m ถูกใช้ในขั้นตอนการทำงาน s โดยผลิตชิ้นงาน k ใช้แผนกระบวนการผลิต p
0 กรณีอื่น

$c_{ms}(kp)$ แทน ต้นทุนการทำงานสำหรับเครื่องจักร m ถูกใช้ในขั้นตอนการทำงาน s โดยผลิตชิ้นงาน k ใช้แผนกระบวนการผลิต p

$t_{ms}(kp)$ แทน เวลาการทำงานสำหรับเครื่องจักร m ถูกใช้ในขั้นตอนการทำงาน s โดยผลิตชิ้นงาน k ใช้แผนกระบวนการผลิต p

$a_s(kp)$ = 1 ถ้าขั้นตอนการทำงาน s ถูกใช้ผลิตชิ้นงาน k ในแผนกระบวนการผลิต p
0 กรณีอื่น

β_{kf} = 1 ถ้าชิ้นงาน k อยู่ในครอบครัวชิ้นงาน f
0 กรณีอื่น

α_{ms} = 1 ถ้าเครื่องจักร m สามารทำงานในขั้นตอนการทำงาน s
0 กรณีอื่น

C_m แทน ต้นทุนสำหรับเครื่องจักร m

OC แทน งบประมาณสำหรับต้นทุนการทำงาน

b_m แทน เวลาสำหรับเครื่องจักร m

d_k แทน ความต้องการชิ้นงาน k

C_v แทน ต้นทุนพัสดุคงคลัง

n แทน จำนวนรอบของการผลิตชิ้นงานทุกประเภทใน 1 ปี

$\sum A_k$ แทน ค่าใช้จ่ายในการออกใบสั่งผลิตทุก ๆ ชิ้นงานต่อรอบการผลิต

J_k แทน ค่าเก็บรักษาชิ้นงาน

P_k แทน อัตราการผลิตชิ้นงาน

O_k แทน ต้นทุนการผลิตผันแปร

m_{sk} แทน Processing Time สำหรับแต่ละขั้นตอนการทำงานของชิ้นงาน k

$R_m(kp)$ = ต้นทุนการผลิตผันแปรสำหรับ Processing Time ของขั้นตอนการทำงาน s ชิ้นงาน k

ภาคผนวก ข
ตัวอย่างโปรแกรม LINDO

```
model:
min = 100*n11+100*n12+100*n13+150*n21+150*n22+150*n23+200*n31+200*n32+200*n33
    +300*n41+300*n42+300*n43+360*n51+360*n52+360*n53+450*n61+450*n62+450*n63
    +500*n71+500*n72+500*n73;
y11+y12 =1;
y21+y22+y23 =1;
y31+y32 =1;
y41+y42 =1;
y51+y52+y53 =1;
y61+y62 =1;
y71+y72 =1;
y81+y82 =1;
y91+y92+y93 =1;
y101+y102 =1;

x53111+x63111-y111 =0;
x14111+x34111-y111 =0;
x32112+x42112-y112 =0;
x35112+x65112-y112 =0;
x27112+x47112-y112 =0;
x11113+x21113-y113 =0;
x32113+x42113-y113 =0;
x56113+x76113-y113 =0;
x53121+x63121-y121 =0;
x14121+x34121-y121 =0;
x32122+x42122-y122 =0;
x35122+x65122-y122 =0;
x27122+x47122-y122 =0;
x11123+x21123-y123 =0;
x32123+x42123-y123 =0;
x56123+x76123-y123 =0;
x53131+x63131-y131 =0;
x11141+x21141-y141 =0;
x35141+x65141-y141 =0;
x53142+x63142-y142 =0;
```

x35142+x65142-y142 =0;

x1112+x2112-y12=0;

x1121+x2121-y21=0;

x1142+x2142-y42=0;

x1151+x2151-y51=0;

x1162+x2162-y62=0;

x1171+x2171-y71=0;

x1181+x2181-y81=0;

x1192+x2192-y92=0;

x3211+x4211-y11=0;

x3222+x4222-y22=0;

x3232+x4232-y32=0;

x3241+x4241-y41=0;

x3251+x4251-y51=0;

x3262+x4262-y62=0;

x3271+x4271-y71=0;

x3293+x4293-y93=0;

x5322+x6322-y22=0;

x5352+x6352-y52=0;

x5361+x6361-y61=0;

x5372+x6372-y72=0;

x5362+x6382-y82=0;

x5392+x6392-y92=0;

x53101+x63101-y101=0;

x1412+x3412-y12=0;

x1423+x3423-y23=0;

x1431+x3431-y31=0;

x1442+x3442-y42=0;

x1453+x3453-y53=0;

x1471+x3471-y71=0;

x1482+x3482-y82=0;

x1491+x3491-y91=0;

x1492+x3492-y92=0;

x14102+x34102-y102=0;

x3521+x6521-y21=0;

x3552+x6552-y52=0;

x3562+x6562-y62=0;

x3572+x6572-y72=0;

```
x3581+x6561-y81=0;
x3593+x6593-y93=0;
x5611+x7611-y11=0;
x5622+x7622-y22=0;
x5632+x7632-y32=0;
x5642+x7642-y42=0;
x5661+x7661-y61=0;
x5681+x7681-y81=0;
x5693+x7693-y93=0;
x56102+x76102-y102=0;
x2711+x4711-y11=0;
x2723+x4723-y23=0;
x2741+x4741-y41=0;
x2751+x4751-y51=0;
x2753+x4753-y53=0;
x2772+x4772-y72=0;
x2762+x4782-y82=0;
x2791+x4791-y91=0;
x27101+x47101-y101=0;

25*x1112+10*x1121+5*x1412+10*x1423+30*x1431+10*x11113+20*x11123+10*x14111
+35*x14121-100*n11 <=0;

20*x1142+55*x1151+35*x1162+45*x1171+55*x1442+45*x1453+10*x1471-100*n12 <=0;

30*x1181+4*x1192+45*x1482+30*x1491+20*x1492+35*x14102+15*x11141-100*n13 <=0;

15*x2112+40*x2121+10*x2711+40*x2723+30*x21113+15*x21123+35*x27112+10*x27122-100*n21
<=0;

35*x2142+10*x2151+35*x2162+40*x2171+45*x2741+20*x2751+40*x2753+50*x2772
-100*n22 <=0;

20*x2181+30*x2192+35*x2782+55*x2791+55*x27101+20*x21141-100*n23 <=0;

10*x3211+15*x3222+35*x3232+45*x3412+20*x3423+55*x3431+10*x3512+10*x32112
+10*x32113+15*x31122+25*x31123+25*x34111+20*x34121+40*x36112+5*x35122-100*n31 <=0;
```

30*x3241+20*x3251+40*x3262+20*x3271+35*x3442+60*x3453+20*x3471+10*x3552
+25*x3562+25*x3572-100*n32 <=0;

5*x3293+30*x3482+35*x3491+45*x3492+40*x34102+50*x3435+45*x3493+40*x35141+10*x35142
-100*n33 <=0;

40*x4211+45*x4222+15*x4232+45*x4711+50*x4723+5*x42112+5*x42113+5*x42122
+20*x42123+40*x47112+15*x47122-100*n41 <=0;

40*x4241+30*x4251+30*x4262+15*x4271+10*x4272+50*x4751+50*x4753+55*x4772
-100*n42 <=0;

5*x4293+45*x4782+35*x4791+40*x47101-100*n43 <=0;

35*x5322+35*x5611+10*x5622+35*x5632+35*x53111+25*x53121+10*x56113+15*x56123
-100*n51 <=0;

10*x5352+45*x5361+5*x5372+45*x5642+55*x5661-100*n52 <=0;

40*x5382+45*x5392+20*x53101+30*x5681+30*x5693+50*x56102+30*x53131+45*x53142-100*n53
<=0;

35*x6322+15*x6521+25*x63111+10*x63121+45*x65112+25*x65122-100*n61 <=0;

15*x6352+25*x6361+10*x6372+10*x6552+20*x6562+40*x6572-100*n62 <=0;

30*x6382+40*x6392+30*x63101+35*x6561+30*x6593+25*x63131+35*x63142+5*x65141
+15*x65142-100*n63 <=0;

20*x7611+50*x7622+40*x7632+20*x76113+30*x76123-100*n71 <=0;

35*x7642+55*x7661-100*n72 <=0;

20*x7681+40*x7693+35*x76102-100*n73 <=0;

15*x1112+5*x1121+15*x1142+50*x1151+35*x1162+25*x1171+35*x1181+50*x1192
+20*x2112+45*x2121+10*x2142+15*x2151+40*x2162+40*x2171+45*x2181+35*x2192

+10*x3211+25*x3222+20*x3232+30*x3241+20*x3251+45*x3262+35*x3271+10*x3293
+5*x4211+40*x4222+25*x4232+40*x4241+30*x4251+15*x4262+25*x4271+5*x4293
+45*x5322+20*x5352+45*x5361+10*x5372+45*x5382+45*x5392+25*x53101+35*x6322
+50*x6352+15*x6361+30*x6372+5*x6382+45*x6392+20*x63101+10*x1412+15*x1423
+25*x1431+25*x1442+10*x1453+15*x1471+5*x1482+30*x1491+40*x1492+45*x14102
+35*x3412+20*x3423+35*x3431+45*x3442+60*x3453+45*x3471+5*x3482+45*x3491
+45*x3492+30*x34102+20*x3521+20*x3552+45*x3562+15*x3572+10*x3581+35*x3593
+50*x6521+5*x6552+35*x6562+40*x6572+5*x6561+50*x6593+15*x5611+20*x5622
+35*x5632+35*x5642+35*x5661+40*x5681+55*x5693+45*x56102+15*x7611+4*x7622
+45*x7632+35*x7642+25*x7661+30*x7681+35*x7693+30*x76102+10*x2711+40*x2723
+25*x2741+25*x2751+45*x2753+45*x2772+45*x2782+30*x2791+40*x27101+10*x4711
+45*x4723+30*x4741+45*x4751+50*x4753+35*x4772+5*x4762+40*x4791+45*x47101
+5*x11113+15*x11123+15*x11141+25*x21113+25*x21123+15*x21141+10*x32112
+10*x32113+10*x32122+20*x32123+5*x42112+5*x42113+5*x42122+15*x42123+25*x53111
+20*x53121+20*x53131+35*x53142+20*x63111+10*x63121+20*x63131+25*x63142
+10*x14111+25*x14121+20*x34111+15*x34121+30*x35112+5*x35122+30*x35141
+10*x35142+35*x65112+20*x65122+5*x65141+20*x65142+10*x56113+10*x56123
+15*x76113+25*x76123+25*x27112+10*x27122+30*x47112+10*x47122<=2000;

n11>=0;
n12>=0;
n13>=0;
n21>=0;
n22>=0;
n23>=0;
n31>=0;
n32>=0;
n33>=0;
n41>=0;
n42>=0;
n43>=0;
n51>=0;
n52>=0;
n53>=0;
n61>=0;
n62>=0;
n63>=0;
n71>=0;

```
n72>=0;
n73>=0;

@bin(x1112);
@bin(x1121);
@bin(x1142);
@bin(x1151);
@bin(x1162);
@bin(x1171);
@bin(x1181);
@bin(x1192);
@bin(x2112);
@bin(x2121);
@bin(x2142);
@bin(x2151);
@bin(x2162);
@bin(x2171);
@bin(x2181);
@bin(x2192);
@bin(x3211);
@bin(x3222);
@bin(x3232);
@bin(x3241);
@bin(x3251);
@bin(x3262);
@bin(x3271);
@bin(x3293);
@bin(x4211);
@bin(x4222);
@bin(x4232);
@bin(x4241);
@bin(x4251);
@bin(x4262);
@bin(x4271);
@bin(x4293);
@bin(x5322);
@bin(x5352);
@bin(x5361);
```

@bin(x5372);
@bin(x5382);
@bin(x5392);
@bin(x53101);
@bin(x6322);
@bin(x6352);
@bin(x6361);
@bin(x6372);
@bin(x6382);
@bin(x6392);
@bin(x63101);
@bin(x1412);
@bin(x1423);
@bin(x1431);
@bin(x1442);
@bin(x1453);
@bin(x1471);
@bin(x1482);
@bin(x1491);
@bin(x1492);
@bin(x14102);
@bin(x3412);
@bin(x3423);
@bin(x3431);
@bin(x3442);
@bin(x3453);
@bin(x3471);
@bin(x3482);
@bin(x3491);
@bin(x3492);
@bin(x34102);
@bin(x3521);
@bin(x3552);
@bin(x3562);
@bin(x3572);
@bin(x3581);
@bin(x3593);
@bin(x6521);

@bin(x6552);
@bin(x6562);
@bin(x6572);
@bin(x6581);
@bin(x6593);
@bin(x5611);
@bin(x5622);
@bin(x5632);
@bin(x5642);
@bin(x5661);
@bin(x5681);
@bin(x5693);
@bin(x56102);
@bin(x7611);
@bin(x7622);
@bin(x7632);
@bin(x7642);
@bin(x7661);
@bin(x7681);
@bin(x7693);
@bin(x76102);
@bin(x2711);
@bin(x2723);
@bin(x2741);
@bin(x2751);
@bin(x2763);
@bin(x2772);
@bin(x2782);
@bin(x2791);
@bin(x27101);
@bin(x4711);
@bin(x4723);
@bin(x4741);
@bin(x4751);
@bin(x4763);
@bin(x4772);
@bin(x4782);
@bin(x4791);

@bin(x47101);
@bin(x53111);
@bin(x63111);
@bin(x14111);
@bin(x34111);
@bin(x32112);
@bin(x42112);
@bin(x35112);
@bin(x65112);
@bin(x27112);
@bin(x47112);
@bin(x11113);
@bin(x21113);
@bin(x32113);
@bin(x42113);
@bin(x56113);
@bin(x76113);
@bin(x53121);
@bin(x63121);
@bin(x14121);
@bin(x34121);
@bin(x32122);
@bin(x42122);
@bin(x35122);
@bin(x65122);
@bin(x27122);
@bin(x47122);
@bin(x11123);
@bin(x21123);
@bin(x32123);
@bin(x42123);
@bin(x56123);
@bin(x76123);
@bin(x53131);
@bin(x63131);
@bin(x11141);
@bin(x21141);
@bin(x35141);

```
@bin(x65141);
@bin(x53142);
@bin(x63142);
@bin(x35142);
@bin(x65142);
@bin(y11);
@bin(y12);
@bin(y21);
@bin(y22);
@bin(y23);
@bin(y31);
@bin(y32);
@bin(y41);
@bin(y42);
@bin(y51);
@bin(y52);
@bin(y53);
@bin(y61);
@bin(y62);
@bin(y71);
@bin(y72);
@bin(y61);
@bin(y82);
@bin(y91);
@bin(y92);
@bin(y93);
@bin(y101);
@bin(y102);
@bin(y111);
@bin(y112);
@bin(y113);
@bin(y121);
@bin(y122);
@bin(y123);
@bin(y131);
@bin(y141);
@bin(y142);
@gin(n11);
```

```
@gin(n12);
@gin(n13);
@gin(n21);
@gin(n22);
@gin(n23);
@gin(n31);
@gin(n32);
@gin(n33);
@gin(n41);
@gin(n42);
@gin(n43);
@gin(n51);
@gin(n52);
@gin(n53);
@gin(n61);
@gin(n62);
@gin(n63);
@gin(n71);
@gin(n72);
@gin(n73);
end
```

ภาคผนวก ค

ตัวอย่างผลการทดลองโปรแกรม LINDO

Rows= 131 Vars= 213 No. integer vars= 207 (all are linear)

Nonzeros= 636 Constraint nonz= 604(275 are +- 1) Density=0.023

Smallest and largest elements in absolute value= 1.00000 2000.00

No. < : 22 No. =: 87 No. > : 21, Obj=MIN, GUBs <= 85

Single cols= 7

Optimal solution found at step: 163683

Objective value: 1350.000

Branch count: 4597

Variable	Value	Reduced Cost
N11	1.000000	0.0000000E+00
N12	1.000000	100.0000
N13	1.000000	100.0000
N21	0.0000000E+00	0.0000000E+00
N22	1.000000	150.0000
N23	1.000000	0.0000000E+00
N31	0.0000000E+00	0.0000000E+00
N32	2.000000	200.0000
N33	0.0000000E+00	0.0000000E+00
N41	0.0000000E+00	180.0000
N42	0.0000000E+00	0.0000000E+00
N43	0.0000000E+00	0.0000000E+00
N51	0.0000000E+00	87.14285
N52	0.0000000E+00	0.0000000E+00
N53	1.000000	0.0000000E+00
N61	0.0000000E+00	272.2222
N62	0.0000000E+00	0.0000000E+00
N63	0.0000000E+00	0.0000000E+00
N71	0.0000000E+00	270.0000
N72	0.0000000E+00	0.0000000E+00
N73	0.0000000E+00	0.0000000E+00
Y11	0.0000000E+00	0.0000000E+00
Y12	1.000000	-1.833332
Y21	1.000000	0.0000000E+00

Y22	0.0000000E+00	58.50794
Y23	0.0000000E+00	0.0000000E+00
Y31	1.000000	0.0000000E+00
Y32	0.0000000E+00	0.0000000E+00
Y41	1.000000	0.0000000E+00
Y42	0.0000000E+00	0.0000000E+00
Y51	1.000000	0.0000000E+00
Y52	0.0000000E+00	233.3333
Y53	0.0000000E+00	0.0000000E+00
Y61	0.0000000E+00	0.0000000E+00
Y82	1.000000	0.0000000E+00
Y71	1.000000	0.0000000E+00
Y72	0.0000000E+00	0.0000000E+00
Y81	1.000000	0.0000000E+00
Y82	0.0000000E+00	0.0000000E+00
Y91	0.0000000E+00	0.0000000E+00
Y92	1.000000	0.0000000E+00
Y93	0.0000000E+00	0.0000000E+00
Y101	1.000000	0.0000000E+00
Y102	0.0000000E+00	0.0000000E+00
X53111	0.0000000E+00	47.55556
X63111	0.0000000E+00	0.0000000E+00
Y111	0.0000000E+00	44.44444
X14111	0.0000000E+00	0.0000000E+00
X34111	0.0000000E+00	50.00000
X32112	0.0000000E+00	14.00000
X42112	0.0000000E+00	0.0000000E+00
Y112	0.0000000E+00	134.0000
X35112	0.0000000E+00	0.0000000E+00
X65112	0.0000000E+00	0.0000000E+00
X27112	0.0000000E+00	4.500000
X47112	0.0000000E+00	0.0000000E+00
X11113	0.0000000E+00	0.0000000E+00
X21113	0.0000000E+00	45.00000
Y113	0.0000000E+00	0.0000000E+00
X32113	0.0000000E+00	14.00000
X42113	0.0000000E+00	0.0000000E+00
X56113	0.0000000E+00	32.28572

X76113	0.0000000E+00	52.00000
X53121	0.0000000E+00	47.93651
X63121	0.0000000E+00	0.0000000E+00
Y121	0.0000000E+00	17.77778
X14121	0.0000000E+00	0.0000000E+00
X34121	0.0000000E+00	40.00000
X32122	0.0000000E+00	0.0000000E+00
X42122	0.0000000E+00	6.000000
Y122	0.0000000E+00	25.00000
X35122	0.0000000E+00	0.0000000E+00
X65122	0.0000000E+00	34.44444
X27122	0.0000000E+00	0.0000000E+00
X47122	0.0000000E+00	3.000001
X11123	0.0000000E+00	0.0000000E+00
X21123	0.0000000E+00	22.50000
Y123	0.0000000E+00	0.0000000E+00
X32123	0.0000000E+00	0.0000000E+00
X42123	0.0000000E+00	24.00000
X56123	0.0000000E+00	39.42857
X76123	0.0000000E+00	69.00000
X53131	0.0000000E+00	0.0000000E+00
X63131	0.0000000E+00	112.5000
Y131	0.0000000E+00	0.0000000E+00
X11141	0.0000000E+00	0.0000000E+00
X21141	0.0000000E+00	0.0000000E+00
Y141	0.0000000E+00	0.0000000E+00
X35141	0.0000000E+00	19.04762
X65141	0.0000000E+00	22.50000
X53142	0.0000000E+00	4.761905
X63142	0.0000000E+00	162.2619
Y142	0.0000000E+00	0.0000000E+00
X35142	0.0000000E+00	0.0000000E+00
X65142	0.0000000E+00	62.73810
X1112	1.000000	-5.633333
X2112	0.0000000E+00	0.0000000E+00
X1121	1.000000	0.0000000E+00
X2121	0.0000000E+00	0.0000000E+00
X1142	0.0000000E+00	0.0000000E+00

X2142	0.0000000E+00	0.0000000E+00
X1151	1.000000	0.0000000E+00
X2151	0.0000000E+00	0.0000000E+00
X1162	0.0000000E+00	0.0000000E+00
X2162	1.000000	0.0000000E+00
X1171	0.0000000E+00	0.0000000E+00
X2171	1.000000	0.0000000E+00
X1181	0.0000000E+00	157.5000
X2181	1.000000	208.9286
X1192	1.000000	0.0000000E+00
X2192	0.0000000E+00	0.0000000E+00
X3211	0.0000000E+00	0.0000000E+00
X4211	0.0000000E+00	26.00000
X3222	0.0000000E+00	0.0000000E+00
X4222	0.0000000E+00	24.00000
X3232	0.0000000E+00	52.00000
X4232	0.0000000E+00	0.0000000E+00
X3241	1.000000	0.0000000E+00
X4241	0.0000000E+00	0.0000000E+00
X3251	1.000000	0.0000000E+00
X4251	0.0000000E+00	0.0000000E+00
X3262	1.000000	0.0000000E+00
X4262	0.0000000E+00	0.0000000E+00
X3271	1.000000	-116.6667
X4271	0.0000000E+00	183.3333
X3293	0.0000000E+00	0.0000000E+00
X4293	0.0000000E+00	0.0000000E+00
X5322	0.0000000E+00	29.77778
X6322	0.0000000E+00	0.0000000E+00
X5352	0.0000000E+00	0.0000000E+00
X6352	0.0000000E+00	0.0000000E+00
X5361	0.0000000E+00	0.0000000E+00
X6361	0.0000000E+00	20.37037
X5372	0.0000000E+00	0.0000000E+00
X6372	0.0000000E+00	38.88889
X5382	0.0000000E+00	0.0000000E+00
X6382	0.0000000E+00	135.0000
X5392	1.000000	0.0000000E+00

X6392	0.0000000E+00	180.0000
X53101	1.000000	215.0000
X63101	0.0000000E+00	0.0000000E+00
X1412	1.000000	0.0000000E+00
X3412	0.0000000E+00	33.33333
X1423	0.0000000E+00	0.0000000E+00
X3423	0.0000000E+00	40.00000
X1431	1.000000	-36.66667
X3431	0.0000000E+00	0.0000000E+00
X1442	0.0000000E+00	0.0000000E+00
X3442	0.0000000E+00	0.0000000E+00
X1453	0.0000000E+00	0.0000000E+00
X3453	0.0000000E+00	0.0000000E+00
X1471	0.0000000E+00	0.0000000E+00
X3471	1.000000	0.0000000E+00
X1482	0.0000000E+00	0.0000000E+00
X3482	0.0000000E+00	14.28571
X1491	0.0000000E+00	0.0000000E+00
X3491	0.0000000E+00	150.0000
X1492	1.000000	-21.42857
X3492	0.0000000E+00	0.0000000E+00
X14102	0.0000000E+00	0.0000000E+00
X34102	0.0000000E+00	19.04762
X3521	1.000000	0.0000000E+00
X6521	0.0000000E+00	26.66667
X3552	0.0000000E+00	0.0000000E+00
X6552	0.0000000E+00	8.148148
X3562	1.000000	0.0000000E+00
X6562	0.0000000E+00	16.29630
X3572	0.0000000E+00	0.0000000E+00
X6572	0.0000000E+00	32.59259
X3581	1.000000	-157.5000
X6581	0.0000000E+00	0.0000000E+00
X3593	0.0000000E+00	0.0000000E+00
X6593	0.0000000E+00	135.0000
X5611	0.0000000E+00	46.00000
X7611	0.0000000E+00	0.0000000E+00
X5622	0.0000000E+00	0.0000000E+00

X7622	0.0000000E+00	88.71428
X5632	0.0000000E+00	0.0000000E+00
X7632	0.0000000E+00	0.0000000E+00
X5642	0.0000000E+00	0.0000000E+00
X7642	0.0000000E+00	175.0000
X5661	0.0000000E+00	0.0000000E+00
X7661	0.0000000E+00	275.0000
X5681	1.000000	0.0000000E+00
X7661	0.0000000E+00	100.0000
X5693	0.0000000E+00	0.0000000E+00
X7693	0.0000000E+00	200.0000
X56102	0.0000000E+00	0.0000000E+00
X76102	0.0000000E+00	175.0000
X2711	0.0000000E+00	0.0000000E+00
X4711	0.0000000E+00	39.00000
X2723	0.0000000E+00	0.0000000E+00
X4723	0.0000000E+00	0.0000000E+00
X2741	0.0000000E+00	0.0000000E+00
X4741	1.000000	0.0000000E+00
X2751	1.000000	0.0000000E+00
X4751	0.0000000E+00	0.0000000E+00
X2753	0.0000000E+00	0.0000000E+00
X4753	0.0000000E+00	0.0000000E+00
X2772	0.0000000E+00	0.0000000E+00
X4772	0.0000000E+00	0.0000000E+00
X2782	0.0000000E+00	0.0000000E+00
X4782	0.0000000E+00	96.42857
X2791	0.0000000E+00	0.0000000E+00
X4791	0.0000000E+00	128.5714
X27101	1.000000	233.5714
X47101	0.0000000E+00	220.7143
X3512	0.0000000E+00	20.00000
X31122	0.0000000E+00	30.00000
X31123	0.0000000E+00	50.00000
X3435	0.0000000E+00	23.80952
X3493	0.0000000E+00	21.42857
X4272	0.0000000E+00	0.0000000E+00

Row	Slack or Surplus	Dual Price
1	1350.000	1.000000
2	0.0000000E+00	-81.00000
3	0.0000000E+00	-60.00000
4	0.0000000E+00	-110.0000
5	0.0000000E+00	0.0000000E+00
6	0.0000000E+00	0.0000000E+00
7	0.0000000E+00	0.0000000E+00
8	0.0000000E+00	-116.6667
9	0.0000000E+00	0.0000000E+00
10	0.0000000E+00	-21.42857
11	0.0000000E+00	0.0000000E+00
12	0.0000000E+00	-44.44444
13	0.0000000E+00	0.0000000E+00
14	0.0000000E+00	-6.000000
15	0.0000000E+00	-80.00000
16	0.0000000E+00	-48.00000
17	0.0000000E+00	0.0000000E+00
18	0.0000000E+00	-6.000000
19	0.0000000E+00	6.000000
20	0.0000000E+00	-17.77776
21	0.0000000E+00	0.0000000E+00
22	0.0000000E+00	0.0000000E+00
23	0.0000000E+00	-10.00000
24	0.0000000E+00	-15.00000
25	0.0000000E+00	0.0000000E+00
26	0.0000000E+00	0.0000000E+00
27	0.0000000E+00	0.0000000E+00
28	0.0000000E+00	0.0000000E+00
29	0.0000000E+00	0.0000000E+00
30	0.0000000E+00	0.0000000E+00
31	0.0000000E+00	4.761905
32	0.0000000E+00	-4.761905
33	0.0000000E+00	-22.50000
34	0.0000000E+00	-60.00000
35	0.0000000E+00	0.0000000E+00
36	0.0000000E+00	0.0000000E+00
37	0.0000000E+00	0.0000000E+00

38	0.0000000E+00	0.0000000E+00
39	0.0000000E+00	157.5000
40	0.0000000E+00	0.0000000E+00
41	0.0000000E+00	-20.00000
42	0.0000000E+00	-30.00000
43	0.0000000E+00	-18.00000
44	0.0000000E+00	0.0000000E+00
45	0.0000000E+00	0.0000000E+00
46	0.0000000E+00	0.0000000E+00
47	0.0000000E+00	-116.6667
48	0.0000000E+00	-21.42857
49	0.0000000E+00	-62.22222
50	0.0000000E+00	-233.3333
51	0.0000000E+00	0.0000000E+00
52	0.0000000E+00	-116.6667
53	0.0000000E+00	0.0000000E+00
54	0.0000000E+00	0.0000000E+00
55	0.0000000E+00	-135.0000
56	0.0000000E+00	-56.66667
57	0.0000000E+00	0.0000000E+00
58	0.0000000E+00	-110.0000
59	0.0000000E+00	0.0000000E+00
60	0.0000000E+00	0.0000000E+00
61	0.0000000E+00	0.0000000E+00
62	0.0000000E+00	0.0000000E+00
63	0.0000000E+00	0.0000000E+00
64	0.0000000E+00	-21.42657
65	0.0000000E+00	0.0000000E+00
66	0.0000000E+00	0.0000000E+00
67	0.0000000E+00	0.0000000E+00
68	0.0000000E+00	0.0000000E+00
69	0.0000000E+00	0.0000000E+00
70	0.0000000E+00	-157.5000
71	0.0000000E+00	0.0000000E+00
72	0.0000000E+00	-46.00000
73	0.0000000E+00	-26.28572
74	0.0000000E+00	-92.00000
75	0.0000000E+00	0.0000000E+00

76	0.0000000E+00	0.0000000E+00
77	0.0000000E+00	0.0000000E+00
78	0.0000000E+00	0.0000000E+00
79	0.0000000E+00	0.0000000E+00
80	0.0000000E+00	-15.00000
81	0.0000000E+00	-60.00000
82	0.0000000E+00	0.0000000E+00
83	0.0000000E+00	0.0000000E+00
84	0.0000000E+00	0.0000000E+00
85	0.0000000E+00	0.0000000E+00
86	0.0000000E+00	0.0000000E+00
87	0.0000000E+00	-21.42857
88	0.0000000E+00	135.0000
89	30.00000	0.0000000E+00
90	45.00000	0.0000000E+00
91	78.00000	0.0000000E+00
92	0.0000000E+00	1.500000
93	5.000000	0.0000000E+00
94	25.00000	0.0000000E+00
95	0.0000000E+00	2.000000
96	45.00000	0.0000000E+00
97	0.0000000E+00	0.4761905
98	0.0000000E+00	1.200000
99	0.0000000E+00	0.0000000E+00
100	0.0000000E+00	2.142857
101	0.0000000E+00	2.628572
102	0.0000000E+00	0.0000000E+00
103	5.000000	0.0000000E+00
104	0.0000000E+00	1.777778
105	0.0000000E+00	0.8148148
106	0.0000000E+00	4.500000
107	0.0000000E+00	2.300000
108	0.0000000E+00	5.000000
109	0.0000000E+00	5.000000
110	1225.000	0.0000000E+00
111	1.000000	0.0000000E+00
112	1.000000	0.0000000E+00
113	1.000000	0.0000000E+00

114	0.0000000E+00	0.0000000E+00
115	1.000000	0.0000000E+00
116	1.000000	0.0000000E+00
117	0.0000000E+00	0.0000000E+00
118	2.000000	0.0000000E+00
119	0.0000000E+00	0.0000000E+00
120	0.0000000E+00	0.0000000E+00
121	0.0000000E+00	0.0000000E+00
122	0.0000000E+00	0.0000000E+00
123	0.0000000E+00	0.0000000E+00
124	0.0000000E+00	0.0000000E+00
125	1.000000	0.0000000E+00
126	0.0000000E+00	0.0000000E+00
127	0.0000000E+00	0.0000000E+00
128	0.0000000E+00	0.0000000E+00
129	0.0000000E+00	0.0000000E+00
130	0.0000000E+00	0.0000000E+00
131	0.0000000E+00	0.0000000E+00

ภาคผนวก ง

รายละเอียดไฟล์ตัวหนังสือของโปรแกรม Model1

เนื้อหารายละเอียดในส่วนนี้จะเสนอถึงรายละเอียดไฟล์ตัวหนังสือของโปรแกรม Model1 หรือโปรแกรมการค้นหาคำตอบแบบทาบูของการเลือกเครื่องจักรในการผลิต โดยมีเงื่อนไขของโปรแกรมดังนี้

1. เลือก Neighborhood List แบบ Swap Pairwise Interchange หรือ Insertion Interchange
2. เลือก Tabu List Size 3, 5, 7
3. ต้นทุนแบบที่ 1

รายละเอียดของโปรแกรมเป็นดังนี้

```
Program enterorp;
uses crt,printer,swaplib;
type  planrect = record
          partnum  : integer;
          partna   : array[1..19] of string[3];
          plannum  : array[1..19] of shortint;
          planna   : array[1..19,1..3] of string[3];
          opmum    : array[1..19,1..3] of shortint;
          opma     : array[1..19,1..3,1..20] of string[3];
          end;
type oprrect = record
          opmumm  : shortint;
          opmame  : array[1..20] of string[3];
          machnum  : array[1..20] of shortint;
          machna   : array[1..20,1..20] of string[3];
          oprcost  : array[1..20,1..20] of integer;
          prodcost : array[1..20,1..20] of integer;
          end;

   fitype = file of planrect;
   fitype1 = file of oprrect;
var cnt,cnt1,part  : integer;
```

```
  cost : longint;
  a,b,c,i  : integer;
  d,l,m,n  : longint;
  planfi   : fitype;
  oprfi    : fitype1;
  planrec  : planrect;
  oprrec   : oprrect;
  tabu     : array[1..20,1..3] of longint;
  method   : array[1..20] of longint;
  solution : array[1..20] of longint;
  solve    : array[1..20] of longint;
  sol      : array[1..20] of longint;
  tabulist : array[1..7] of longint;
  tabupos  : array[1..7] of integer;
  aspiration : array[1..7] of longint;
  tempo    : array[1..20] of longint;
  ran1,ran2,count1,solve1,num,more : integer;
  zum1,minimum2,tabu1,minimum,sum,cunt : longint;
  num1,num2,num3,num4,num5,num6,num7,num8,num9,num10,num11,
  num12,num13,num14,num15,num16,num17,num18,num19,num20 : integer;
  a1,a2,a3,a4,a5,a6,a7,a8,a9,a10,a11,a12,a13,a14,a15,a16,a17,a18,a19,a20 : integer;

Procedure prepare(var fi : fitype);
var finame : string[12];
begin
 clrscr;
 write('Enter file "s name : '); readln(finame);
 assign(fi,finame);
 {$I-} reset(fi); {$I+}
 if ioresult <> 0 then
 begin rewrite(fi); close(fi); reset(fi); end;
end;

Procedure prepare1(var fi1 : fitype1);
```

```
var finame : string[12];
begin
 clrscr;
 write('Enter file "s name : '); readln(finame);
 assign(fi1,finame);
 {$I-} reset(fi1); {$I+}
 if ioresult <> 0 then
 begin rewrite(fi1); close(fi1); reset(fi1); end;
end;

Procedure Enpart;
var part,partnum1,plan,plannum1,opr,oprnum1 : integer;
begin
 prepare(planfi);
 with planrec do
begin
 clrscr;
 write('Enter number of Part : '); readln(partnum);
 for part := 1 to partnum do
begin
 clrscr;
 writeln('Part ',part);
 write('Enter number of Plan : '); readln(plannum[part]);
 for plan := 1 to plannum[part] do
begin
 clrscr;
 writeln('Part ',part,' Plan ',plan);
 write('Enter number of Operation : '); readln(oprnum[part,plan]);
 for opr := 1 to opmum[part,plan] do
begin
 clrscr;
 writeln('Part ',part,' Plan ',plan,' Operation ',opr);
 write('Enter Part name     : '); readln(partna[part]);
 write('Enter Plan name     : '); readln(planna[part,plan]);
```

```
 write('Enter Operation name : '); readln(opma[part,plan,opr]);
end;{for_opr}
end;{for_plan}
end;{for_part}
write(planfi,planrec);
end;{with}
close(planfi);
end;{main}

Procedure Enopr(var fi1 : fitype1);
var opr2,oprnum2,mach,machnum1 : integer;
begin
  prepare1(oprfi);
  with oprrec do
begin
  clrscr;
  write('Enter number of Operation : '); readln(oprnumm);
  for opr2 := 1 to oprnumm do
begin
  clrscr;
  writeln('Operation ',opr2);
  write('Enter number of machine : '); readln(machnum[opr2]);
  for mach :=  1 to machnum[opr2] do
begin
  clrscr;
  writeln('Operation ',opr2,' Machine ',mach);
  write('Enter Operation name  : '); readln(oprname[opr2]);
  write('Enter Machine name    : '); readln(machna[opr2,mach]);
  write('Enter Operation Cost  : '); readln(oprcost[opr2,mach]);
  write('Enter Production Cost : '); readln(prodcost[opr2,mach]);
 end;{for_mach}
 end;{for_opr}
  write(oprfi,oprrec);
 end;{with}
```

```
  close(oprfi);
end;{main}

Function compword(astr,wrd : string; p : byte) : boolean;
var s : byte;
begin
  compword := false;
  for s := 2 to length(wrd) do
begin
  p := p+1;
  if wrd[s] <> astr[p] then exit;
end;
  compword := true;
end;

Function findword(astr,wrd : string) : byte;
var p : byte;
begin
  for p := 1 to length(astr) do
  if wrd[1] = astr[p] then
  if compword(astr,wrd,p) then
begin
  findword := p; exit
end;
  findword := 0;
end;

Function count(num,tem : integer) : integer;
begin
  with planrec do
  with oprrec do
begin
  repeat
  num := num+1;
```

```
 findword(opma[cnt,cnt1,tem],opmame[num]);
 until opma[cnt,cnt1,tem]=opmame[num];
 count := num;
end;
end;

Procedure machine1;
begin
 with planrec do
 with oprrec do
begin
 writeln(machna[count(num1,1),a1]);
 cost :=oprcost[count(num1,1),a1]+prodcost[count(num1,1),a1];
 {writeln(cost);}
 sum:=sum+cost;
end; end;

Procedure machine2;
begin
 with planrec do
 with oprrec do
begin
 writeln(machna[count(num1,1),a1],'-',machna[count(num2,2),a2]);
 cost := oprcost[count(num1,1),a1]+oprcost[count(num2,2),a2]
      +prodcost[count(num1,1),a1]+prodcost[count(num2,2),a2];
 {writeln(cost);}
 sum:=sum+cost;
end;
end;

Procedure machine3;
begin
 with planrec do
 with oprrec do
```

```
begin
 writeln(machna[count(num1,1),a1],'-',machna[count(num2,2),a2]
 ,'-',machna[count(num3,3),a3]);
 cost :=oprcost[count(num1,1),a1]+oprcost[count(num2,2),a2]
      +oprcost[count(num3,3),a3]+prodcost[count(num1,1),a1]
      +prodcost[count(num2,2),a2]+prodcost[count(num3,3),a3];
 {writeln(cost);}
 sum:=sum+cost;
end;
end;

Procedure machine4;
begin
 with planrec do
 with oprrec do
begin
 writeln(machna[count(num1,1),a1],'-',machna[count(num2,2),a2]
 ,'-',machna[count(num3,3),a3],'-',machna[count(num4,4),a4]);
 cost :=oprcost[count(num1,1),a1]+oprcost[count(num2,2),a2]
      +oprcost[count(num3,3),a3]+oprcost[count(num4,4),a4]
      +prodcost[count(num1,1),a1]+prodcost[count(num2,2),a2]
      +prodcost[count(num3,3),a3]+prodcost[count(num4,4),a4];
 {writeln(cost);}
 sum:=sum+cost;
end; end;

Procedure machine5;
begin
 with planrec do
 with oprrec do
begin
 writeln(machna[count(num1,1),a1],'-',machna[count(num2,2),a2]
 ,'-',machna[count(num3,3),a3],'-',machna[count(num4,4),a4]
 ,'-',machna[count(num5,5),a5]);
```

```
 cost :=oprcost[count(num1,1),a1]+oprcost[count(num2,2),a2]
      +oprcost[count(num3,3),a3]+oprcost[count(num4,4),a4]
      +oprcost[count(num5,5),a5]+prodcost[count(num1,1),a1]
      +prodcost[count(num2,2),a2]+prodcost[count(num3,3),a3]
      +prodcost[count(num4,4),a4]+prodcost[count(num5,5),a5];
 {writeln(cost);}
 sum:=sum+cost;
end; end;

Procedure machine6;
begin
 with planrec do
 with oprrec do
begin
 writeln(machna[count(num1,1),a1],'-',machna[count(num2,2),a2]
 ,'-',machna[count(num3,3),a3],'-',machna[count(num4,4),a4]
 ,'-',machna[count(num5,5),a5],'-',machna[count(num6,6),a6]);
 cost :=oprcost[count(num1,1),a1]+oprcost[count(num2,2),a2]
      +oprcost[count(num3,3),a3]+oprcost[count(num4,4),a4]
      +oprcost[count(num5,5),a5]+oprcost[count(num6,6),a6]
      +prodcost[count(num1,1),a1]+prodcost[count(num2,2),a2]
      +prodcost[count(num3,3),a3]+prodcost[count(num4,4),a4]
      +prodcost[count(num5,5),a5]+prodcost[count(num6,6),a6];
 {writeln(cost);}
 sum:=sum+cost;
end; end;

Procedure machine7;
begin
 with planrec do
 with oprrec do
begin
 writeln(machna[count(num1,1),a1],'-',machna[count(num2,2),a2]
 ,'-',machna[count(num3,3),a3],'-',machna[count(num4,4),a4]
```

```
 ,'-',machna[count(num5,5),a5],'-',machna[count(num6,6),a6]
 ,'-',machna[count(num7,7),a7]);
 cost :=oprcost[count(num1,1),a1]+oprcost[count(num2,2),a2]
      +oprcost[count(num3,3),a3]+oprcost[count(num4,4),a4]
      +oprcost[count(num5,5),a5]+oprcost[count(num6,6),a6]
      +oprcost[count(num7,7),a7]+prodcost[count(num1,1),a1]
      +prodcost[count(num2,2),a2]+prodcost[count(num3,3),a3]
      +prodcost[count(num4,4),a4]+prodcost[count(num5,5),a5]
      +prodcost[count(num6,6),a6]+prodcost[count(num7,7),a7];
 {writeln(cost);}
 sum:=sum+cost;
end; end;

Procedure machine8;
begin
 with planrec do
 with oprrec do
begin
 writeln(machna[count(num1,1),a1],'-',machna[count(num2,2),a2]
 ,'-',machna[count(num3,3),a3],'-',machna[count(num4,4),a4]
 ,'-',machna[count(num5,5),a5],'-',machna[count(num6,6),a6]
 ,'-',machna[count(num7,7),a7],'-',machna[count(num8,8),a8]);
 cost :=oprcost[count(num1,1),a1]+oprcost[count(num2,2),a2]
      +oprcost[count(num3,3),a3]+oprcost[count(num4,4),a4]
      +oprcost[count(num5,5),a5]+oprcost[count(num6,6),a8]
      +oprcost[count(num7,7),a7]+oprcost[count(num8,8),a8]
      +prodcost[count(num1,1),a1]+prodcost[count(num2,2),a2]
      +prodcost[count(num3,3),a3]+prodcost[count(num4,4),a4]
      +prodcost[count(num5,5),a5]+prodcost[count(num6,6),a6]
      +prodcost[count(num7,7),a7]+prodcost[count(num8,8),a8];
 {writeln(cost);}
 sum:=sum+cost;
end; end;
```

```
Procedure machine9;
begin
 with planrec do
 with oprrec do
begin
 writeln(machna[count(num1,1),a1],'-',machna[count(num2,2),a2]
 ,'-',machna[count(num3,3),a3],'-',machna[count(num4,4),a4]
 ,'-',machna[count(num5,5),a5],'-',machna[count(num6,6),a6]
 ,'-',machna[count(num7,7),a7],'-',machna[count(num8,8),a8]
 ,'-',machna[count(num9,9),a9]);
 cost :=oprcost[count(num1,1),a1]+oprcost[count(num2,2),a2]
     +oprcost[count(num3,3),a3]+oprcost[count(num4,4),a4]
     +oprcost[count(num5,5),a5]+oprcost[count(num6,6),a6]
     +oprcost[count(num7,7),a7]+oprcost[count(num8,8),a8]
     +oprcost[count(num9,9),a9]+prodcost[count(num1,1),a1]
     +prodcost[count(num2,2),a2]+prodcost[count(num3,3),a3]
     +prodcost[count(num4,4),a4]+prodcost[count(num5,5),a5]
     +prodcost[count(num6,6),a6]+prodcost[count(num7,7),a7]
     +prodcost[count(num8,8),a8]+prodcost[count(num9,9),a9];
 {writeln(cost);}
 sum:=sum+cost;
end; end;

Procedure machine10;
begin
 with planrec do
 with oprrec do
begin
 writeln(machna[count(num1,1),a1],'-',machna[count(num2,2),a2]
 ,'-',machna[count(num3,3),a3],'-',machna[count(num4,4),a4]
 ,'-',machna[count(num5,5),a5],'-',machna[count(num6,6),a6]
 ,'-',machna[count(num7,7),a7],'-',machna[count(num8,8),a8]
 ,'-',machna[count(num9,9),a9],'-',machna[count(num10,10),a10]);
 cost :=oprcost[count(num1,1),a1]+oprcost[count(num2,2),a2]
```

```
      +oprcost[count(num3,3),a3]+oprcost[count(num4,4),a4]
      +oprcost[count(num5,5),a5]+oprcost[count(num6,6),a6]
      +oprcost[count(num7,7),a7]+oprcost[count(num8,8),a8]
      +oprcost[count(num9,9),a9]+oprcost[count(num10,10),a10]
      +prodcost[count(num1,1),a1]+prodcost[count(num2,2),a2]
      +prodcost[count(num3,3),a3]+prodcost[count(num4,4),a4]
      +prodcost[count(num5,5),a5]+prodcost[count(num6,6),a6]
      +prodcost[count(num7,7),a7]+prodcost[count(num8,8),a8]
      +prodcost[count(num9,9),a9]+prodcost[count(num10,10),a10];
 {writeln(cost);}
 sum:=sum+cost;
end; end;

Procedure machine11;
begin
 with planrec do
 with oprrec do
begin
 writeln(machna[count(num1,1),a1],'-',machna[count(num2,2),a2]
 ,'-',machna[count(num3,3),a3],'-',machna[count(num4,4),a4]
 ,'-',machna[count(num5,5),a5],'-',machna[count(num6,6),a6]
 ,'-',machna[count(num7,7),a7],'-',machna[count(num8,8),a8]
 ,'-',machna[count(num9,9),a9],'-',machna[count(num10,10),a10]
 ,'-',machna[count(num11,11),a11]);
 cost :=oprcost[count(num1,1),a1]+oprcost[count(num2,2),a2]
      +oprcost[count(num3,3),a3]+oprcost[count(num4,4),a4]
      +oprcost[count(num5,5),a5]+oprcost[count(num6,6),a6]
      +oprcost[count(num7,7),a7]+oprcost[count(num8,8),a8]
      +oprcost[count(num9,9),a9]+oprcost[count(num10,10),a10]
      +oprcost[count(num11,11),a11]+prodcost[count(num1,1),a1]
      +prodcost[count(num2,2),a2]+prodcost[count(num3,3),a3]
      +prodcost[count(num4,4),a4]+prodcost[count(num5,5),a5]
      +prodcost[count(num6,6),a6]+prodcost[count(num7,7),a7]
      +prodcost[count(num8,8),a8]+prodcost[count(num9,9),a9]
```

```
      +prodcost[count(num10,10),a10]+prodcost[count(num11,11),a11];
 {writeln(cost);}
 sum:=sum+cost;
end; end;

Procedure machine12;
begin
 with planrec do
 with oprrec do
begin
 writeln(machna[count(num1,1),a1],'-',machna[count(num2,2),a2]
 ,'-',machna[count(num3,3),a3],'-',machna[count(num4,4),a4]
 ,'-',machna[count(num5,5),a5],'-',machna[count(num6,6),a6]
 ,'-',machna[count(num7,7),a7],'-',machna[count(num8,8),a8]
 ,'-',machna[count(num9,9),a9],'-',machna[count(num10,10),a10]
 ,'-',machna[count(num11,11),a11],'-',machna[count(num12,12),a12]);
 cost :=oprcost[count(num1,1),a1]+oprcost[count(num2,2),a2]
      +oprcost[count(num3,3),a3]+oprcost[count(num4,4),a4]
      +oprcost[count(num5,5),a5]+oprcost[count(num6,6),a6]
      +oprcost[count(num7,7),a7]+oprcost[count(num8,8),a8]
      +oprcost[count(num9,9),a9]+oprcost[count(num10,10),a10]
      +oprcost[count(num11,11),a11]+oprcost[count(num12,12),a12]
      +prodcost[count(num1,1),a1]+prodcost[count(num2,2),a2]
      +prodcost[count(num3,3),a3]+prodcost[count(num4,4),a4]
      +prodcost[count(num5,5),a5]+prodcost[count(num6,6),a6]
      +prodcost[count(num7,7),a7]+prodcost[count(num8,8),a8]
      +prodcost[count(num9,9),a9]+prodcost[count(num10,10),a10]
      +prodcost[count(num11,11),a11]+prodcost[count(num12,12),a12];
 {writeln(cost);}
 sum:=sum+cost;
end; end;

Procedure machine13;
begin
```

```
 with planrec do
 with oprrec do
begin
 writeln(machna[count(num1,1),a1],'-',machna[count(num2,2),a2]
 ,'-',machna[count(num3,3),a3],'-',machna[count(num4,4),a4]
 ,'-',machna[count(num5,5),a5],'-',machna[count(num6,6),a6]
 ,'-',machna[count(num7,7),a7],'-',machna[count(num8,8),a8]
 ,'-',machna[count(num9,9),a9],'-',machna[count(num10,10),a10]
 ,'-',machna[count(num11,11),a11],'-',machna[count(num12,12),a12]
 ,'-',machna[count(num13,13),a13]);
 cost :=oprcost[count(num1,1),a1]+oprcost[count(num2,2),a2]
      +oprcost[count(num3,3),a3]+oprcost[count(num4,4),a4]
      +oprcost[count(num5,5),a5]+oprcost[count(num6,6),a6]
      +oprcost[count(num7,7),a7]+oprcost[count(num8,6),a6]
      +oprcost[count(num9,9),a9]+oprcost[count(num10,10),a10]
      +oprcost[count(num11,11),a11]+oprcost[count(num12,12),a12]
      +oprcost[count(num13,13),a13]+prodcost[count(num1,1),a1]
      +prodcost[count(num2,2),a2]+prodcost[count(num3,3),a3]
      +prodcost[count(num4,4),a4]+prodcost[count(num5,5),a5]
      +prodcost[count(num6,6),a6]+prodcost[count(num7,7),a7]
      +prodcost[count(num8,8),a8]+prodcost[count(num9,9),a9]
      +prodcost[count(num10,10),a10]+prodcost[count(num11,11),a11]
      +prodcost[count(num12,12),a12]+prodcost[count(num13,13),a13];
 {writeln(cost);}
 sum:=sum+cost;
end; end;

Procedure machine14;
begin
 with planrec do
 with oprrec do
begin
 writeln(machna[count(num1,1),a1],'-',machna[count(num2,2),a2]
 ,'-',machna[count(num3,3),a3],'-',machna[count(num4,4),a4]
```

```
,'-',machna[count(num5,5),a5],'-',machna[count(num6,6),a6]
,'-',machna[count(num7,7),a7],'-',machna[count(num8,8),a8]
,'-',machna[count(num9,9),a9],'-',machna[count(num10,10),a10]
,'-',machna[count(num11,11),a11],'-',machna[count(num12,12),a12]
,'-',machna[count(num13,13),a13],'-',machna[count(num14,14),a14]);
cost :=oprcost[count(num1,1),a1]+oprcost[count(num2,2),a2]
    +oprcost[count(num3,3),a3]+oprcost[count(num4,4),a4]
    +oprcost[count(num5,5),a5]+oprcost[count(num6,6),a6]
    +oprcost[count(num7,7),a7]+oprcost[count(num8,8),a8]
    +oprcost[count(num9,9),a9]+oprcost[count(num10,10),a10]
    +oprcost[count(num11,11),a11]+oprcost[count(num12,12),a12]
    +oprcost[count(num13,13),a13]+oprcost[count(num14,14),a14]
    +prodcost[count(num1,1),a1]+prodcost[count(num2,2),a2]
    +prodcost[count(num3,3),a3]+prodcost[count(num4,4),a4]
    +prodcost[count(num5,5),a5]+prodcost[count(num6,6),a6]
    +prodcost[count(num7,7),a7]+prodcost[count(num8,8),a8]
    +prodcost[count(num9,9),a9]+prodcost[count(num10,10),a10]
    +prodcost[count(num11,11),a11]+prodcost[count(num12,12),a12]
    +prodcost[count(num13,13),a13]+prodcost[count(num14,14),a14];
{ writeln(cost);}
 sum:=sum+cost;
end; end;

Procedure machine15;
begin
 with planrec do
 with oprrec do
begin
 writeln(machna[count(num1,1),a1],'-',machna[count(num2,2),a2]
 ,'-',machna[count(num3,3),a3],'-',machna[count(num4,4),a4]
 ,'-',machna[count(num5,5),a5] ,'-',machna[count(num6,6),a6]
 ,'-',machna[count(num7,7),a7],'-',machna[count(num8,8),a8]
 ,'-',machna[count(num9,9),a9],'-',machna[count(num10,10),a10]
 ,'-',machna[count(num11,11),a11],'-',machna[count(num12,12),a12]
```

```
  ,'-',machna[count(num13,13),a13],'-',machna[count(num14,14),a14]
  ,'-',machna[count(num15,15),a15]);
  cost :=oprcost[count(num1,1),a1]+oprcost[count(num2,2),a2]
      +oprcost[count(num3,3),a3]+oprcost[count(num4,4),a4]
      +oprcost[count(num5,5),a5]+oprcost[count(num6,6),a6]
      +oprcost[count(num7,7),a7]+oprcost[count(num8,8),a8]
      +oprcost[count(num9,9),a9]+oprcost[count(num10,10),a10]
      +oprcost[count(num11,11),a11]+oprcost[count(num12,12),a12]
      +oprcost[count(num13,13),a13]+oprcost[count(num14,14),a14]
      +oprcost[count(num15,15),a15]+prodcost[count(num1,1),a1]
      +prodcost[count(num2,2),a2]+prodcost[count(num3,3),a3]
      +prodcost[count(num4,4),a4]+prodcost[count(num5,5),a5]
      +prodcost[count(num6,6),a6]+prodcost[count(num7,7),a7]
      +prodcost[count(num8,8),a8]+prodcost[count(num9,9),a9]
      +prodcost[count(num10,10),a10]+prodcost[count(num11,11),a11]
      +prodcost[count(num12,12),a12]+prodcost[count(num13,13),a13]
      +prodcost[count(num14,14),a14]+prodcost[count(num15,15),a15];
  {writeln(cost);}
  sum:=sum+cost;
end; end;

Procedure machine16;
begin
  with planrec do
  with oprrec do
begin
  writeln(machna[count(num1,1),a1],'-',machna[count(num2,2),a2]
  ,'-',machna[count(num3,3),a3],'-',machna[count(num4,4),a4]
  ,'-',machna[count(num5,5),a5],'-',machna[count(num6,6),a6]
  ,'-',machna[count(num7,7),a7],'-',machna[count(num8,8),a8]
  ,'-',machna[count(num9,9),a9],'-',machna[count(num10,10),a10]
  ,'-',machna[count(num11,11),a11],'-',machna[count(num12,12),a12]
  ,'-',machna[count(num13,13),a13],'-',machna[count(num14,14),a14]
  ,'-',machna[count(num15,15),a15],'-',machna[count(num16,16),a16]);
```

```
  cost :=oprcost[count(num1,1),a1]+oprcost[count(num2,2),a2]
       +oprcost[count(num3,3),a3]+oprcost[count(num4,4),a4]
       +oprcost[count(num5,5),a5]+oprcost[count(num6,6),a6]
       +oprcost[count(num7,7),a7]+oprcost[count(num8,8),a8]
       +oprcost[count(num9,9),a9]+oprcost[count(num10,10),a10]
       +oprcost[count(num11,11),a11]+oprcost[count(num12,12),a12]
       +oprcost[count(num13,13),a13]+oprcost[count(num14,14),a14]
       +oprcost[count(num15,15),a15]+oprcost[count(num16,16),a16]
       +prodcost[count(num1,1),a1]+prodcost[count(num2,2),a2]
       +prodcost[count(num3,3),a3]+prodcost[count(num4,4),a4]
       +prodcost[count(num5,5),a5]+prodcost[count(num6,6),a6]
       +prodcost[count(num7,7),a7]+prodcost[count(num8,8),a8]
       +prodcost[count(num9,9),a9]+prodcost[count(num10,10),a10]
       +prodcost[count(num11,11),a11]+prodcost[count(num12,12),a12]
       +prodcost[count(num13,13),a13]+prodcost[count(num14,14),a14]
       +prodcost[count(num15,15),a15]+prodcost[count(num16,16),a16];
{  writeln(cost);}
  sum:=sum+cost;
end; end;

Procedure machine17;
begin
  with planrec do
  with oprrec do
begin
  writeln(machna[count(num1,1),a1],'-',machna[count(num2,2),a2]
  ,'-',machna[count(num3,3),a3],'-',machna[count(num4,4),a4]
  ,'-',machna[count(num5,5),a5],'-',machna[count(num6,6),a6]
  ,'-',machna[count(num7,7),a7],'-',machna[count(num8,8),a8]
  ,'-',machna[count(num9,9),a9],'-',machna[count(num10,10),a10]
  ,'-',machna[count(num11,11),a11],'-',machna[count(num12,12),a12]
  ,'-',machna[count(num13,13),a13],'-',machna[count(num14,14),a14]
  ,'-',machna[count(num15,15),a15],'-',machna[count(num16,16),a16]
  ,'-',machna[count(num17,17),a17]);
```

```
cost :=oprcost[count(num1,1),a1]+oprcost[count(num2,2),a2]
    +oprcost[count(num3,3),a3]+oprcost[count(num4,4),a4]
    +oprcost[count(num5,5),a5]+oprcost[count(num6,6),a6]
    +oprcost[count(num7,7),a7]+oprcost[count(num8,8),a8]
    +oprcost[count(num9,9),a9]+oprcost[count(num10,10),a10]
    +oprcost[count(num11,11),a11]+oprcost[count(num12,12),a12]
    +oprcost[count(num13,13),a13]+oprcost[count(num14,14),a14]
    +oprcost[count(num15,15),a15]+oprcost[count(num16,16),a16]
    +oprcost[count(num17,17),a17]+prodcost[count(num1,1),a1]
    +prodcost[count(num2,2),a2]+prodcost[count(num3,3),a3]
    +prodcost[count(num4,4),a4]+prodcost[count(num5,5),a5]
    +prodcost[count(num6,6),a6]+prodcost[count(num7,7),a7]
    +prodcost[count(num8,8),a8]+prodcost[count(num9,9),a9]
    +prodcost[count(num10,10),a10]+prodcost[count(num11,11),a11]
    +prodcost[count(num12,12),a12]+prodcost[count(num13,13),a13]
    +prodcost[count(num14,14),a14]+prodcost[count(num15,15),a15]
    +prodcost[count(num16,16),a16]+prodcost[count(num17,17),a17];
 {writeln(cost);}
 sum:=sum+cost;
end; end;

Procedure machine18;
begin
 with planrec do
 with oprrec do
begin
 writeln(machna[count(num1,1),a1],'-',machna[count(num2,2),a2]
 ,'-',machna[count(num3,3),a3],'-',machna[count(num4,4),a4]
 ,'-',machna[count(num5,5),a5],'-',machna[count(num6,6),a6]
 ,'-',machna[count(num7,7),a7],'-',machna[count(num8,8),a8]
 ,'-',machna[count(num9,9),a9],'-',machna[count(num10,10),a10]
 ,'-',machna[count(num11,11),a11],'-',machna[count(num12,12),a12]
 ,'-',machna[count(num13,13),a13],'-',machna[count(num14,14),a14]
 ,'-',machna[count(num15,15),a15],'-',machna[count(num16,16),a16]
```

```
,'-',machna[count(num17,17),a17],'-',machna[count(num18,18),a18]);
cost :=oprcost[count(num1,1),a1]+oprcost[count(num2,2),a2]
     +oprcost[count(num3,3),a3]+oprcost[count(num4,4),a4]
     +oprcost[count(num5,5),a5]+oprcost[count(num6,6),a6]
     +oprcost[count(num7,7),a7]+oprcost[count(num8,8),a8]
     +oprcost[count(num9,9),a9]+oprcost[count(num10,10),a10]
     +oprcost[count(num11,11),a11]+oprcost[count(num12,12),a12]
     +oprcost[count(num13,13),a13]+oprcost[count(num14,14),a14]
     +oprcost[count(num15,15),a15]+oprcost[count(num16,16),a16]
     +oprcost[count(num17,17),a17]+oprcost[count(num18,18),a18]
     +prodcost[count(num1,1),a1]+prodcost[count(num2,2),a2]
     +prodcost[count(num3,3),a3]+prodcost[count(num4,4),a4]
     +prodcost[count(num5,5),a5]+prodcost[count(num6,6),a6]
     +prodcost[count(num7,7),a7]+prodcost[count(num8,8),a8]
     +prodcost[count(num9,9),a9]+prodcost[count(num10,10),a10]
     +prodcost[count(num11,11),a11]+prodcost[count(num12,12),a12]
     +prodcost[count(num13,13),a13]+prodcost[count(num14,14),a14]
     +prodcost[count(num15,15),a15]+prodcost[count(num16,16),a16]
     +prodcost[count(num17,17),a17]+prodcost[count(num18,18),a18];
{ writeln(cost);}
 sum:=sum+cost;
end; end;

Procedure machine19;
begin
 with planrec do
 with oprrec do
begin
 writeln(machna[count(num1,1),a1],'-',machna[count(num2,2),a2]
 ,'-',machna[count(num3,3),a3],'-',machna[count(num4,4),a4]
 ,'-',machna[count(num5,5),a5],'-',machna[count(num6,6),a6]
 ,'-',machna[count(num7,7),a7],'-',machna[count(num8,8),a8]
 ,'-',machna[count(num9,9),a9],'-',machna[count(num10,10),a10]
 ,'-',machna[count(num11,11),a11],'-',machna[count(num12,12),a12]
```

```
,'-',machna[count(num13,13),a13],'-',machna[count(num14,14),a14]
,'-',machna[count(num15,15),a15],'-',machna[count(num16,16),a16]
,'-',machna[count(num17,17),a17],'-',machna[count(num18,18),a18]
,'-',machna[count(num19,19),a19]);
cost :=oprcost[count(num1,1),a1]+oprcost[count(num2,2),a2]
    +oprcost[count(num3,3),a3]+oprcost[count(num4,4),a4]
    +oprcost[count(num5,5),a5]+oprcost[count(num6,6),a6]
    +oprcost[count(num7,7),a7]+oprcost[count(num6,8),a8]
    +oprcost[count(num9,9),a9]+oprcost[count(num10,10),a10]
    +oprcost[count(num11,11),a11]+oprcost[count(num12,12),a12]
    +oprcost[count(num13,13),a13]+oprcost[count(num14,14),a14]
    +oprcost[count(num15,15),a15]+oprcost[count(num16,16),a16]
    +oprcost[count(num17,17),a17]+oprcost[count(num18,18),a18]
    +oprcost[count(num19,19),a19]+oprcost[count(num20,20),a20]
    +prodcost[count(num1,1),a1]+prodcost[count(num2,2),a2]
    +prodcost[count(num3,3),a3]+prodcost[count(num4,4),a4]
    +prodcost[count(num5,5),a5]+prodcost[count(num6,6),a6]
    +prodcost[count(num7,7),a7]+prodcost[count(num8,8),a8]
    +prodcost[count(num9,9),a9]+prodcost[count(num10,10),a10]
    +prodcost[count(num11,11),a11]+prodcost[count(num12,12),a12]
    +prodcost[count(num13,13),a13]+prodcost[count(num14,14),a14]
    +prodcost[count(num15,15),a15]+prodcost[count(num16,16),a16]
    +prodcost[count(num17,17),a17]+prodcost[count(num18,18),a18]
    +prodcost[count(num19,19),a19];
 {writeln(cost);}
 sum:=sum+cost;
end; end;

Procedure machine20;
begin
 with planrec do
 with oprrec do
begin
 writeln(machna[count(num1,1),a1],'-',machna[count(num2,2),a2]
```

```
,'-',machna[count(num3,3),a3],'-',machna[count(num4,4),a4]
,'-',machna[count(num5,5),a5],'-',machna[count(num6,6),a6]
,'-',machna[count(num7,7),a7],'-',machna[count(num8,8),a8]
,'-',machna[count(num9,9),a9],'-',machna[count(num10,10),a10]
,'-',machna[count(num11,11),a11],'-',machna[count(num12,12),a12]
,'-',machna[count(num13,13),a13],'-',machna[count(num14,14),a14]
,'-',machna[count(num15,15),a15],'-',machna[count(num16,16),a16]
,'-',machna[count(num17,17),a17],'-',machna[count(num18,18),a18]
,'-',machna[count(num19,19),a19],'-',machna[count(num20,20),a20]);
cost :=oprcost[count(num1,1),a1]+oprcost[count(num2,2),a2]
    +oprcost[count(num3,3),a3]+oprcost[count(num4,4),a4]
    +oprcost[count(num5,5),a5]+oprcost[count(num6,6),a6]
    +oprcost[count(num7,7),a7]+oprcost[count(num8,8),a8]
    +oprcost[count(num9,9),a9]+oprcost[count(num10,10),a10]
    +oprcost[count(num11,11),a11]+oprcost[count(num12,12),a12]
    +oprcost[count(num13,13),a13]+oprcost[count(num14,14),a14]
    +oprcost[count(num15,15),a15]+oprcost[count(num16,16),a16]
    +oprcost[count(num17,17),a17]+oprcost[count(num18,18),a18]
    +oprcost[count(num19,19),a19]+oprcost[count(num20,20),a20]
    +prodcost[count(num1,1),a1]+prodcost[count(num2,2),a2]
    +prodcost[count(num3,3),a3]+prodcost[count(num4,4),a4]
    +prodcost[count(num5,5),a5]+prodcost[count(num6,6),a6]
    +prodcost[count(num7,7),a7]+prodcost[count(num8,8),a8]
    +prodcost[count(num9,9),a9]+prodcost[count(num10,10),a10]
    +prodcost[count(num11,11),a11]+prodcost[count(num12,12),a12]
    +prodcost[count(num13,13),a13]+prodcost[count(num14,14),a14]
    +prodcost[count(num15,15),a15]+prodcost[count(num16,16),a16]
    +prodcost[count(num17,17),a17]+prodcost[count(num18,18),a18]
    +prodcost[count(num19,19),a19]+prodcost[count(num20,20),a20];
{ writeln(cost);}
  sum:=sum+cost;
end; end;{if}

Procedure operation;
```

```
begin
 with planrec do
 with oprec do
begin
 case opnum[cnt,cnt1] of
1 : begin
 num1:=0; cunt:=0;
 count(num1,1);
 for a1 := 1 to machnum[count(num1,1)] do
begin
 cunt := cunt+1;
 if cunt=solution[cnt] then
 machine1;
end;  end;
2 : begin
 num1:=0; num2:=0; cunt:=0;
 count(num1,1);
 count(num2,2);
 for a1 := 1 to machnum[count(num1,1)] do
 for a2 := 1 to machnum[count(num2,2)] do
 begin
 cunt := cunt+1;
 if cunt=solution[cnt] then
 machine2;
end;  end;
3 : begin
 num1:=0; num2:=0; num3:=0; cunt:=0;
 count(num1,1);
 count(num2,2);
 count(num3,3);
 for a1 := 1 to machnum[count(num1,1)] do
 for a2 := 1 to machnum[count(num2,2)] do
 for a3 := 1 to machnum[count(num3,3)] do
begin
```

```
  cunt:=cunt+1;
  if cunt=solution[cnt] then
  machine3;
end; end;
4 : begin
  num1:=0; num2:=0; num3:=0; num4:=0; cunt:=0;
  count(num1,1);
  count(num2,2);
  count(num3,3);
  count(num4,4);
  for a1 := 1 to machnum[count(num1,1)] do
  for a2 := 1 to machnum[count(num2,2)] do
  for a3 := 1 to machnum[count(num3,3)] do
  for a4 := 1 to machnum[count(num4,4)] do
begin
  cunt := cunt+1;
  if cunt=solution[cnt] then
  machine4;
end; end;
5 : begin
  num1:=0; num2:=0; num3:=0; num4:=0; num5:=0; cunt:=0;
  count(num1,1);
  count(num2,2);
  count(num3,3);
  count(num4,4);
  count(num5,5);
  for a1 := 1 to machnum[count(num1,1)] do
  for a2 := 1 to machnum[count(num2,2)] do
  for a3 := 1 to machnum[count(num3,3)] do
  for a4 := 1 to machnum[count(num4,4)] do
  for a5 := 1 to machnum[count(num5,5)] do
begin
  cunt := cunt+1;
  if cunt=solution[cnt] then
```

```
  machine5;
end; end;
6 : begin
  num1:=0; num2:=0; num3:=0; num4:=0; num5:=0; num6:=0; cunt:=0;
  count(num1,1);
  count(num2,2);
  count(num3,3);
  count(num4,4);
  count(num5,5);
  count(num6,6);
  for a1 := 1 to machnum[count(num1,1)] do
  for a2 := 1 to machnum[count(num2,2)] do
  for a3 := 1 to machnum[count(num3,3)] do
  for a4 := 1 to machnum[count(num4,4)] do
  for a5 := 1 to machnum[count(num5,5)] do
  for a6 := 1 to machnum[count(num6,6)] do
begin
  cunt := cunt+1;
  if cunt=solution[cnt] then
  machine6;
end; end;
7 : begin
  num1:=0; num2:=0; num3:=0; num4:=0; num5:=0; num6:=0; num7:=0; cunt:=0;
  count(num1,1);
  count(num2,2);
  count(num3,3);
  count(num4,4);
  count(num5,5);
  count(num6,6);
  count(num7,7);
  for a1 := 1 to machnum[count(num1,1)] do
  for a2 := 1 to machnum[count(num2,2)] do
  for a3 := 1 to machnum[count(num3,3)] do
  for a4 := 1 to machnum[count(num4,4)] do
```

```
  for a5 := 1 to machnum[count(num5,5)] do
  for a6 := 1 to machnum[count(num6,6)] do
  for a7 := 1 to machnum[count(num7,7)] do
begin
  cunt := cunt+1;
  if cunt=solution[cnt] then
  machine7;
end; end;
8 : begin
  num1:=0; num2:=0; num3:=0; num4:=0; num5:=0; num6:=0; num7:=0; num8:=0; cunt:=0;
  count(num1,1);
  count(num2,2);
  count(num3,3);
  count(num4,4);
  count(num5,5);
  count(num6,6);
  count(num7,7);
  count(num8,8);
  for a1 := 1 to machnum[count(num1,1)] do
  for a2 := 1 to machnum[count(num2,2)] do
  for a3 := 1 to machnum[count(num3,3)] do
  for a4 := 1 to machnum[count(num4,4)] do
  for a5 := 1 to machnum[count(num5,5)] do
  for a6 := 1 to machnum[count(num6,6)] do
  for a7 := 1 to machnum[count(num7,7)] do
  for a8 := 1 to machnum[count(num8,8)] do
begin
  cunt := cunt+1;
  if cunt=solution[cnt] then
  machine8;
end; end;
9 : begin
  num1:=0; num2:=0; num3:=0; num4:=0; num5:=0; num6:=0; num7:=0; num8:=0; num9:=0;
  cunt:=0;
```

```
 count(num1,1);
 count(num2,2);
 count(num3,3);
 count(num4,4);
 count(num5,5);
 count(num6,6);
 count(num7,7);
 count(num8,8);
 count(num9,9);
 for a1 := 1 to machnum[count(num1,1)] do
 for a2 := 1 to machnum[count(num2,2)] do
 for a3 := 1 to machnum[count(num3,3)] do
 for a4 := 1 to machnum[count(num4,4)] do
 for a5 := 1 to machnum[count(num5,5)] do
 for a6 := 1 to machnum[count(num6,6)] do
 for a7 := 1 to machnum[count(num7,7)] do
 for a8 := 1 to machnum[count(num8,8)] do
 for a9 := 1 to machnum[count(num9,9)] do
begin
 cunt := cunt+1;
 if cunt=solution[cnt] then
 machine9;
end; end;
10 :  begin
 num1:=0; num2:=0; num3:=0; num4:=0; num5:=0; num6:=0; num7:=0; num8:=0; num9:=0;
num10:=0;
 cunt:=0;
 count(num1,1);
 count(num2,2);
 count(num3,3);
 count(num4,4);
 count(num5,5);
 count(num6,6);
 count(num7,7);
```

```
count(num8,8);
count(num9,9);
count(num10,10);
for a1 := 1 to machnum[count(num1,1)] do
for a2 := 1 to machnum[count(num2,2)] do
for a3 := 1 to machnum[count(num3,3)] do
for a4 := 1 to machnum[count(num4,4)] do
for a5 := 1 to machnum[count(num5,5)] do
for a6 := 1 to machnum[count(num6,6)] do
for a7 := 1 to machnum[count(num7,7)] do
for a8 := 1 to machnum[count(num8,8)] do
for a9 := 1 to machnum[count(num9,9)] do
for a10 := 1 to machnum[count(num10,10)] do
begin
cunt := cunt+1;
if cunt=solution[cnt] then
machine10;
end; end;
11 : begin
num1:=0; num2:=0; num3:=0; num4:=0; num5:=0; num6:=0; num7:=0; num8:=0; num9:=0;
num10:=0;
num11:=0; cunt:=0;
count(num1,1);
count(num2,2);
count(num3,3);
count(num4,4);
count(num5,5);
count(num6,6);
count(num7,7);
count(num8,8);
count(num9,9);
count(num10,10);
count(num11,11);
for a1 := 1 to machnum[count(num1,1)] do
```

```
 for a2 := 1 to machnum[count(num2,2)] do
 for a3 := 1 to machnum[count(num3,3)] do
 for a4 := 1 to machnum[count(num4,4)] do
 for a5 := 1 to machnum[count(num5,5)] do
 for a6 := 1 to machnum[count(num6,6)] do
 for a7 := 1 to machnum[count(num7,7)] do
 for a8 := 1 to machnum[count(num8,8)] do
 for a9 := 1 to machnum[count(num9,9)] do
 for a10 := 1 to machnum[count(num10,10)] do
 for a11 := 1 to machnum[count(num11,11)] do
begin
 cunt := cunt+1;
 if cunt=solution[cnt] then
 machine11;
end; end;
12 : begin
 num1:=0; num2:=0; num3:=0; num4:=0; num5:=0; num6:=0; num7:=0; num8:=0; num9:=0;
num10:=0;
 num11:=0; num12:=0; cunt:=0;
 count(num1,1);
 count(num2,2);
 count(num3,3);
 count(num4,4);
 count(num5,5);
 count(num6,6);
 count(num7,7);
 count(num8,8);
 count(num9,9);
 count(num10,10);
 count(num11,11);
 count(num12,12);
 for a1 := 1 to machnum[count(num1,1)] do
 for a2 := 1 to machnum[count(num2,2)] do
 for a3 := 1 to machnum[count(num3,3)] do
```

```
 for a4 := 1 to machnum[count(num4,4)] do
 for a5 := 1 to machnum[count(num5,5)] do
 for a6 := 1 to machnum[count(num6,6)] do
 for a7 := 1 to machnum[count(num7,7)] do
 for a8 := 1 to machnum[count(num8,8)] do
 for a9 := 1 to machnum[count(num9,9)] do
 for a10 := 1 to machnum[count(num10,10)] do
 for a11 := 1 to machnum[count(num11,11)] do
 for a12 := 1 to machnum[count(num12,12)] do
begin
 cunt := cunt+1;
 if cunt=solution[cnt] then
 machine12;
end; end;
13 : begin
 num1:=0; num2:=0; num3:=0; num4:=0; num5:=0; num6:=0; num7:=0; num8:=0; num9:=0;
num10:=0;
 num11:=0; num12:=0; num13:=0; cunt:=0;
 count(num1,1);
 count(num2,2);
 count(num3,3);
 count(num4,4);
 count(num5,5);
 count(num6,6);
 count(num7,7);
 count(num8,8);
 count(num9,9);
 count(num10,10);
 count(num11,11);
 count(num12,12);
 count(num13,13);
 for a1 := 1 to machnum[count(num1,1)] do
 for a2 := 1 to machnum[count(num2,2)] do
 for a3 := 1 to machnum[count(num3,3)] do
```

```
for a4 := 1 to machnum[count(num4,4)] do
for a5 := 1 to machnum[count(num5,5)] do
for a6 := 1 to machnum[count(num6,6)] do
for a7 := 1 to machnum[count(num7,7)] do
for a8 := 1 to machnum[count(num8,8)] do
for a9 := 1 to machnum[count(num9,9)] do
for a10 := 1 to machnum[count(num10,10)] do
for a11 := 1 to machnum[count(num11,11)] do
for a12 := 1 to machnum[count(num12,12)] do
for a13 := 1 to machnum[count(num13,13)] do
begin
cunt := cunt+1;
if cunt=solution[cnt] then
machine13;
end; end;
14 : begin
num1:=0; num2:=0; num3:=0; num4:=0; num5:=0; num6:=0; num7:=0; num8:=0; num9:=0;
num10:=0;
num11:=0; num12:=0; num13:=0; num14:=0; cunt:=0;
count(num1,1);
count(num2,2);
count(num3,3);
count(num4,4);
count(num5,5);
count(num6,6);
count(num7,7);
count(num8,8);
count(num9,9);
count(num10,10);
count(num11,11);
count(num12,12);
count(num13,13);
count(num14,14);
for a1 := 1 to machnum[count(num1,1)] do
```

```
 for a2 := 1 to machnum[count(num2,2)] do
 for a3 := 1 to machnum[count(num3,3)] do
 for a4 := 1 to machnum[count(num4,4)] do
 for a5 := 1 to machnum[count(num5,5)] do
 for a6 := 1 to machnum[count(num6,6)] do
 for a7 := 1 to machnum[count(num7,7)] do
 for a8 := 1 to machnum[count(num6,8)] do
 for a9 := 1 to machnum[count(num9,9)] do
 for a10 := 1 to machnum[count(num10,10)] do
 for a11 := 1 to machnum[count(num11,11)] do
 for a12 := 1 to machnum[count(num12,12)] do
 for a13 := 1 to machnum[count(num13,13)] do
 for a14 := 1 to machnum[count(num14,14)] do
begin
 cunt := cunt+1;
 if cunt=solution[cnt] then
 machine14;
end; end;
15 : begin
 num1:=0; num2:=0; num3:=0; num4:=0; num5:=0; num6:=0; num7:=0; num8:=0; num9:=0;
num10:=0;
 num11:=0; num12:=0; num13:=0; num14:=0; num15:=0; cunt:=0;
 count(num1,1);
 count(num2,2);
 count(num3,3);
 count(num4,4);
 count(num5,5);
 count(num6,6);
 count(num7,7);
 count(num6,8);
 count(num9,9);
 count(num10,10);
 count(num11,11);
 count(num12,12);
```

```
count(num13,13);
count(num14,14);
count(num15,15);
for a1 := 1 to machnum[count(num1,1)] do
for a2 := 1 to machnum[count(num2,2)] do
for a3 := 1 to machnum[count(num3,3)] do
for a4 := 1 to machnum[count(num4,4)] do
for a5 := 1 to machnum[count(num5,5)] do
for a6 := 1 to machnum[count(num6,6)] do
for a7 := 1 to machnum[count(num7,7)] do
for a8 := 1 to machnum[count(num8,8)] do
for a9 := 1 to machnum[count(num9,9)] do
for a10 := 1 to machnum[count(num10,10)] do
for a11 := 1 to machnum[count(num11,11)] do
for a12 := 1 to machnum[count(num12,12)] do
for a13 := 1 to machnum[count(num13,13)] do
for a14 := 1 to machnum[count(num14,14)] do
for a15 := 1 to machnum[count(num15,15)] do
begin
 cunt := cunt+1;
 if cunt=solution[cnt] then
 machine15;
end; end;
16 : begin
 num1:=0; num2:=0; num3:=0; num4:=0; num5:=0; num6:=0; num7:=0; num8:=0; num9:=0;
num10:=0;
 num11:=0; num12:=0; num13:=0; num14:=0; num15:=0; num16:=0; cunt:=0;
 count(num1,1);
 count(num2,2);
 count(num3,3);
 count(num4,4);
 count(num5,5);
 count(num6,6);
 count(num7,7);
```

```
 count(num8,8);
 count(num9,9);
 count(num10,10);
 count(num11,11);
 count(num12,12);
 count(num13,13);
 count(num14,14);
 count(num15,15);
 count(num16,16);
 for a1 := 1 to machnum[count(num1,1)] do
 for a2 := 1 to machnum[count(num2,2)] do
 for a3 := 1 to machnum[count(num3,3)] do
 for a4 := 1 to machnum[count(num4,4)] do
 for a5 := 1 to machnum[count(num5,5)] do
 for a6 := 1 to machnum[count(num6,6)] do
 for a7 := 1 to machnum[count(num7,7)] do
 for a8 := 1 to machnum[count(num8,8)] do
 for a9 := 1 to machnum[count(num9,9)] do
 for a10 := 1 to machnum[count(num10,10)] do
 for a11 := 1 to machnum[count(num11,11)] do
 for a12 := 1 to machnum[count(num12,12)] do
 for a13 := 1 to machnum[count(num13,13)] do
 for a14 := 1 to machnum[count(num14,14)] do
 for a15 := 1 to machnum[count(num15,15)] do
 for a16 := 1 to machnum[count(num16,16)] do
begin
 cunt := cunt+1;
 if cunt=solution[cnt] then
 machine16;
end; end;
17 : begin
 num1:=0; num2:=0; num3:=0; num4:=0; num5:=0; num6:=0; num7:=0; num8:=0; num9:=0;
num10:=0;
 num11:=0; num12:=0; num13:=0; num14:=0; num15:=0; num16:=0; num17:=0;
```

```
cunt:=0;
count(num1,1);
count(num2,2);
count(num3,3);
count(num4,4);
count(num5,5);
count(num6,6);
count(num7,7);
count(num8,8);
count(num9,9);
count(num10,10);
count(num11,11);
count(num12,12);
count(num13,13);
count(num14,14);
count(num15,15);
count(num16,16);
count(num17,17);
for a1 := 1 to machnum[count(num1,1)] do
for a2 := 1 to machnum[count(num2,2)] do
for a3 := 1 to machnum[count(num3,3)] do
for a4 := 1 to machnum[count(num4,4)] do
for a5 := 1 to machnum[count(num5,5)] do
for a6 := 1 to machnum[count(num6,6)] do
for a7 := 1 to machnum[count(num7,7)] do
for a8 := 1 to machnum[count(num8,8)] do
for a9 := 1 to machnum[count(num9,9)] do
for a10 := 1 to machnum[count(num10,10)] do
for a11 := 1 to machnum[count(num11,11)] do
for a12 := 1 to machnum[count(num12,12)] do
for a13 := 1 to machnum[count(num13,13)] do
for a14 := 1 to machnum[count(num14,14)] do
for a15 := 1 to machnum[count(num15,15)] do
for a16 := 1 to machnum[count(num16,16)] do
```

```
 for a17 := 1 to machnum[count(num17,17)] do
begin
 cunt := cunt+1;
 if cunt=solution[cnt] then
 machine17;
end; end;
18 : begin
 num1:=0; num2:=0; num3:=0; num4:=0; num5:=0; num6:=0; num7:=0; num8:=0; num9:=0;
num10:=0;
 num11:=0; num12:=0; num13:=0; num14:=0; num15:=0; num16:=0; num17:=0; num18:=0;
 cunt:=0;
 count(num1,1);
 count(num2,2);
 count(num3,3);
 count(num4,4);
 count(num5,5);
 count(num6,6);
 count(num7,7);
 count(num8,8);
 count(num9,9);
 count(num10,10);
 count(num11,11);
 count(num12,12);
 count(num13,13);
 count(num14,14);
 count(num15,15);
 count(num16,16);
 count(num17,17);
 count(num18,18);
 for a1 := 1 to machnum[count(num1,1)] do
 for a2 := 1 to machnum[count(num2,2)] do
 for a3 := 1 to machnum[count(num3,3)] do
 for a4 := 1 to machnum[count(num4,4)] do
 for a5 := 1 to machnum[count(num5,5)] do
```

```
for a6 := 1 to machnum[count(num6,6)] do
for a7 := 1 to machnum[count(num7,7)] do
for a8 := 1 to machnum[count(num8,8)] do
for a9 := 1 to machnum[count(num9,9)] do
for a10 := 1 to machnum[count(num10,10)] do
for a11 := 1 to machnum[count(num11,11)] do
for a12 := 1 to machnum[count(num12,12)] do
for a13 := 1 to machnum[count(num13,13)] do
for a14 := 1 to machnum[count(num14,14)] do
for a15 := 1 to machnum[count(num15,15)] do
for a16 := 1 to machnum[count(num16,16)] do
for a17 := 1 to machnum[count(num17,17)] do
for a18 := 1 to machnum[count(num18,18)] do
begin
 cunt := cunt+1;
 if cunt=solution[cnt] then
 machine18;
end; end;
19 : begin
 num1:=0; num2:=0; num3:=0; num4:=0; num5:=0; num6:=0; num7:=0; num8:=0; num9:=0;
num10:=0;
 num11:=0; num12:=0; num13:=0; num14:=0; num15:=0; num16:=0; num17:=0; num18:=0;
num19:=0;
 cunt:=0;
 count(num1,1);
 count(num2,2);
 count(num3,3);
 count(num4,4);
 count(num5,5);
 count(num6,6);
 count(num7,7);
 count(num8,8);
 count(num9,9);
 count(num10,10);
```

```
  count(num11,11);
  count(num12,12);
  count(num13,13);
  count(num14,14);
  count(num15,15);
  count(num16,16);
  count(num17,17);
  count(num18,18);
  count(num19,19);
  for a1 := 1 to machnum[count(num1,1)] do
  for a2 := 1 to machnum[count(num2,2)] do
  for a3 := 1 to machnum[count(num3,3)] do
  for a4 := 1 to machnum[count(num4,4)] do
  for a5 := 1 to machnum[count(num5,5)] do
  for a6 := 1 to machnum[count(num6,6)] do
  for a7 := 1 to machnum[count(num7,7)] do
  for a8 := 1 to machnum[count(num8,8)] do
  for a9 := 1 to machnum[count(num9,9)] do
  for a10 := 1 to machnum[count(num10,10)] do
  for a11 := 1 to machnum[count(num11,11)] do
  for a12 := 1 to machnum[count(num12,12)] do
  for a13 := 1 to machnum[count(num13,13)] do
  for a14 := 1 to machnum[count(num14,14)] do
  for a15 := 1 to machnum[count(num15,15)] do
  for a16 := 1 to machnum[count(num16,16)] do
  for a17 := 1 to machnum[count(num17,17)] do
  for a18 := 1 to machnum[count(num18,18)] do
  for a19 := 1 to machnum[count(num19,19)] do
 begin
  cunt := cunt+1;
  if cunt=solution[cnt] then
  machine19;
end; end;
20 : begin
```

```
 num1:=0; num2:=0; num3:=0; num4:=0; num5:=0; num6:=0; num7:=0; num8:=0; num9:=0;
num10:=0;
 num11:=0; num12:=0; num13:=0; num14:=0; num15:=0; num16:=0; num17:=0; num18:=0;
num19:=0; num20:=0;
 cunt:=0;
 count(num1,1);
 count(num2,2);
 count(num3,3);
 count(num4,4);
 count(num5,5);
 count(num6,6);
 count(num7,7);
 count(num8,8);
 count(num9,9);
 count(num10,10);
 count(num11,11);
 count(num12,12);
 count(num13,13);
 count(num14,14);
 count(num15,15);
 count(num16,16);
 count(num17,17);
 count(num18,18);
 count(num19,19);
 count(num20,20);
 for a1 := 1 to machnum[count(num1,1)] do
 for a2 := 1 to machnum[count(num2,2)] do
 for a3 := 1 to machnum[count(num3,3)] do
 for a4 := 1 to machnum[count(num4,4)] do
 for a5 := 1 to machnum[count(num5,5)] do
 for a6 := 1 to machnum[count(num6,6)] do
 for a7 := 1 to machnum[count(num7,7)] do
 for a8 := 1 to machnum[count(num8,8)] do
 for a9 := 1 to machnum[count(num9,9)] do
```

```
 for a10 := 1 to machnum[count(num10,10)] do
 for a11 := 1 to machnum[count(num11,11)] do
 for a12 := 1 to machnum[count(num12,12)] do
 for a13 := 1 to machnum[count(num13,13)] do
 for a14 := 1 to machnum[count(num14,14)] do
 for a15 := 1 to machnum[count(num15,15)] do
 for a16 := 1 to machnum[count(num16,16)] do
 for a17 := 1 to machnum[count(num17,17)] do
 for a18 := 1 to machnum[count(num18,18)] do
 for a19 := 1 to machnum[count(num19,19)] do
 for a20 := 1 to machnum[count(num20,20)] do
begin
 cunt:=cunt+1;
 if cunt=solution[cnt] then
 machine20;
end;{for}
end;{case20}
end;{case}
end;
end;

Procedure Tabupattern(var fi : fitype; var fi1 : fitype1);
begin
 with planrec do
 with oprrec do
begin
 sum:=0;
 for cnt := 1 to partnum do
begin
 repeat
 randomize;
 l := random(method[cnt]);
 until l > 0;
 sol[cnt] := l;
```

```
  cnt1:=1;
  if l <= tabu[cnt,cnt1] then
begin
  solution[cnt]:=l;
  operation;
end{l<tabu}
  else if l > tabu[cnt,cnt1] then
begin
  m:=l-tabu[cnt,cnt1];
  cnt1:=cnt1+1;
  if m <= tabu[cnt,cnt1] then
begin
  l:=m;
  solution[cnt]:=l;
  operation;
end{m<tabu}
  else if m > tabu[cnt,cnt1] then
begin
  n:=m-tabu[cnt,cnt1];
  cnt1:=cnt1+1;
  l:=n;
  solution[cnt]:=l;
  operation;
end;{solution_n}
end;{l>tabu}
end;{for_cnt}
end;{with}
end;{main}

Procedure Tabupattern2(var fi : fitype; var fi1 : fitype1);
begin
  with planrec do
  with oprrec do
begin
```

```
 sum:=0;
 for cnt := 1 to partnum do
begin
 sol[cnt] := solution[cnt];
 l := solution[cnt];
 cnt1:=1;
 if l <= tabu[cnt,cnt1] then
begin
 solution[cnt]:=l;
 operation;
end{l<tabu}
 else if l > tabu[cnt,cnt1] then
begin
 m:=l-tabu[cnt,cnt1];
 cnt1:=cnt1+1;
 if m <= tabu[cnt,cnt1] then
begin
 l:=m;
 solution[cnt]:=l;
 operation;
end{m<tabu}
 else if m > tabu[cnt,cnt1] then
begin
 n:=m-tabu[cnt,cnt1];
 cnt1:=cnt1+1;
 l:=n;
 solution[cnt]:=l;
 operation;
end;{solution_n}
end;{l>tabu}
end;{for_cnt}
end;{with}
end;{main}
```

```
Function swap3(d,g : integer) : longint;
begin
  solve[ran2] := solution[ran2];
  solution[ran2] := solution[ran1];
  solution[ran1] := solve[ran2];
  writeln('Iteration ',num);
  tabupattern2(planfi,oprfi);
  for l := 1 to part do
begin
  solution[l] := sol[l];
end;
  if aspiration[d] > sum then
begin
  more := more+1;
  if more = g then more := 1;
  tabupos[d] := 0;
  tabulist[d] := 0;
  aspiration[d] := 9999999;
  tabulist[more] := solution[ran2];
  tabupos[more] := ran2;
  aspiration[more] := sum;
  if sum < minimum then minimum := sum;
end else
begin
end;  end;

Procedure swap(var fi : fitype; var fi1 : fitype1);
var cnt2,cnt3,next,mach1,size     : integer;
  zum,times,newtimes,minimum1,maximum   : longint;
    ch,key : char;
begin
  clrscr;
  writeln('Neighborhood List :Swap pairwise interchange');
  writeln;
```

```
{ until cnt=partnum;}
end;{for_cnt}
 readln(key);
 if key = 'O' then
begin
 writeln('Iteration 0');
 tabupattern(planfi,oprfi);
 minimum := sum;
 write('Sequence : ');
 for I := 1 to partnum do
begin
 write(sol[I],',');
end;
 writeln;
 writeln('Makespan : ',sum,'  Best : ',minimum);
end else
begin
 for I := 1 to partnum do
begin
 write('Enter seed : '); readln(sol[I]);
end;
 write('Enter minimum : '); readln(minimum); end;
 {SWAP PAIRWISE}
 num:=1; more:=0;
 for I := 1 to partnum do
begin
 solution[I] := sol[I];
end;
repeat
 repeat
 randomize;
 ran1 := random(partnum);
 until ran1 >0;
 repeat
```

```
  randomize;
  ran2 := random(partnum);
  until (ran2>0) and (ran2<>ran1) and (solution[ran1]<=method[ran2])
  and (solution[ran2]<=method[ran1]);
  case size of
  3 : begin
   if (solution[ran1] = tabulist[1]) and (ran2 = tabupos[1]) then
begin
  swap3(1,4);
end else
  if (solution[ran1] = tabulist[2]) and (ran2 = tabupos[2]) then
begin
  swap3(2,4);
end else
  if (solution[ran1] = tabulist[3]) and (ran2 = tabupos[3]) then
begin
  swap3(3,4);
end else
begin
  solve[ran2] := solution[ran2];
  solution[ran2] := solution[ran1];
  solution[ran1] := solve[ran2];
  writeln('Iteration ',num);
  tabupattern2(planfi,oprfi);
  for l := 1 to partnum do
begin
  solution[l] := sol[l];
end;
  more := more+1;
  if more = 4 then more := 1;
  tabulist[more] := solution[ran2];
  tabupos[more] := ran2;
  aspiration[more] := sum;
  if sum < minimum then minimum := sum;
```

```
    end;
     minimum1 := minimum-minimum1;
     if minimum1 = 0 then begin minimum1 := minimum; zum := zum+1; end
     else begin minimum1 := minimum; zum:=0; end;
     write('Sequence : ');
     for i := 1 to partnum do
    begin
     write(sol[i],',');
    end;
     writeln;
     writeln('T((i),j)  Makespan   Tabulist');
     write(sol[ran2]:2,ran2:3,sum:9,'':6);
     for m := 1 to 3 do
   begin
     write(tabulist[m],',');
   end;
     writeln;
     writeln('Best : ',minimum);
  end;
    5 : begin
    if (solution[ran1] = tabulist[1]) and (ran2 = tabupos[1]) then
  begin
    swap3(1,6);
  end else
    if (solution[ran1] = tabulist[2]) and (ran2 = tabupos[2]) then
  begin
    swap3(2,6);
  end else
    if (solution[ran1] = tabulist[3]) and (ran2 = tabupos[3]) then
  begin
    swap3(3,6);
  end else
    if (solution[ran1] = tabulist[4]) and (ran2 = tabupos[4]) then
  begin
```

```
      swap3(4,6);
    end else
     if (solution[ran1] = tabulist[5]) and (ran2 = tabupos[5]) then
    begin
     swap3(5,6);
    end else
    begin
     solve[ran2] := solution[ran2];
     solution[ran2] := solution[ran1];
     solution[ran1] := solve[ran2];
     writeln('Iteration ',num);
     tabupattern2(planfi,oprfi);
     for I := 1 to partnum do
   begin
    solution[I] := sol[I];
   end;
    more := more+1;
    if more = 6 then more := 1;
    tabulist[more] := solution[ran2];
    tabupos[more] := ran2;
    aspiration[more] := sum;
    if sum < minimum then minimum := sum;
  end;
    minimum1 := minimum-minimum1;
    if minimum1 = 0 then begin minimum1 := minimum; zum := zum+1; end
    else begin minimum1 := minimum; zum:=0; end;
    write('Sequence : ');
    for I := 1 to partnum do
 begin
  write(sol[I],',');
 end;
  writeln;
  writeln('T((i),j)  Makespan   Tabulist');
  write(sol[ran2]:2,ran2:3,sum:9,'':6);
```

```
    for m := 1 to 5 do
   begin
    write(tabulist[m],',');
   end;
    writeln;
    writeln('Best : ',minimum);
   end;
    7 : begin
    if (solution[ran1] = tabulist[1]) and (ran2 = tabupos[1]) then
   begin
    swap3(1,8);
   end else
    if (solution[ran1] = tabulist[2]) and (ran2 = tabupos[2]) then
   begin
    swap3(2,8);
   end else
    if (solution[ran1] = tabulist[3]) and (ran2 = tabupos[3]) then
   begin
    swap3(3,8);
   end else
    if (solution[ran1] = tabulist[4]) and (ran2 = tabupos[4]) then
   begin
    swap3(4,8);
   end else
    if (solution[ran1] = tabulist[5]) and (ran2 = tabupos[5]) then
   begin
    swap3(5,8);
   end else
    if (solution[ran1] = tabulist[6]) and (ran2 = tabupos[6]) then
   begin
    swap3(6,6);
   end else
    if (solution[ran1] = tabulist[7]) and (ran2 = tabupos[7]) then
   begin
```

```
      swap3(7,8);
    end else
    begin
      solve[ran2] := solution[ran2];
      solution[ran2] := solution[ran1];
      solution[ran1] := solve[ran2];
      writeln('Iteration ',num);
      tabupattern2(planfi,oprfi);
      for l := 1 to partnum do
    begin
      solution[l] := sol[l];
    end;
      more := more+1;
      if more = 8 then more := 1;
      tabulist[more] := solution[ran2];
      tabupos[more] := ran2;
      aspiration[more] := sum;
      if sum < minimum then minimum := sum;
   end;
     minimum1 := minimum-minimum1;
     if minimum1 = 0 then begin minimum1 := minimum; zum := zum+1; end
     else begin minimum1 := minimum; zum:=0; end;
     write('Sequence : ');
     for l := 1 to partnum do
  begin
     write(sol[i],',');
  end;
     writeln;
     writeln('T((i),j)  Makespan   Tabulist');
     write(sol[ran2]:2,ran2:3,sum:9,'':6);
     for m := 1 to 7 do
 begin
     write(tabulist[m],',');
 end;
```

```
    writeln;
    writeln('Best : ',minimum);
   end  else exit; end;
    {if num mod 20 = 0 then ch := readkey;}
    num := num+1;
    until (num=1001) or (zum=10);
  end;{with_opr}
  end;{main}

  Procedure Tabupattern3(var fi : fitype; var fi1 : fitype1);
  begin
   with planrec do
   with oprrec do
  begin
   sum:=0;
   for cnt := 1 to partnum do
  begin
   repeat
   randomize;
   I := random(method[cnt]);
   until I > 0;
   cnt1:=1;
   sol[cnt] := I;
   if I <= tabu[cnt,cnt1] then
 begin
   solution[cnt]:=I;
   operation;
 end{I<tabu}
   else if I > tabu[cnt,cnt1] then
begin
  m:=I-tabu[cnt,cnt1];
  cnt1:=cnt1+1;
  if m <= tabu[cnt,cnt1] then
begin
```

```
    l:=m;
    solution[cnt]:=l;
    operation;
   end{m<tabu}
    else if m > tabu[cnt,cnt1] then
   begin
    n:=m-tabu[cnt,cnt1];
    cnt1:=cnt1+1;
    l:=n;
    solution[cnt]:=l;
    operation;
   end;{solution_n}
   end;{l>tabu}
   end;{for_cnt}
   end;{with}
   end;{main}

   Procedure Tabupattern4(var fi : fitype; var fi1 : fitype1);
   begin
    with planrec do
    with oprrec do
   begin
    sum:=0;
    for cnt := 1 to partnum do
   begin
    sol[cnt] := solution[cnt];
    l := solution[cnt];
    cnt1:=1;
    if l <= tabu[cnt,cnt1] then
   begin
    solution[cnt]:=l;
    operation;
   end{l<tabu}
    else if l > tabu[cnt,cnt1] then
```

```
begin
 m:=l-tabu[cnt,cnt1];
 cnt1:=cnt1+1;
 if m <= tabu[cnt,cnt1] then
begin
 l:=m;
 solution[cnt]:=l;
 operation;
end{m<tabu}
 else if m > tabu[cnt,cnt1] then
begin
 n:=m-tabu[cnt,cnt1];
 cnt1:=cnt1+1;
 l:=n;
 solution[cnt]:=l;
 operation;
end;{solution_n}
end;{l>tabu}
end;{for_cnt}
end;{with}
end;{main}

Function insert3(e,f : integer) : longint;
var j   : integer;
   tem : longint;
begin
 with planrec do
 with oprrec do
begin
 for m := 1 to partnum do
begin
 tempo[m] := solution[m];
end;
 tempo[1] := tempo[ran1];
```

```
  l := ran1;
  l := l-1;
  if l=0 then
 begin
 end  else
 begin
  for j := 1 to l do
 begin
  tempo[j+1] := solution[j];
 end;
 end;{else}
  count1 := 1;
 {  if (count1=ran1) or (count1=ran1-1) or (count1<>ran2) then
 begin  writeln('Iteration ',num);
 end else)
  if (ran1>=3) and (ran2=1) then
 begin
  for m := 1 to partnum do
 begin
  solution[m] := tempo[m];
 end;
  writeln('Iteration ',num);
  tabupattern4(planfi,oprfi);
  for l := 1 to partnum do
 begin
  solution[l] := sol[l];
 end;
  if aspiration[e] > sum then
 begin
  more := more+1;
  if more = f then more := 1;
  tabulist[e] := 0;
  aspiration[e] := 999999;
  tabulist[more] := tabu1;
```

```
 aspiration[more] := sum;
 if sum < minimum then minimum := sum;
end  else
begin  end; end;
 for i := 1 to partnum-1 do
begin
 count1 := count1+1;
 if (count1=ran1) or (count1=ran1-1) or (count1<>ran2) then
begin
 tem := tempo[i+1];
 tempo[i+1] := tempo[i];
 tempo[i] := tem;
end else
begin
 tem := tempo[i+1];
 tempo[i+1] := tempo[i];
 tempo[i] := tem;
 for m := 1 to partnum do
begin
 solution[m] := tempo[m];
end;
 writeln(partnum); readln;
 writeln('Iteration ',num);
 tabupattern4(planfi,oprfi);
 for l := 1 to partnum do
begin
 solution[l] := sol[l];
end;
 if aspiration[e] > sum then
begin
 more := more+1;
 if more = f then more := 1;
 tabulist[e] := 0;
 aspiration[e] := 999999;
```

```
 tabulist[more] := tabu1;
 aspiration[more] := sum;
 if sum < minimum then minimum := sum;
end else begin  end; end; end; end; end;

Function insert4(g,h : integer) : longint;
var j   : integer;
   tem : longint;
begin
 with planrec do
 with oprrec do
begin
 for m := 1 to partnum do
begin
 tempo[m] := solution[m];
end;
 tempo[1] := tempo[ran1];
 l := ran1;
 l := l-1;
 if l=0 then
begin
end  else
begin
 for j := 1 to l do
begin
 tempo[j+1] := solution[j];
end;
end;{else}
 count1 := 1;
{  if (count1=ran1) or (count1=ran1-1) or (count1<>ran2) then
begin
end else}
 if (ran1>=3) and (ran2=1) then
begin
```

```
  for m := 1 to partnum do
 begin
  solution[m] := tempo[m];
 end;
  writeln('Iteration ',num);
  tabupattern4(planfi,oprfi);
  for l := 1 to partnum do
 begin
  solution[l] := sol[l];
 end;
  more := more+1;
  if more = h then more := 1;
  tabulist[more] := tabu1;
  aspiration[more] := sum;
  if sum < minimum then minimum := sum;
end;
  for i := 1 to partnum-1 do
begin
  count1 := count1+1;
  if (count1=ran1) or (count1=ran1-1) or (count1<>ran2) then
begin
  tem := tempo[i+1];
  tempo[i+1] := tempo[i];
  tempo[i] := tem;
end else
begin
  tem := tempo[i+1];
  tempo[i+1] := tempo[i];
  tempo[i] := tem;
  for m := 1 to partnum do
begin
  solution[m] := tempo[m];
end;
  writeln('Iteration ',num);
```

```
 tabupattern4(planfi,oprfi);
 for l := 1 to partnum do
begin
 solution[l] := sol[l];
end;
 more := more+1;
 if more = h then more := 1;
 tabulist[more] := tabu1;
 aspiration[more] := sum;
 if sum < minimum then minimum := sum;
end; end; end; end;

Procedure Insertion(var fi : fitype; var fi1 : fitype1);
var j,cnt2,next,mach1,size1      : integer;
   times,newtimes,tem,maximum   : longint;
   key1 : char;
begin
 clrscr;
 writeln('Neighborhood List :Insertion interchange');
 writeln;
 write('Select Tabu size [3,5,7] ; '); readln(size1);
 prepare(planfi);
 prepare1(oprfi);
 read(fi,planrec);
 read(fi1,oprrec);
 with planrec do
 with oprrec do
begin
 writeln('N: Enter new seed');
 writeln('O: Use old seed');
 sum:=0;  maximum:=0;  zum1:=0;  minimum2:=0;
 for cnt := 1 to partnum do
begin
 newtimes:=0;
```

```
 for cnt1 := 1 to plannum[cnt] do
begin
 times:=1;
 for cnt2 := 1 to opnum[cnt,cnt1] do
begin
 next :=0;
 repeat
 next := next+1;
 findword(opna[cnt,cnt1,cnt2],opname[next]);
 until opna[cnt,cnt1,cnt2]=opname[next];
 mach1 := machnum[next];{Total plan for part,plan}
 times := times*mach1;
end;{for_cnt2}
 tabu[cnt,cnt1] := times;
 newtimes := newtimes+tabu[cnt,cnt1];
end;{for_cnt1}
 method[cnt] := newtimes;
{ repeat}
 if method[cnt]>maximum then maximum:=method[cnt];
{ until cnt=partnum;}
end;{for_cnt}
 readln(key1);
 if key1 = 'O' then
begin
 writeln('Iteration 0');
 tabupattern3(planfi,oprfi);
 minimum := sum;
 write('Sequence : ');
 for i := 1 to partnum do
begin
 write(sol[i],',');
end;
 writeln;
 writeln('Makespan : ',sum,'  Best : ',minimum);
```

```
end else
begin
 for l := 1 to partnum do
begin
 writeln('Enter seed : ');
 readln(sol[l]);
end;
 write('Enter minimum : ');
 readln(minimum);
end;
 {INSERTION}
 num:=1;  more:=0;
 for l := 1 to partnum do
begin
 solution[l] := sol[l];
end;
repeat
 repeat
 randomize;
 ran1 := random(partnum);
 until ran1 >0;
 repeat
 randomize;
 ran2 := random(partnum);
 until (ran2>0) and (ran2<>ran1) and (ran2<>ran1-1) and (solution[ran1]<=method[ran2])
 and (solution[ran2]<=method[ran1]);
 tabu1 := solution[ran1];
 case size1 of
 3 : begin
 if tabulist[1] = solution[ran1] then
begin
 insert3(1,4);
end else
 if tabulist[2] = solution[ran1] then
```

```
begin
 insert3(2,4);
end else
 if tabulist[3] = solution[ran1] then
begin
 insert3(3,4);
end else
begin
 insert4(3,4);
end;
 minimum2 := minimum-minimum2;
 if minimum2=0 then begin minimum2 := minimum;  zum1 := zum1+1; end
 else begin minimum2 := minimum;  zum1:=0; end;
 write('Sequence : ');
 for l := 1 to partnum do
begin
 write(sol[l],',');
end;
 writeln;
 writeln('T(i)    Makespan   Tabulist');
 write(tabu1:2,sum:12,'':6);
 for j := 1 to 3 do
begin
 write(tabulist[j],',');
end;
 writeln;
 writeln('Best : ',minimum);
end;
 5 : begin
 if tabulist[1] = solution[ran1] then
begin
 insert3(1,6);
end else
 if tabulist[2] = solution[ran1] then
```

```
begin
 insert3(2,6);
end else
 if tabulist[3] = solution[ran1] then
begin
 insert3(3,6);
end else
 if tabulist[4] = solution[ran1] then
begin
 insert3(4,6);
end else
 if tabulist[5] = solution[ran1] then
begin
 insert3(5,6);
end else
begin
 insert4(5,6);
end;
 minimum2 := minimum-minimum2;
 if minimum2=0 then begin minimum2 := minimum;  zum1 := zum1+1; end
 else begin minimum2 := minimum;  zum1:=0; end;
 write('Sequence : ');
 for l := 1 to partnum do
begin
 write(sol[l],',');
end;
 writeln;
 writeln('T(i)    Makespan   Tabulist');
 write(tabu1:2,sum:12,':6);
 for j := 1 to 5 do
begin
 write(tabulist[j],',');
end;
 writeln;
```

```
writeln('Best : ',minimum); end;
7 : begin
if tabulist[1] = solution[ran1] then
begin
insert3(1,8);
end else
if tabulist[2] = solution[ran1] then
begin
insert3(2,8);
end else
if tabulist[3] = solution[ran1] then
begin
insert3(3,8);
end else
if tabulist[4] = solution[ran1] then
begin
insert3(4,8);
end else
if tabulist[5] = solution[ran1] then
begin
insert3(5,8);
end else
if tabulist[6] = solution[ran1] then
begin
insert3(6,8);
end else
if tabulist[7] = solution[ran1] then
begin
insert3(7,8);
end else
begin
insert4(7,8);
end;
minimum2 := minimum-minimum2;
```

```
 if minimum2=0 then begin minimum2 := minimum;  zum1 := zum1+1; end
 else begin minimum2 := minimum;  zum1:=0; end;
 write('Sequence : ');
 for l := 1 to partnum do
begin
 write(sol[l],',');
end;
 writeln;
 writeln('T(i)    Makespan   Tabulist');
 write(tabu1:2,sum:12,'':6);
 for j := 1 to 7 do
begin
 write(tabulist[j],',');
end;
 writeln;
 writeln('Best : ',minimum);
 end; else exit end;
 num := num+1; readln;
 until (num = 1001) or (zum1 = 10);
end;{with_opr}
end;{main}

Procedure Display(var fi : fitype);
var num1,num2,num3 : integer;
begin
 prepare(planfi);
 read(fi,planrec);
 with planrec do
begin
 clrscr;
 writeln('Part.   Plan     Operation');
 for num1 := 1 to partnum do
 for num2 := 1 to plannum[num1] do
 for num3 := 1 to opmum[num1,num2] do
```

```
 writeln(partna[num1],'      ',planna[num1,num2],'      ',opma[num1,num2,num3]);
end;
end;

Procedure Display1(var fi1 : fitype1);
var num4,num5 : integer;
begin
 prepare1(oprfi);
 read(fi1,oprrec);
 with oprrec do
begin
 clrscr;
 writeln('Operation   Machine   Opr Cost   Prod Cost ');
 for num4 := 1 to oprnumm do
 for num5 := 1 to machnum[num4] do
 writeln('   ',oprname[num4],'      ',machna[num4,num5],'      ',oprcost[num4,num5]
 ,'      ',prodcost[num4,num5]);
end;
end;

Procedure Tabusearch;
var ch2 : char;
begin
 clrscr;
 writeln;  writeln;
 writeln('Type of Tabu Search ');
 writeln('S : Swap Pairwise ');
 writeln('I : Insertion ');
 writeln('Q : Quit ');
 writeln;
 write('Select type of Search : ');
 repeat ch2 := upcase(readkey) until ch2 in ['S','I','Q'];
    case ch2 of
     'S' : Swap(planfi,oprfi);
```

```
    'I' : Insertion(planfi,oprfi);
    'Q' : begin clrscr; Exit; end;
   end;
end;

Procedure getoption(var ch1 : char);
begin
 writeln;  writeln;
 writeln('E : Enter Part ');
 writeln('O : Enter Operation ');
 writeln('D : Display Part ');
 writeln('K : Display Operation ');
 writeln('T : Tabu Search ');
 writeln('Q : Quit ');
 writeln; write('Please Select  ');
 repeat ch1 := upcase(readkey) until ch1 in ['E','D','K','O','T','Q'];
 writeln(ch1);
end;

var sel : char;
begin
 clrscr;
 repeat
  getoption(sel);
  case sel of
    'E' : Enpart;
    'O' : Enopr(oprfi);
    'D' : Display(planfi);
    'K' : Display1(oprfi);
    'T' : Tabusearch;
   end;
 until sel = 'Q'
end.
```

ภาคผนวก จ
รายละเอียดไฟล์ตัวหนังสือของโปรแกรม Model2

เนื้อหารายละเอียดในส่วนนี้จะเสนอถึงรายละเอียดไฟล์ตัวหนังสือของโปรแกรม Model2 หรือโปรแกรมการค้นหาคำตอบแบบทาบูของการเลือกเครื่องจักรในการผลิต โดยมีเงื่อนไขของโปรแกรมดังนี้

1. เลือก Neighborhood List แบบ Swap Pairwise Interchange หรือ Insertion Interchange
2. เลือก Tabu List Size 3, 5, 7
3. ต้นทุนแบบที่ 2

รายละเอียดของโปรแกรมเป็นดังนี้

```
Program enterorp;
uses crt,printer,swaplib;
type  planrect = record
            partnum  : integer;
            partna   : array[1..19] of string[3];
            invcost  : array[1..19] of longint;
            plannum  : array[1..19] of shortint;
            planna   : array[1..19,1..3] of string[3];
            opnum    : array[1..19,1..3] of shortint;
            opna     : array[1..19,1..3,1..20] of string[3];
            end;
type oprrect = record
            opnumm  : shortint;
            opname  : array[1..20] of string[3];
            machnum  : array[1..20] of shortint;
            machna   : array[1..20,1..20] of string[3];
            machcost : array[1..20,1..20] of integer;
            setcost  : array[1..20,1..20] of integer;
            prodcost : array[1..20,1..20] of longint;
            end;

   fitype = file of planrect;
   fitype1 = file of oprrect;
var cnt,cnt1,part  : integer;
```

```
    cost : longint;
    a,b,c,i   : integer;
    d,l,m,n     : longint;
    planfi    : fitype;
    oprfi     : fitype1;
    planrec   : planrect;
    oprrec    : oprrect;
    tabu      : array[1..20,1..3] of longint;
    method    : array[1..20] of longint;
    solution  : array[1..20] of longint;
    solve     : array[1..20] of longint;
    sol       : array[1..20] of longint;
    tabulist  : array[1..7] of longint;
    tabupos   : array[1..7] of integer;
    aspiration : array[1..7] of longint;
    tempo     : array[1..20] of longint;
    solve1 : integer;
    num,more,ran1,ran2 : integer;
    tabu1,minimum,sum,cunt : longint;
    num1,num2,num3,num4,num5,num6,num7,num8,num9,num10,num11,
    num12,num13,num14,num15,num16,num17,num18,num19,num20  : integer;
    a1,a2,a3,a4,a5,a6,a7,a8,a9,a10,a11,a12,a13,a14,a15,a16,a17,a16,a19,a20 : integer;

Procedure prepare(var fi : fitype);
var finame : string[12];
begin
  clrscr;
  write('Enter file "s name : '); readln(finame);
  assign(fi,finame);
  {$I-} reset(fi); {$I+}
  if ioresult <> 0 then
  begin  rewrite(fi);  close(fi);  reset(fi);  end;
end;

Procedure prepare1(var fi1 : fitype1);
var finame : string[12];
begin
  clrscr;
```

```
    write('Enter file "s name : '); readln(finame);
    assign(fi1,finame);
    {$I-} reset(fi1); {$I+}
    if ioresult <> 0 then
    begin rewrite(fi1); close(fi1); reset(fi1); end;
  end;

  Procedure Enpart;
  var part,partnum1,plan,plannum1,opr,oprnum1 : integer;
  begin
   prepare(planfi);
   with planrec do
  begin
   clrscr;
   write('Enter number of Part : '); readln(partnum);
   for part := 1 to partnum do
  begin
   clrscr;
   writeln('Part ',part);
   write('Enter number of Plan : '); readln(plannum[part]);
   for plan := 1 to plannum[part] do
 begin
   clrscr;
   writeln('Part ',part,' Plan ',plan);
   write('Enter number of Operation : '); readln(opmum[part,plan]);
   for opr := 1 to opmum[part,plan] do
 begin
   clrscr;
   writeln('Part ',part,' Plan ',plan,' Operation ',opr);
   write('Enter Part name      : '); readln(partna[part]);
   write('Enter Inventory cost : '); readln(invcost[part]);
   write('Enter Plan name      : '); readln(planna[part,plan]);
   write('Enter Operation name : '); readln(opma[part,plan,opr]);
end;{for_opr}
end;{for_plan}
end;{for_part}
write(planfi,planrec);
end;{with}
```

```
close(planfi);
end;{main}

Procedure Enopr(var fi1 : fitype1);
var opr2,oprnum2,mach,machnum1 : integer;
begin
  prepare1(oprfi);
  with oprrec do
begin
  clrscr;
  write('Enter number of Operation : '); readln(oprnumm);
  for opr2 := 1 to oprnumm do
begin
  clrscr;
  writeln('Operation ',opr2);
  write('Enter number of machine : '); readln(machnum[opr2]);
  for mach :=  1 to machnum[opr2] do
begin
  clrscr;
  writeln('Operation ',opr2,' Machine ',mach);
  write('Enter Operation name  : '); readln(oprname[opr2]);
  write('Enter Machine name    : '); readln(machna[opr2,mach]);
  write('Enter Machine Cost  : '); readln(machcost[opr2,mach]);
  write('Enter Machine Setup Cost : '); readln(setcost[opr2,mach]);
  write('Enter Production Cost : '); readln(prodcost[opr2,mach]);
end;{for_mach}
end;{for_opr}
  write(oprfi,oprrec);
end;{with}
  close(oprfi);
end;{main}

Function compword(astr,wrd : string; p : byte) : boolean;
var s : byte;
begin
  compword := false;
  for s := 2 to length(wrd) do
begin
```

```
    p := p+1;
    if wrd[s] <> astr[p] then exit;
   end;
    compword := true;
   end;

   Function findword(astr,wrd : string) : byte;
   var p : byte;
   begin
    for p := 1 to length(astr) do
    if wrd[1] = astr[p] then
    if compword(astr,wrd,p) then
  begin
   findword := p; exit
  end;
   findword := 0;
  end;

  Function count(num,tem : integer) : integer;
  begin
   with planrec do
   with oprrec do
  begin
   repeat
   num := num+1;
   findword(oprna[cnt,cnt1,tem],opname[num]);
   until oprna[cnt,cnt1,tem]=opname[num];
   count := num;
 end;
 end;

 Procedure machine1;
begin
 with planrec do
 with oprrec do
begin
 writeln(machna[count(num1,1),a1]);
 cost :=machcost[count(num1,1),a1]+setcost[count(num1,1),a1]+prodcost[count(num1,1),a1];
```

```
    sum:=sum+cost+invcost[cnt];
  end; end;

  Procedure machine2;
  begin
    with planrec do
    with oprrec do
  begin
    writeln(machna[count(num1,1),a1],'-',machna[count(num2,2),a2]);
    cost := machcost[count(num1,1),a1]+machcost[count(num2,2),a2]
          +setcost[count(num1,1),a1]+setcost[count(num2,2),a2]
          +prodcost[count(num1,1),a1]+prodcost[count(num2,2),a2];
    sum:=sum+cost+invcost[cnt];
  end; end;

  Procedure machine3;
  begin
    with planrec do
    with oprrec do
  begin
    writeln(machna[count(num1,1),a1],'-',machna[count(num2,2),a2]
    ,'-',machna[count(num3,3),a3]);
    cost :=machcost[count(num1,1),a1]+machcost[count(num2,2),a2]
         +machcost[count(num3,3),a3]+setcost[count(num1,1),a1]
         +setcost[count(num2,2),a2]+setcost[count(num3,3),a3]
         +prodcost[count(num1,1),a1]+prodcost[count(num2,2),a2]
         +prodcost[count(num3,3),a3];
    sum:=sum+cost+invcost[cnt];
  end; end;

  Procedure machine4;
  begin
    with planrec do
    with oprrec do
  begin
    writeln(machna[count(num1,1),a1],'-',machna[count(num2,2),a2]
    ,'-',machna[count(num3,3),a3],'-',machna[count(num4,4),a4]);
    cost :=machcost[count(num1,1),a1]+machcost[count(num2,2),a2]
```

```
        +machcost[count(num3,3),a3]+machcost[count(num4,4),a4]
        +setcost[count(num1,1),a1]+setcost[count(num2,2),a2]
        +setcost[count(num3,3),a3]+setcost[count(num4,4),a4]
        +prodcost[count(num1,1),a1]+prodcost[count(num2,2),a2]
        +prodcost[count(num3,3),a3]+prodcost[count(num4,4),a4];
   sum:=sum+cost+invcost[cnt];
 end; end;

 Procedure machine5;
 begin
  with planrec do
  with oprrec do
 begin
  writeln(machna[count(num1,1),a1],'-',machna[count(num2,2),a2]
  ,'-',machna[count(num3,3),a3],'-',machna[count(num4,4),a4]
  ,'-',machna[count(num5,5),a5]);
  cost :=machcost[count(num1,1),a1]+machcost[count(num2,2),a2]+machcost[count(num3,3),a3]
       +machcost[count(num4,4),a4]+machcost[count(num5,5),a5]+setcost[count(num1,1),a1]
       +setcost[count(num2,2),a2]+setcost[count(num3,3),a3]+setcost[count(num4,4),a4]+setcost
[count(num5,5),a5]
       +prodcost[count(num1,1),a1]+prodcost[count(num2,2),a2]+prodcost[count(num3,3),a3]
       +prodcost[count(num4,4),a4]+prodcost[count(num5,5),a5];
  sum:=sum+cost+invcost[cnt];
end; end;

Procedure machine6;
begin
 with planrec do
 with oprrec do
begin
 writeln(machna[count(num1,1),a1],'-',machna[count(num2,2),a2]
 ,'-',machna[count(num3,3),a3],'-',machna[count(num4,4),a4]
 ,'-',machna[count(num5,5),a5],'-',machna[count(num6,6),a6]);
 cost :=machcost[count(num1,1),a1]+machcost[count(num2,2),a2]+machcost[count(num3,3),a3]
      +machcost[count(num4,4),a4]+machcost[count(num5,5),a5]+machcost[count(num6,6),a6]
      +setcost[count(num1,1),a1]+setcost[count(num2,2),a2]+setcost[count(num3,3),a3]
      +setcost[count(num4,4),a4]+setcost[count(num5,5),a5]+setcost[count(num6,6),a6]
      +prodcost[count(num1,1),a1]+prodcost[count(num2,2),a2]+prodcost[count(num3,3),a3]
```

```
        +prodcost[count(num4,4),a4]+prodcost[count(num5,5),a5]+prodcost[count(num6,6),a6];
    sum:=sum+cost+invcost[cnt];
  end; end;

  Procedure machine7;
  begin
   with planrec do
   with oprrec do
  begin
   writeln(machna[count(num1,1),a1],'-',machna[count(num2,2),a2]
   ,'-',machna[count(num3,3),a3],'-',machna[count(num4,4),a4]
   ,'-',machna[count(num5,5),a5],'-',machna[count(num6,6),a6]
   ,'-',machna[count(num7,7),a7]);
   cost :=machcost[count(num1,1),a1]+machcost[count(num2,2),a2]+machcost[count(num3,3),a3]
       +machcost[count(num4,4),a4]+machcost[count(num5,5),a5]+machcost[count(num6,6),a6]
       +machcost[count(num7,7),a7]+setcost[count(num1,1),a1]+setcost[count(num2,2),a2]
       +setcost[count(num3,3),a3]+setcost[count(num4,4),a4]+setcost[count(num5,5),a5]
       +setcost[count(num6,6),a6]+setcost[count(num7,7),a7]+prodcost[count(num1,1),a1]
       +prodcost[count(num2,2),a2]+prodcost[count(num3,3),a3]+prodcost[count(num4,4),a4]
       +prodcost[count(num5,5),a5]+prodcost[count(num6,6),a6]+prodcost[count(num7,7),a7];
   sum:=sum+cost+invcost[cnt];
  end; end;

  Procedure machine8;
  begin
   with planrec do
   with oprrec do
  begin
   writeln(machna[count(num1,1),a1],'-',machna[count(num2,2),a2]
   ,'-',machna[count(num3,3),a3],'-',machna[count(num4,4),a4]
   ,'-',machna[count(num5,5),a5],'-',machna[count(num6,6),a6]
   ,'-',machna[count(num7,7),a7],'-',machna[count(num8,8),a8]);
   cost :=machcost[count(num1,1),a1]+machcost[count(num2,2),a2]+machcost[count(num3,3),a3]
       +machcost[count(num4,4),a4]+machcost[count(num5,5),a5]+machcost[count(num6,6),a6]
       +machcost[count(num7,7),a7]+machcost[count(num8,8),a8]+setcost[count(num1,1),a1]
       +setcost[count(num2,2),a2]+setcost[count(num3,3),a3]+setcost[count(num4,4),a4]
       +setcost[count(num5,5),a5]+setcost[count(num6,6),a6]+setcost[count(num7,7),a7]
       +setcost[count(num8,8),a8]+prodcost[count(num1,1),a1]+prodcost[count(num2,2),a2]
```

```
        +prodcost[count(num3,3),a3]+prodcost[count(num4,4),a4]+prodcost[count(num5,5),a5]
        +prodcost[count(num6,6),a6]+prodcost[count(num7,7),a7]+prodcost[count(num8,8),a8];
   sum:=sum+cost+invcost[cnt];
  end; end;

  Procedure machine9;
  begin
   with planrec do
   with oprrec do
  begin
   writeln(machna[count(num1,1),a1],'-',machna[count(num2,2),a2]
   ,'-',machna[count(num3,3),a3],'-',machna[count(num4,4),a4]
   ,'-',machna[count(num5,5),a5],'-',machna[count(num6,6),a6]
   ,'-',machna[count(num7,7),a7],'-',machna[count(num8,8),a8]
   ,'-',machna[count(num9,9),a9]);
   cost :=machcost[count(num1,1),a1]+machcost[count(num2,2),a2]+machcost[count(num3,3),a3]
        +machcost[count(num4,4),a4]+machcost[count(num5,5),a5]+machcost[count(num6,6),a6]
        +machcost[count(num7,7),a7]+machcost[count(num8,8),a8]+machcost[count(num9,9),a9]
        +setcost[count(num1,1),a1]+setcost[count(num2,2),a2]+setcost[count(num3,3),a3]
        +setcost[count(num4,4),a4]+setcost[count(num5,5),a5]+setcost[count(num6,6),a6]
        +setcost[count(num7,7),a7]+setcost[count(num8,8),a8]+setcost[count(num9,9),a9]
        +prodcost[count(num1,1),a1]+prodcost[count(num2,2),a2]+prodcost[count(num3,3),a3]
        +prodcost[count(num4,4),a4]+prodcost[count(num5,5),a5]+prodcost[count(num6,6),a6]
        +prodcost[count(num7,7),a7]+prodcost[count(num8,8),a8]+prodcost[count(num9,9),a9];
   sum:=sum+cost+invcost[cnt];
  end; end;

  Procedure machine10;
  begin
   with planrec do
   with oprrec do
  begin
   writeln(machna[count(num1,1),a1],'-',machna[count(num2,2),a2]
   ,'-',machna[count(num3,3),a3],'-',machna[count(num4,4),a4]
   ,'-',machna[count(num5,5),a5],'-',machna[count(num6,6),a6]
   ,'-',machna[count(num7,7),a7],'-',machna[count(num8,8),a8]
   ,'-',machna[count(num9,9),a9],'-',machna[count(num10,10),a10]);
   cost :=machcost[count(num1,1),a1]+machcost[count(num2,2),a2]+machcost[count(num3,3),a3]
```

```
      +machcost[count(num4,4),a4]+machcost[count(num5,5),a5]+machcost[count(num6,6),a6]
      +machcost[count(num7,7),a7]+machcost[count(num8,8),a8]+machcost[count(num9,9),a9]
      +machcost[count(num10,10),a10]+setcost[count(num1,1),a1]+setcost[count(num2,2),a2]
      +setcost[count(num3,3),a3]+setcost[count(num4,4),a4]+setcost[count(num5,5),a5]
      +setcost[count(num6,6),a6]+setcost[count(num7,7),a7]+setcost[count(num8,8),a8]
      +setcost[count(num9,9),a9]+setcost[count(num10,10),a10]+prodcost[count(num1,1),a1]
      +prodcost[count(num2,2),a2]+prodcost[count(num3,3),a3]+prodcost[count(num4,4),a4]
      +prodcost[count(num5,5),a5]+prodcost[count(num6,6),a6]+prodcost[count(num7,7),a7]
      +prodcost[count(num8,6),a8]+prodcost[count(num9,9),a9]+prodcost[count(num10,10),a10];
  sum:=sum+cost+invcost[cnt];
end; end;

Procedure machine11;
begin
  with planrec do
  with oprrec do
begin
  writeln(machna[count(num1,1),a1],'-',machna[count(num2,2),a2]
  ,'-',machna[count(num3,3),a3],'-',machna[count(num4,4),a4]
  ,'-',machna[count(num5,5),a5],'-',machna[count(num6,6),a6]
  ,'-',machna[count(num7,7),a7],'-',machna[count(num8,8),a8]
  ,'-',machna[count(num9,9),a9],'-',machna[count(num10,10),a10]
  ,'-',machna[count(num11,11),a11]);
  cost :=machcost[count(num1,1),a1]+machcost[count(num2,2),a2]+machcost[count(num3,3),a3]
      +machcost[count(num4,4),a4]+machcost[count(num5,5),a5]+machcost[count(num6,6),a6]
      +machcost[count(num7,7),a7]+machcost[count(num8,8),a8]+machcost[count(num9,9),a9]
      +machcost[count(num10,10),a10]+machcost[count(num11,11),a11]+setcost[count(num1,1),a1]
      +setcost[count(num2,2),a2]+setcost[count(num3,3),a3]+setcost[count(num4,4),a4]
      +setcost[count(num5,5),a5]+setcost[count(num6,6),a6]+setcost[count(num7,7),a7]
      +setcost[count(num8,8),a8]+setcost[count(num9,9),a9]+setcost[count(num10,10),a10]
      +setcost[count(num11,11),a11]+prodcost[count(num1,1),a1]+prodcost[count(num2,2),a2]
      +prodcost[count(num3,3),a3]+prodcost[count(num4,4),a4]+prodcost[count(num5,5),a5]
      +prodcost[count(num6,6),a6]+prodcost[count(num7,7),a7]+prodcost[count(num8,8),a8]
      +prodcost[count(num9,9),a9]+prodcost[count(num10,10),a10]+prodcost[count(num11,11),a11];
  sum:=sum+cost+invcost[cnt];
end; end;

Procedure machine12;
```

```
begin
  with planrec do
  with oprrec do
begin
  writeln(machna[count(num1,1),a1],'-',machna[count(num2,2),a2]
  ,'-',machna[count(num3,3),a3],'-',machna[count(num4,4),a4]
  ,'-',machna[count(num5,5),a5],'-',machna[count(num6,6),a6]
  ,'-',machna[count(num7,7),a7],'-',machna[count(num8,8),a8]
  ,'-',machna[count(num9,9),a9],'-',machna[count(num10,10),a10]
  ,'-',machna[count(num11,11),a11],'-',machna[count(num12,12),a12]);
  cost :=machcost[count(num1,1),a1]+machcost[count(num2,2),a2]+machcost[count(num3,3),a3]
       +machcost[count(num4,4),a4]+machcost[count(num5,5),a5]+machcost[count(num6,6),a6]
       +machcost[count(num7,7),a7]+machcost[count(num8,8),a8]+machcost[count(num9,9),a9]
       +machcost[count(num10,10),a10]+machcost[count(num11,11),a11]+machcost[count
(num12,12),a12]
       +setcost[count(num1,1),a1]+setcost[count(num2,2),a2]+setcost[count(num3,3),a3]
       +setcost[count(num4,4),a4]+setcost[count(num5,5),a5]+setcost[count(num6,6),a6]
       +setcost[count(num7,7),a7]+setcost[count(num8,8),a8]+setcost[count(num9,9),a9]
       +setcost[count(num10,10),a10]+setcost[count(num11,11),a11]+setcost[count(num12,12),a12]
       +prodcost[count(num1,1),a1]+prodcost[count(num2,2),a2]+prodcost[count(num3,3),a3]
       +prodcost[count(num4,4),a4]+prodcost[count(num5,5),a5]+prodcost[count(num6,6),a6]
       +prodcost[count(num7,7),a7]+prodcost[count(num8,8),a8]+prodcost[count(num9,9),a9]
       +prodcost[count(num10,10),a10]+prodcost[count(num11,11),a11]+prodcost[count
(num12,12),a12];
  sum:=sum+cost+invcost[cnt];
end; end;

Procedure machine13;
begin
  with planrec do
  with oprrec do
begin
  writeln(machna[count(num1,1),a1],'-',machna[count(num2,2),a2]
  ,'-',machna[count(num3,3),a3],'-',machna[count(num4,4),a4]
  ,'-',machna[count(num5,5),a5],'-',machna[count(num6,6),a6]
  ,'-',machna[count(num7,7),a7],'-',machna[count(num8,8),a8]
  ,'-',machna[count(num9,9),a9],'-',machna[count(num10,10),a10]
  ,'-',machna[count(num11,11),a11],'-',machna[count(num12,12),a12]
```

```
  ,'-',machna[count(num13,13),a13]);
  cost :=machcost[count(num1,1),a1]+machcost[count(num2,2),a2]+machcost[count(num3,3),a3]
       +machcost[count(num4,4),a4]+machcost[count(num5,5),a5]+machcost[count(num6,6),a6]
       +machcost[count(num7,7),a7]+machcost[count(num8,8),a8]+machcost[count(num9,9),a9]
       +machcost[count(num10,10),a10]+machcost[count(num11,11),a11]+machcost[count
(num12,12),a12]
       +machcost[count(num13,13),a13]+setcost[count(num1,1),a1]+setcost[count(num2,2),a2]
       +setcost[count(num3,3),a3]+setcost[count(num4,4),a4]+setcost[count(num5,5),a5]
       +setcost[count(num6,6),a6]+setcost[count(num7,7),a7]+setcost[count(num8,8),a8]
       +setcost[count(num9,9),a9]+setcost[count(num10,10),a10]+setcost[count(num11,11),a11]
       +setcost[count(num12,12),a12]+setcost[count(num13,13),a13]+prodcost[count(num1,1),a1]
       +prodcost[count(num2,2),a2]+prodcost[count(num3,3),a3]+prodcost[count(num4,4),a4]
       +prodcost[count(num5,5),a5]+prodcost[count(num6,6),a6]+prodcost[count(num7,7),a7]
       +prodcost[count(num8,8),a8]+prodcost[count(num9,9),a9]+prodcost[count(num10,10),a10]
       +prodcost[count(num11,11),a11]+prodcost[count(num12,12),a12]+prodcost[count
(num13,13),a13];
  sum:=sum+cost+invcost[cnt];
end; end;

Procedure machine14;
begin
  with planrec do
  with oprrec do
begin
  writeln(machna[count(num1,1),a1],'-',machna[count(num2,2),a2]
  ,'-',machna[count(num3,3),a3],'-',machna[count(num4,4),a4]
  ,'-',machna[count(num5,5),a5],'-',machna[count(num6,6),a6]
  ,'-',machna[count(num7,7),a7],'-',machna[count(num8,8),a8]
  ,'-',machna[count(num9,9),a9],'-',machna[count(num10,10),a10]
  ,'-',machna[count(num11,11),a11],'-',machna[count(num12,12),a12]
  ,'-',machna[count(num13,13),a13],'-',machna[count(num14,14),a14]);
  cost :=machcost[count(num1,1),a1]+machcost[count(num2,2),a2]+machcost[count(num3,3),a3]
       +machcost[count(num4,4),a4]+machcost[count(num5,5),a5]+machcost[count(num6,6),a6]
       +machcost[count(num7,7),a7]+machcost[count(num8,8),a8]+machcost[count(num9,9),a9]
       +machcost[count(num10,10),a10]+machcost[count(num11,11),a11]+machcost[count
(num12,12),a12]
       +machcost[count(num13,13),a13]+machcost[count(num14,14),a14]+setcost[count(num1,1),a1]
       +setcost[count(num2,2),a2]+setcost[count(num3,3),a3]+setcost[count(num4,4),a4]
```

```
      +setcost[count(num5,5),a5]+setcost[count(num6,6),a6]+setcost[count(num7,7),a7]
      +setcost[count(num8,8),a8]+setcost[count(num9,9),a9]+setcost[count(num10,10),a10]
      +setcost[count(num11,11),a11]+setcost[count(num12,12),a12]+setcost[count(num13,13),a13]
      +setcost[count(num14,14),a14]+prodcost[count(num1,1),a1]+prodcost[count(num2,2),a2]
      +prodcost[count(num3,3),a3]+prodcost[count(num4,4),a4]+prodcost[count(num5,5),a5]
      +prodcost[count(num6,6),a6]+prodcost[count(num7,7),a7]+prodcost[count(num8,8),a8]
      +prodcost[count(num9,9),a9]+prodcost[count(num10,10),a10]+prodcost[count(num11,11),a11]
      +prodcost[count(num12,12),a12]+prodcost[count(num13,13),a13]+prodcost[count
(num14,14),a14];
  sum:=sum+cost+invcost[cnt];
end; end;

Procedure machine15;
begin
  with planrec do
  with oprrec do
begin
  writeln(machna[count(num1,1),a1],'-',machna[count(num2,2),a2]
  ,'-',machna[count(num3,3),a3],'-',machna[count(num4,4),a4]
  ,'-',machna[count(num5,5),a5] ,'-',machna[count(num6,6),a6]
  ,'-',machna[count(num7,7),a7],'-',machna[count(num8,8),a8]
  ,'-',machna[count(num9,9),a9],'-',machna[count(num10,10),a10]
  ,'-',machna[count(num11,11),a11],'-',machna[count(num12,12),a12]
  ,'-',machna[count(num13,13),a13],'-',machna[count(num14,14),a14]
  ,'-',machna[count(num15,15),a15]);
  cost :=machcost[count(num1,1),a1]+machcost[count(num2,2),a2]+machcost[count(num3,3),a3]
      +machcost[count(num4,4),a4]+machcost[count(num5,5),a5]+machcost[count(num6,6),a6]
      +machcost[count(num7,7),a7]+machcost[count(num8,8),a8]+machcost[count(num9,9),a9]
      +machcost[count(num10,10),a10]+machcost[count(num11,11),a11]+machcost[count
(num12,12),a12]
      +machcost[count(num13,13),a13]+machcost[count(num14,14),a14]+machcost[count
(num15,15),a15]
      +setcost[count(num1,1),a1]+setcost[count(num2,2),a2]+setcost[count(num3,3),a3]
      +setcost[count(num4,4),a4]+setcost[count(num5,5),a5]+setcost[count(num6,6),a6]
      +setcost[count(num7,7),a7]+setcost[count(num8,8),a8]+setcost[count(num9,9),a9]
      +setcost[count(num10,10),a10]+setcost[count(num11,11),a11]+setcost[count(num12,12),a12]
      +setcost[count(num13,13),a13]+setcost[count(num14,14),a14]+setcost[count(num15,15),a15]
      +prodcost[count(num1,1),a1]+prodcost[count(num2,2),a2]+prodcost[count(num3,3),a3]
```

```
      +prodcost[count(num4,4),a4]+prodcost[count(num5,5),a5]+prodcost[count(num6,6),a6]
      +prodcost[count(num7,7),a7]+prodcost[count(num8,6),a8]+prodcost[count(num9,9),a9]
      +prodcost[count(num10,10),a10]+prodcost[count(num11,11),a11]+prodcost[count
(num12,12),a12]
      +prodcost[count(num13,13),a13]+prodcost[count(num14,14),a14]+prodcost[count
(num15,15),a15];
  sum:=sum+cost+invcost[cnt];
end; end;

Procedure machine16;
begin
  with planrec do
  with oprrec do
begin
  writeln(machna[count(num1,1),a1],'-',machna[count(num2,2),a2]
  ,'-',machna[count(num3,3),a3],'-',machna[count(num4,4),a4]
  ,'-',machna[count(num5,5),a5],'-',machna[count(num6,6),a6]
  ,'-',machna[count(num7,7),a7],'-',machna[count(num8,8),a8]
  ,'-',machna[count(num9,9),a9],'-',machna[count(num10,10),a10]
  ,'-',machna[count(num11,11),a11],'-',machna[count(num12,12),a12]
  ,'-',machna[count(num13,13),a13],'-',machna[count(num14,14),a14]
  ,'-',machna[count(num15,15),a15],'-',machna[count(num16,16),a16]);
  cost :=machcost[count(num1,1),a1]+machcost[count(num2,2),a2]+machcost[count(num3,3),a3]
      +machcost[count(num4,4),a4]+machcost[count(num5,5),a5]+machcost[count(num6,6),a6]
      +machcost[count(num7,7),a7]+machcost[count(num6,8),a8]+machcost[count(num9,9),a9]
      +machcost[count(num10,10),a10]+machcost[count(num11,11),a11]+maohcost[count
(num12,12),a12]
      +machcost[count(num13,13),a13]+machcost[count(num14,14),a14]+machcost[count
(num15,15),a15]
      +machcost[count(num16,16),a16]+setcost[count(num1,1),a1]+setcost[count(num2,2),a2]
      +setcost[count(num3,3),a3]+setcost[count(num4,4),a4]+setcost[count(num5,5),a5]
      +setcost[count(num6,6),a6]+setcost[count(num7,7),a7]+setcost[count(num6,8),a8]
      +setcost[count(num9,9),a9]+setcost[count(num10,10),a10]+setcost[count(num11,11),a11]
      +setcost[count(num12,12),a12]+setcost[count(num13,13),a13]+setcost[count(num14,14),a14]
      +setcost[count(num15,15),a15]+setcost[count(num18,16),a16]+prodcost[count(num1,1),a1]
      +prodcost[count(num2,2),a2]+prodcost[count(num3,3),a3]+prodcost[count(num4,4),a4]
      +prodcost[count(num5,5),a5]+prodcost[count(num6,8),a6]+prodcost[count(num7,7),a7]
      +prodcost[count(num8,8),a8]+prodcost[count(num9,9),a9]+prodcost[count(num10,10),a10]
```

```
      +prodcost[count(num11,11),a11]+prodcost[count(num12,12),a12]+prodcost[count
(num13,13),a13]
      +prodcost[count(num14,14),a14]+prodcost[count(num15,15),a15]+prodcost[count
(num16,16),a16];
  sum:=sum+cost+invcost[cnt];
end; end;

Procedure machine17;
begin
  with planrec do
  with oprrec do
begin
  writeln(machna[count(num1,1),a1],'-',machna[count(num2,2),a2]
  ,'-',machna[count(num3,3),a3],'-',machna[count(num4,4),a4]
  ,'-',machna[count(num5,5),a5],'-',machna[count(num6,6),a6]
  ,'-',machna[count(num7,7),a7],'-',machna[count(num8,8),a8]
  ,'-',machna[count(num9,9),a9],'-',machna[count(num10,10),a10]
  ,'-',machna[count(num11,11),a11],'-',machna[count(num12,12),a12]
  ,'-',machna[count(num13,13),a13],'-',machna[count(num14,14),a14]
  ,'-',machna[count(num15,15),a15],'-',machna[count(num16,16),a16]
  ,'-',machna[count(num17,17),a17]);
  cost := machcost[count(num1,1),a1]+machcost[count(num2,2),a2]+machcost[count(num3,3),a3]
      +machcost[count(num4,4),a4]+machcost[count(num5,5),a5]+machcost[count(num6,6),a6]
      +machcost[count(num7,7),a7]+machcost[count(num8,8),a8]+machcost[count(num9,9),a9]
      +machcost[count(num10,10),a10]+machcost[count(num11,11),a11]+machcost[count
(num12,12),a12]
      +machcost[count(num13,13),a13]+machcost[count(num14,14),a14]+machcost[count
(num15,15),a15]
      +machcost[count(num16,16),a16]+machcost[count(num17,17),a17]+setcost[count(num1,1),a1]
      +setcost[count(num2,2),a2]+setcost[count(num3,3),a3]+setcost[count(num4,4),a4]
      +setcost[count(num5,5),a5]+setcost[count(num6,6),a6]+setcost[count(num7,7),a7]
      +setcost[count(num8,8),a8]+setcost[count(num9,9),a9]+setcost[count(num10,10),a10]
      +setcost[count(num11,11),a11]+setcost[count(num12,12),a12]+setcost[count(num13,13),a13]
      +setcost[count(num14,14),a14]+setcost[count(num15,15),a15]+setcost[count(num16,16),a18]
      +setcost[count(num17,17),a17]+prodcost[count(num1,1),a1]+prodcost[count(num2,2),a2]
      +prodcost[count(num3,3),a3]+prodcost[count(num4,4),a4]+prodcost[count(num5,5),a5]
      +prodcost[count(num6,6),a6]+prodcost[count(num7,7),a7]+prodcost[count(num8,8),a8]
      +prodcost[count(num9,9),a9]+prodcost[count(num10,10),a10]+prodcost[count(num11,11),a11]
```

```
      +prodcost[count(num12,12),a12]+prodcost[count(num13,13),a13]+prodcost[count
(num14,14),a14]
      +prodcost[count(num15,15),a15]+prodcost[count(num16,16),a16]+prodcost[count
(num17,17),a17];
  sum:=sum+cost+invcost[cnt];
end; end;

Procedure machine18;
begin
  with planrec do
  with oprrec do
begin
  writeln(machna[count(num1,1),a1],'-',machna[count(num2,2),a2]
  ,'-',machna[count(num3,3),a3],'-',machna[count(num4,4),a4]
  ,'-',machna[count(num5,5),a5],'-',machna[count(num6,6),a6]
  ,'-',machna[count(num7,7),a7],'-',machna[count(num8,8),a8]
  ,'-',machna[count(num9,9),a9],'-',machna[count(num10,10),a10]
  ,'-',machna[count(num11,11),a11],'-',machna[count(num12,12),a12]
  ,'-',machna[count(num13,13),a13],'-',machna[count(num14,14),a14]
  ,'-',machna[count(num15,15),a15],'-',machna[count(num16,16),a16]
  ,'-',machna[count(num17,17),a17],'-',machna[count(num18,18),a18]);
  cost :=machcost[count(num1,1),a1]+machcost[count(num2,2),a2]+machcost[count(num3,3),a3]
      +machcost[count(num4,4),a4]+machcost[count(num5,5),a5]+machcost[count(num6,6),a6]
      +machcost[count(num7,7),a7]+machcost[count(num8,8),a8]+machcost[count(num9,9),a9]
      +machcost[count(num10,10),a10]+machcost[count(num11,11),a11]+machcost[count
(num12,12),a12]
      +machcost[count(num13,13),a13]+machcost[count(num14,14),a14]+machcost[count
(num15,15),a15]
      +machcost[count(num16,16),a16]+machcost[count(num17,17),a17]+machcost[count
(num18,18),a18]
      +setcost[count(num1,1),a1]+setcost[count(num2,2),a2]+setcost[count(num3,3),a3]
      +setcost[count(num4,4),a4]+setcost[count(num5,5),a5]+setcost[count(num6,6),a6]
      +setcost[count(num7,7),a7]+setcost[count(num8,8),a8]+setcost[count(num9,9),a9]
      +setcost[count(num10,10),a10]+setcost[count(num11,11),a11]+setcost[count(num12,12),a12]
      +setcost[count(num13,13),a13]+setcost[count(num14,14),a14]+setcost[count(num15,15),a15]
      +setcost[count(num16,16),a16]+setcost[count(num17,17),a17]+setcost[count(num16,18),a18]
      +prodcost[count(num1,1),a1]+prodcost[count(num2,2),a2]+prodcost[count(num3,3),a3]
      +prodcost[count(num4,4),a4]+prodcost[count(num5,5),a5]+prodcost[count(num6,6),a6]
```

```
      +prodcost[count(num7,7),a7]+prodcost[count(num8,8),a8]+prodcost[count(num9,9),a9]
      +prodcost[count(num10,10),a10]+prodcost[count(num11,11),a11]+prodcost[count
(num12,12),a12]
      +prodcost[count(num13,13),a13]+prodcost[count(num14,14),a14]+prodcost[count
(num15,15),a15]
      +prodcost[count(num16,16),a16]+prodcost[count(num17,17),a17]+prodcost[count
(num18,18),a16];
  sum:=sum+cost+invcost[cnt];
end; end;

Procedure machine19;
begin
  with planrec do
  with oprrec do
 begin
  writeln(machna[count(num1,1),a1],'-',machna[count(num2,2),a2]
  ,'-',machna[count(num3,3),a3],'-',machna[count(num4,4),a4]
  ,'-',machna[count(num5,5),a5],'-',machna[count(num6,6),a6]
  ,'-',machna[count(num7,7),a7],'-',machna[count(num6,6),a6]
  ,'-',machna[count(num9,9),a9],'-',machna[count(num10,10),a10]
  ,'-',machna[count(num11,11),a11],'-',machna[count(num12,12),a12]
  ,'-',machna[count(num13,13),a13],'-',machna[count(num14,14),a14]
  ,'-',machna[count(num15,15),a15],'-',machna[count(num16,16),a16]
  ,'-',machna[count(num17,17),a17],'-',machna[count(num18,18),a16]
  ,'-',machna[count(num19,19),a19]);
  cost :=machcost[count(num1,1),a1]+machcost[count(num2,2),a2]+machcost[count(num3,3),a3]
       +machcost[count(num4,4),a4]+machcost[count(num5,5),a5]+machcost[count(num6,6),a6]
       +machcost[count(num7,7),a7]+machcost[count(num6,8),a8]+machcost[count(num9,9),a9]
       +machcost[count(num10,10),a10]+machcost[count(num11,11),a11]+machcost[count
 (num12,12),a12]
       +machcost[count(num13,13),a13]+machcost[count(num14,14),a14]+machcost[count
 (num15,15),a15]
       +machcost[count(num16,16),a16]+machcost[count(num17,17),a17]+machcost[count
 (num18,18),a18]
       +machcost[count(num19,19),a19]+setcost[count(num1,1),a1]+setcost[count(num2,2),a2]
       +setcost[count(num3,3),a3]+setcost[count(num4,4),a4]+setcost[count(num5,5),a5]
       +setcost[count(num6,6),a6]+setcost[count(num7,7),a7]+setcost[count(num8,6),a8]
       +setcost[count(num9,9),a9]+setcost[count(num10,10),a10]+setcost[count(num11,11),a11]
```

```
      +setcost[count(num12,12),a12]+setcost[count(num13,13),a13]+setcost[count
(num14,14),a14]
      +setcost[count(num15,15),a15]+setcost[count(num16,16),a16]+setcost[count(num17,17),a17]
      +setcost[count(num18,18),a18]+setcost[count(num19,19),a19]+prodcost[count(num1,1),a1]
      +prodcost[count(num2,2),a2]+prodcost[count(num3,3),a3]+prodcost[count(num4,4),a4]
      +prodcost[count(num5,5),a5]+prodcost[count(num6,6),a6]+prodcost[count(num7,7),a7]
      +prodcost[count(num8,8),a8]+prodcost[count(num9,9),a9]+prodcost[count(num10,10),a10]
      +prodcost[count(num11,11),a11]+prodcost[count(num12,12),a12]+prodcost[count
(num13,13),a13]
      +prodcost[count(num14,14),a14]+prodcost[count(num15,15),a15]+prodcost[count
(num16,16),a16]
      +prodcost[count(num17,17),a17]+prodcost[count(num18,18),a18]+prodcost[count
(num19,19),a19];
  sum:=sum+cost+invcost[ont];
end; end;

Procedure machine20;
begin
  with planrec do
  with oprrec do
begin
  writeln(machna[count(num1,1),a1],'-',machna[count(num2,2),a2]
  ,'-',machna[count(num3,3),a3],'-',machna[count(num4,4),a4]
  ,'-',machna[count(num5,5),a5],'-',machna[count(num6,6),a6]
  ,'-',machna[count(num7,7),a7],'-',machna[count(num8,8),a8]
  ,'-',machna[count(num9,9),a9],'-',machna[count(num10,10),a10]
  ,'-',machna[count(num11,11),a11],'-',machna[count(num12,12),a12]
  ,'-',machna[count(num13,13),a13],'-',machna[count(num14,14),a14]
  ,'-',machna[count(num15,15),a15],'-',machna[count(num16,16),a16]
  ,'-',machna[count(num17,17),a17],'-',machna[count(num18,18),a18]
  ,'-',machna[count(num19,19),a19],'-',machna[count(num20,20),a20]);
  cost :=machcost[count(num1,1),a1]+machcost[count(num2,2),a2]+machcost[count(num3,3),a3]
      +machcost[count(num4,4),a4]+machcost[count(num5,5),a5]+machcost[count(num6,6),a6]
      +machcost[count(num7,7),a7]+machcost[count(num8,8),a8]+machcost[count(num9,9),a9]
      +machcost[count(num10,10),a10]+machcost[count(num11,11),a11]+machcost[count
(num12,12),a12]
      +machcost[count(num13,13),a13]+machcost[count(num14,14),a14]+machcost[count
(num15,15),a15]
```

```
      +machcost[count(num16,16),a16]+machcost[count(num17,17),a17]+machcost[count
(num18,18),a18]
      +machcost[count(num19,19),a19]+machcost[count(num20,20),a20]+setcost[count(num1,1),a1]
      +setcost[count(num2,2),a2]+setcost[count(num3,3),a3]+setcost[count(num4,4),a4]
      +setcost[count(num5,5),a5]+setcost[count(num6,6),a6]+setcost[count(num7,7),a7]
      +setcost[count(num8,8),a8]+setcost[count(num9,9),a9]+setcost[count(num10,10),a10]
      +setcost[count(num11,11),a11]+setcost[count(num12,12),a12]+setcost[count(num13,13),a13]
      +setcost[count(num14,14),a14]+setcost[count(num15,15),a15]+setcost[count(num16,16),a16]
      +setcost[count(num17,17),a17]+setcost[count(num18,18),a18]+setcost[count(num19,19),a19]
      +setcost[count(num20,20),a20]+prodcost[count(num1,1),a1]+prodcost[count(num2,2),a2]
      +prodcost[count(num3,3),a3]+prodcost[count(num4,4),a4]+prodcost[count(num5,5),a5]
      +prodcost[count(num6,6),a6]+prodcost[count(num7,7),a7]+prodcost[count(num8,8),a8]
      +prodcost[count(num9,9),a9]+prodcost[count(num10,10),a10]+prodcost[count(num11,11),a11]
      +prodcost[count(num12,12),a12]+prodcost[count(num13,13),a13]+prodcost[count
(num14,14),a14]
      +prodcost[count(num15,15),a15]+prodcost[count(num16,16),a16]+prodcost[count
(num17,17),a17]
      +prodcost[count(num18,18),a18]+prodcost[count(num19,19),a19]+prodcost[count
(num20,20),a20];
  sum:=sum+cost+invcost[cnt];
end; end;{if}

Procedure operation;
begin
  with planrec do
  with oprrec do
begin
  case oprnum[cnt,cnt1] of
1 :
begin
  num1:=0; cunt:=0;
  count(num1,1);
  for a1 := 1 to machnum[count(num1,1)] do
begin
  cunt := cunt+1;
  if cunt=solution[cnt] then
  machine1;
end;  end;
```

```
2 :
begin
  num1:=0; num2:=0; cunt:=0;
  count(num1,1);
  count(num2,2);
  for a1 := 1 to machnum[count(num1,1)] do
  for a2 := 1 to machnum[count(num2,2)] do
  begin
  cunt := cunt+1;
  if cunt=solution[cnt] then
  machine2;
end;  end;
3 :
begin
  num1:=0; num2:=0; num3:=0; cunt:=0;
  count(num1,1);
  count(num2,2);
  count(num3,3);
  for a1 := 1 to machnum[count(num1,1)] do
  for a2 := 1 to machnum[count(num2,2)] do
  for a3 := 1 to machnum[count(num3,3)] do
begin
  cunt:=cunt+1;
  if cunt=solution[cnt] then
  machine3;
end; end;
4 :
begin
  num1:=0; num2:=0; num3:=0; num4:=0; cunt:=0;
  count(num1,1);
  count(num2,2);
  count(num3,3);
  count(num4,4);
  for a1 := 1 to machnum[count(num1,1)] do
  for a2 := 1 to machnum[count(num2,2)] do
  for a3 := 1 to machnum[count(num3,3)] do
  for a4 := 1 to machnum[count(num4,4)] do
begin
```

```
  cunt := cunt+1;
  if cunt=solution[cnt] then
  machine4;
end; end;
5 :
  begin
  num1:=0; num2:=0; num3:=0; num4:=0; num5:=0; cunt:=0;
  count(num1,1);
  count(num2,2);
  count(num3,3);
  count(num4,4);
  count(num5,5);
  for a1 := 1 to machnum[count(num1,1)] do
  for a2 := 1 to machnum[count(num2,2)] do
  for a3 := 1 to machnum[count(num3,3)] do
  for a4 := 1 to machnum[count(num4,4)] do
  for a5 := 1 to machnum[count(num5,5)] do
begin
  cunt := cunt+1;
  if cunt=solution[cnt] then
  machine5;
end; end;
6 :
  begin
  num1:=0; num2:=0; num3:=0; num4:=0; num5:=0; num6:=0; cunt:=0;
  count(num1,1);
  count(num2,2);
  count(num3,3);
  count(num4,4);
  count(num5,5);
  count(num6,6);
  for a1 := 1 to machnum[count(num1,1)] do
  for a2 := 1 to machnum[count(num2,2)] do
  for a3 := 1 to machnum[count(num3,3)] do
  for a4 := 1 to machnum[count(num4,4)] do
  for a5 := 1 to machnum[count(num5,5)] do
  for a6 := 1 to machnum[count(num6,6)] do
begin
```

```
 cunt := cunt+1;
 If cunt=solution[cnt] then
 machine6;
end; end;
7 :
 begin
 num1:=0; num2:=0; num3:=0; num4:=0; num5:=0; num6:=0; num7:=0; cunt:=0;
 count(num1,1);
 count(num2,2);
 count(num3,3);
 count(num4,4);
 count(num5,5);
 count(num6,6);
 count(num7,7);
 for a1 := 1 to machnum[count(num1,1)] do
 for a2 := 1 to machnum[count(num2,2)] do
 for a3 := 1 to machnum[count(num3,3)] do
 for a4 := 1 to machnum[count(num4,4)] do
 for a5 := 1 to machnum[count(num5,5)] do
 for a6 := 1 to machnum[count(num6,6)] do
 for a7 := 1 to machnum[count(num7,7)] do
begin
 cunt := cunt+1;
 If cunt=solution[cnt] then
 machine7;
end; end;
8 :
 begin
 num1:=0; num2:=0; num3:=0; num4:=0; num5:=0; num6:=0; num7:=0; num8:=0; cunt:=0;
 count(num1,1);
 count(num2,2);
 count(num3,3);
 count(num4,4);
 count(num5,5);
 count(num6,6);
 count(num7,7);
 count(num8,8);
 for a1 := 1 to machnum[count(num1,1)] do
```

```
for a2 := 1 to machnum[count(num2,2)] do
for a3 := 1 to machnum[count(num3,3)] do
for a4 := 1 to machnum[count(num4,4)] do
for a5 := 1 to machnum[count(num5,5)] do
for a6 := 1 to machnum[count(num6,6)] do
for a7 := 1 to machnum[count(num7,7)] do
for a8 := 1 to machnum[count(num8,8)] do
begin
 cunt := cunt+1;
 if cunt=solution[cnt] then
 machine8;
end; end;
9 :
  begin
  num1:=0; num2:=0; num3:=0; num4:=0; num5:=0; num6:=0; num7:=0; num8:=0; num9:=0;
  cunt:=0;
  count(num1,1);
  count(num2,2);
  count(num3,3);
  count(num4,4);
  count(num5,5);
  count(num6,6);
  count(num7,7);
  count(num8,8);
  count(num9,9);
  for a1 := 1 to machnum[count(num1,1)] do
  for a2 := 1 to machnum[count(num2,2)] do
  for a3 := 1 to machnum[count(num3,3)] do
  for a4 := 1 to machnum[count(num4,4)] do
  for a5 := 1 to machnum[count(num5,5)] do
  for a6 := 1 to machnum[count(num6,6)] do
  for a7 := 1 to machnum[count(num7,7)] do
  for a8 := 1 to machnum[count(num8,8)] do
  for a9 := 1 to machnum[count(num9,9)] do
  begin
   cunt := cunt+1;
   if cunt=solution[cnt] then
   machine9;
```

```
end; end;
10 :
begin
  num1:=0; num2:=0; num3:=0; num4:=0; num5:=0; num6:=0; num7:=0; num8:=0; num9:=0;
num10:=0;
  cunt:=0;
  count(num1,1);
  count(num2,2);
  count(num3,3);
  count(num4,4);
  count(num5,5);
  count(num6,6);
  count(num7,7);
  count(num8,8);
  count(num9,9);
  count(num10,10);
  for a1 := 1 to machnum[count(num1,1)] do
  for a2 := 1 to machnum[count(num2,2)] do
  for a3 := 1 to machnum[count(num3,3)] do
  for a4 := 1 to machnum[count(num4,4)] do
  for a5 := 1 to machnum[count(num5,5)] do
  for a6 := 1 to machnum[count(num6,6)] do
  for a7 := 1 to machnum[count(num7,7)] do
  for a8 := 1 to machnum[count(num8,8)] do
  for a9 := 1 to machnum[count(num9,9)] do
  for a10 := 1 to machnum[count(num10,10)] do
begin
  cunt := cunt+1;
  if cunt=solution[cnt] then
  machine10;
end; end;
11 :
begin
  num1:=0; num2:=0; num3:=0; num4:=0; num5:=0; num6:=0; num7:=0; num8:=0; num9:=0;
num10:=0;
  num11:=0; cunt:=0;
  count(num1,1);
  count(num2,2);
```

```
  count(num3,3);
  count(num4,4);
  count(num5,5);
  count(num6,6);
  count(num7,7);
  count(num8,8);
  count(num9,9);
  count(num10,10);
  count(num11,11);
  for a1 := 1 to machnum[count(num1,1)] do
  for a2 := 1 to machnum[count(num2,2)] do
  for a3 := 1 to machnum[count(num3,3)] do
  for a4 := 1 to machnum[count(num4,4)] do
  for a5 := 1 to machnum[count(num5,5)] do
  for a6 := 1 to machnum[count(num6,6)] do
  for a7 := 1 to machnum[count(num7,7)] do
  for a8 := 1 to machnum[count(num8,8)] do
  for a9 := 1 to machnum[count(num9,9)] do
  for a10 := 1 to machnum[count(num10,10)] do
  for a11 := 1 to machnum[count(num11,11)] do
begin
  cunt := cunt+1;
  if cunt=solution[cnt] then
  machine11;
end; end;
12 :
begin
  num1:=0; num2:=0; num3:=0; num4:=0; num5:=0; num6:=0; num7:=0; num8:=0; num9:=0;
num10:=0;
  num11:=0; num12:=0; cunt:=0;
  count(num1,1);
  count(num2,2);
  count(num3,3);
  count(num4,4);
  count(num5,5);
  count(num6,6);
  count(num7,7);
  count(num8,8);
```

```
  count(num9,9);
  count(num10,10);
  count(num11,11);
  count(num12,12);
  for a1 := 1 to machnum[count(num1,1)] do
  for a2 := 1 to machnum[count(num2,2)] do
  for a3 := 1 to machnum[count(num3,3)] do
  for a4 := 1 to machnum[count(num4,4)] do
  for a5 := 1 to machnum[count(num5,5)] do
  for a6 := 1 to machnum[count(num6,6)] do
  for a7 := 1 to machnum[count(num7,7)] do
  for a8 := 1 to machnum[count(num8,8)] do
  for a9 := 1 to machnum[count(num9,9)] do
  for a10 := 1 to machnum[count(num10,10)] do
  for a11 := 1 to machnum[count(num11,11)] do
  for a12 := 1 to machnum[count(num12,12)] do
begin
  cunt := cunt+1;
  if cunt=solution[cnt] then
  machine12;
end; end;
13 :
begin
  num1:=0; num2:=0; num3:=0; num4:=0; num5:=0; num6:=0; num7:=0; num8:=0; num9:=0;
num10:=0;
  num11:=0; num12:=0; num13:=0; cunt:=0;
  count(num1,1);
  count(num2,2);
  count(num3,3);
  count(num4,4);
  count(num5,5);
  count(num6,6);
  count(num7,7);
  count(num8,8);
  count(num9,9);
  count(num10,10);
  count(num11,11);
  count(num12,12);
```

```
  count(num13,13);
  for a1 := 1 to machnum[count(num1,1)] do
  for a2 := 1 to machnum[count(num2,2)] do
  for a3 := 1 to machnum[count(num3,3)] do
  for a4 := 1 to machnum[count(num4,4)] do
  for a5 := 1 to machnum[count(num5,5)] do
  for a6 := 1 to machnum[count(num6,6)] do
  for a7 := 1 to machnum[count(num7,7)] do
  for a8 := 1 to machnum[count(num6,8)] do
  for a9 := 1 to machnum[count(num9,9)] do
  for a10 := 1 to machnum[count(num10,10)] do
  for a11 := 1 to machnum[count(num11,11)] do
  for a12 := 1 to machnum[count(num12,12)] do
  for a13 := 1 to machnum[count(num13,13)] do
begin
  cunt := cunt+1;
  if cunt=solution[cnt] then
  machine13;
end; end;
14 :
begin
  num1:=0; num2:=0; num3:=0; num4:=0; num5:=0; num6:=0; num7:=0; num8:=0; num9:=0;
num10:=0;
  num11:=0; num12:=0; num13:=0; num14:=0; cunt:=0;
  count(num1,1);
  count(num2,2);
  count(num3,3);
  count(num4,4);
  count(num5,5);
  count(num6,6);
  count(num7,7);
  count(num8,8);
  count(num9,9);
  count(num10,10);
  count(num11,11);
  count(num12,12);
  count(num13,13);
  count(num14,14);
```

```
  for a1 := 1 to machnum[count(num1,1)] do
  for a2 := 1 to machnum[count(num2,2)] do
  for a3 := 1 to machnum[count(num3,3)] do
  for a4 := 1 to machnum[count(num4,4)] do
  for a5 := 1 to machnum[count(num5,5)] do
  for a6 := 1 to machnum[count(num6,6)] do
  for a7 := 1 to machnum[count(num7,7)] do
  for a8 := 1 to machnum[count(num8,8)] do
  for a9 := 1 to machnum[count(num9,9)] do
  for a10 := 1 to machnum[count(num10,10)] do
  for a11 := 1 to machnum[count(num11,11)] do
  for a12 := 1 to machnum[count(num12,12)] do
  for a13 := 1 to machnum[count(num13,13)] do
  for a14 := 1 to machnum[count(num14,14)] do
begin
  cunt := cunt+1;
  if cunt=solution[cnt] then
  machine14;
end; end;
15 :
begin
  num1:=0; num2:=0; num3:=0; num4:=0; num5:=0; num6:=0; num7:=0; num8:=0; num9:=0;
num10:=0;
  num11:=0; num12:=0; num13:=0; num14:=0; num15:=0; cunt:=0;
  count(num1,1);
  count(num2,2);
  count(num3,3);
  count(num4,4);
  count(num5,5);
  count(num6,6);
  count(num7,7);
  count(num8,8);
  count(num9,9);
  count(num10,10);
  count(num11,11);
  count(num12,12);
  count(num13,13);
  count(num14,14);
```

```
 count(num15,15);
 for a1 := 1 to machnum[count(num1,1)] do
 for a2 := 1 to machnum[count(num2,2)] do
 for a3 := 1 to machnum[count(num3,3)] do
 for a4 := 1 to machnum[count(num4,4)] do
 for a5 := 1 to machnum[count(num5,5)] do
 for a6 := 1 to machnum[count(num6,6)] do
 for a7 := 1 to machnum[count(num7,7)] do
 for a8 := 1 to machnum[count(num8,8)] do
 for a9 := 1 to machnum[count(num9,9)] do
 for a10 := 1 to machnum[count(num10,10)] do
 for a11 := 1 to machnum[count(num11,11)] do
 for a12 := 1 to machnum[count(num12,12)] do
 for a13 := 1 to machnum[count(num13,13)] do
 for a14 := 1 to machnum[count(num14,14)] do
 for a15 := 1 to machnum[count(num15,15)] do
begin
 cunt := ount+1;
 if cunt=solution[cnt] then
 machine15;
end; end;
16 :
begin
 num1:=0; num2:=0; num3:=0; num4:=0; num5:=0; num6:=0; num7:=0; num8:=0; num9:=0;
num10:=0;
 num11:=0; num12:=0; num13:=0; num14:=0; num15:=0; num16:=0; cunt:=0;
 count(num1,1);
 count(num2,2);
 count(num3,3);
 count(num4,4);
 count(num5,5);
 count(num6,6);
 count(num7,7);
 count(num8,8);
 count(num9,9);
 count(num10,10);
 count(num11,11);
 count(num12,12);
```

```
 count(num13,13);
 count(num14,14);
 count(num15,15);
 count(num16,16);
 for a1 := 1 to machnum[count(num1,1)] do
 for a2 := 1 to machnum[count(num2,2)] do
 for a3 := 1 to machnum[count(num3,3)] do
 for a4 := 1 to machnum[count(num4,4)] do
 for a5 := 1 to machnum[count(num5,5)] do
 for a6 := 1 to machnum[count(num6,6)] do
 for a7 := 1 to machnum[count(num7,7)] do
 for a8 := 1 to machnum[count(num8,8)] do
 for a9 := 1 to machnum[count(num9,9)] do
 for a10 := 1 to machnum[count(num10,10)] do
 for a11 := 1 to machnum[count(num11,11)] do
 for a12 := 1 to machnum[count(num12,12)] do
 for a13 := 1 to machnum[count(num13,13)] do
 for a14 := 1 to machnum[count(num14,14)] do
 for a15 := 1 to machnum[count(num15,15)] do
 for a16 := 1 to machnum[count(num16,16)] do
begin
 cunt := cunt+1;
 if cunt=solution[cnt] then
 machine16;
end; end;
17 :
begin
 num1:=0; num2:=0; num3:=0; num4:=0; num5:=0; num6:=0; num7:=0; num8:=0; num9:=0;
num10:=0;
 num11:=0; num12:=0; num13:=0; num14:=0; num15:=0; num16:=0; num17:=0;
 cunt:=0;
 count(num1,1);
 count(num2,2);
 count(num3,3);
 count(num4,4);
 count(num5,5);
 count(num6,6);
 count(num7,7);
```

```
  count(num8,8);
  count(num9,9);
  count(num10,10);
  count(num11,11);
  count(num12,12);
  count(num13,13);
  count(num14,14);
  count(num15,15);
  count(num16,16);
  count(num17,17);
  for a1 := 1 to machnum[count(num1,1)] do
  for a2 := 1 to machnum[count(num2,2)] do
  for a3 := 1 to machnum[count(num3,3)] do
  for a4 := 1 to machnum[count(num4,4)] do
  for a5 := 1 to machnum[count(num5,5)] do
  for a6 := 1 to machnum[count(num6,6)] do
  for a7 := 1 to machnum[count(num7,7)] do
  for a8 := 1 to machnum[count(num8,8)] do
  for a9 := 1 to machnum[count(num9,9)] do
  for a10 := 1 to machnum[count(num10,10)] do
  for a11 := 1 to machnum[count(num11,11)] do
  for a12 := 1 to machnum[count(num12,12)] do
  for a13 := 1 to machnum[count(num13,13)] do
  for a14 := 1 to machnum[count(num14,14)] do
  for a15 := 1 to machnum[count(num15,15)] do
  for a16 := 1 to machnum[count(num16,16)] do
  for a17 := 1 to machnum[count(num17,17)] do
begin
  cunt := cunt+1;
  if cunt=solution[cnt] then
  machine17;
end; end;
18 :
begin
  num1:=0; num2:=0; num3:=0; num4:=0; num5:=0; num6:=0; num7:=0; num8:=0; num9:=0;
num10:=0;
  num11:=0; num12:=0; num13:=0; num14:=0; num15:=0; num16:=0; num17:=0; num18:=0;
  cunt:=0;
```

```
  count(num1,1);
  count(num2,2);
  count(num3,3);
  count(num4,4);
  count(num5,5);
  count(num6,6);
  count(num7,7);
  count(num8,8);
  count(num9,9);
  count(num10,10);
  count(num11,11);
  count(num12,12);
  count(num13,13);
  count(num14,14);
  count(num15,15);
  count(num16,16);
  count(num17,17);
  count(num18,18);
  for a1 := 1 to machnum[count(num1,1)] do
  for a2 := 1 to machnum[count(num2,2)] do
  for a3 := 1 to machnum[count(num3,3)] do
  for a4 := 1 to machnum[count(num4,4)] do
  for a5 := 1 to machnum[count(num5,5)] do
  for a6 := 1 to machnum[count(num6,6)] do
  for a7 := 1 to machnum[count(num7,7)] do
  for a8 := 1 to machnum[count(num8,8)] do
  for a9 := 1 to machnum[count(num9,9)] do
  for a10 := 1 to machnum[count(num10,10)] do
  for a11 := 1 to machnum[count(num11,11)] do
  for a12 := 1 to machnum[count(num12,12)] do
  for a13 := 1 to machnum[count(num13,13)] do
  for a14 := 1 to machnum[count(num14,14)] do
  for a15 := 1 to machnum[count(num15,15)] do
  for a16 := 1 to machnum[count(num16,16)] do
  for a17 := 1 to machnum[count(num17,17)] do
  for a18 := 1 to machnum[count(num18,18)] do
begin
  cunt := cunt+1;
```

```
  if cunt=solution[cnt] then
  machine18;
end; end;
19 :
begin
  num1:=0; num2:=0; num3:=0; num4:=0; num5:=0; num6:=0; num7:=0; num8:=0; num9:=0;
num10:=0;
  num11:=0; num12:=0; num13:=0; num14:=0; num15:=0; num16:=0; num17:=0; num18:=0;
num19:=0;
  cunt:=0;
  count(num1,1);
  count(num2,2);
  count(num3,3);
  count(num4,4);
  count(num5,5);
  count(num6,6);
  count(num7,7);
  count(num8,8);
  count(num9,9);
  count(num10,10);
  count(num11,11);
  count(num12,12);
  count(num13,13);
  count(num14,14);
  count(num15,15);
  count(num16,16);
  count(num17,17);
  count(num18,18);
  count(num19,19);
  for a1 := 1 to machnum[count(num1,1)] do
  for a2 := 1 to machnum[count(num2,2)] do
  for a3 := 1 to machnum[count(num3,3)] do
  for a4 := 1 to machnum[count(num4,4)] do
  for a5 := 1 to machnum[count(num5,5)] do
  for a6 := 1 to machnum[count(num6,6)] do
  for a7 := 1 to machnum[count(num7,7)] do
  for a8 := 1 to machnum[count(num8,8)] do
  for a9 := 1 to machnum[count(num9,9)] do
```

```
  for a10 := 1 to machnum[count(num10,10)] do
  for a11 := 1 to machnum[count(num11,11)] do
  for a12 := 1 to machnum[count(num12,12)] do
  for a13 := 1 to machnum[count(num13,13)] do
  for a14 := 1 to machnum[count(num14,14)] do
  for a15 := 1 to machnum[count(num15,15)] do
  for a16 := 1 to machnum[count(num16,16)] do
  for a17 := 1 to machnum[count(num17,17)] do
  for a18 := 1 to machnum[count(num18,18)] do
  for a19 := 1 to machnum[count(num19,19)] do
begin
  cunt := cunt+1;
  if cunt=solution[cnt] then
  machine19;
end; end;
20 :
begin
  num1:=0; num2:=0; num3:=0; num4:=0; num5:=0; num6:=0; num7:=0; num8:=0; num9:=0;
num10:=0;
  num11:=0; num12:=0; num13:=0; num14:=0; num15:=0; num16:=0; num17:=0; num18:=0;
num19:=0; num20:=0;
 cunt:=0;
 count(num1,1);
 count(num2,2);
 count(num3,3);
 count(num4,4);
 count(num5,5);
 count(num6,6);
 count(num7,7);
 count(num8,8);
 count(num9,9);
 count(num10,10);
 count(num11,11);
 count(num12,12);
 count(num13,13);
 count(num14,14);
 count(num15,15);
 count(num16,16);
```

```
    count(num17,17);
    count(num18,18);
    count(num19,19);
    count(num20,20);
    for a1 := 1 to machnum[count(num1,1)] do
    for a2 := 1 to machnum[count(num2,2)] do
    for a3 := 1 to machnum[count(num3,3)] do
    for a4 := 1 to machnum[count(num4,4)] do
    for a5 := 1 to machnum[count(num5,5)] do
    for a6 := 1 to machnum[count(num6,6)] do
    for a7 := 1 to machnum[count(num7,7)] do
    for a8 := 1 to machnum[count(num8,8)] do
    for a9 := 1 to machnum[count(num9,9)] do
    for a10 := 1 to machnum[count(num10,10)] do
    for a11 := 1 to machnum[count(num11,11)] do
    for a12 := 1 to machnum[count(num12,12)] do
    for a13 := 1 to machnum[count(num13,13)] do
    for a14 := 1 to machnum[count(num14,14)] do
    for a15 := 1 to machnum[count(num15,15)] do
    for a16 := 1 to maohnum[count(num16,16)] do
    for a17 := 1 to machnum[count(num17,17)] do
    for a18 := 1 to machnum[count(num18,18)] do
    for a19 := 1 to machnum[count(num19,19)] do
    for a20 := 1 to machnum[count(num20,20)] do
  begin
    cunt:=cunt+1;
    if cunt=solution[cnt] then
    machine20;
  end;{for}
  end;{case20}
  end;{case}
  end;
  end;

  Procedure Tabupattern(var fi : fitype; var fi1 : fitypa1);
  begin
    with planrec do
    with oprrec do
```

```
begin
  sum:=0;
  for cnt := 1 to partnum do
begin
  repeat
  randomize;
  l := random(method[cnt]);
  until l > 0;
  sol[cnt] := l;
  cnt1:=1;
  if l <= tabu[cnt,cnt1] then
begin
  solution[cnt]:=l;
  operation;
end{l<tabu}
  else if l > tabu[cnt,cnt1] then
begin
  m:=l-tabu[cnt,cnt1];
  cnt1:=cnt1+1;
  if m <= tabu[cnt,cnt1] then
begin
  l:=m;
  solution[cnt]:=l;
  operation;
end{m<tabu}
  else if m > tabu[cnt,cnt1] then
begin
  n:=m-tabu[cnt,cnt1];
  cnt1:=cnt1+1;
  l:=n;
  solution[cnt]:=l;
  operation;
end;{solution_n}
end;{l>tabu}
end;{for_cnt}
end;{with}
end;{main}
```

```
Procedure Tabupattern2(var fi : fitype; var fi1 : fitype1);
begin
  with planrec do
  with oprrec do
begin
  sum:=0;
  for cnt := 1 to partnum do
begin
  sol[cnt] := solution[cnt];
  l := solution[cnt];
  cnt1:=1;
  if l <= tabu[cnt,cnt1] then
begin
  solution[cnt]:=l;
  operation;
end{l<tabu}
  else if l > tabu[cnt,cnt1] then
begin
  m:=l-tabu[cnt,cnt1];
  cnt1:=cnt1+1;
  if m <= tabu[cnt,cnt1] then
begin
  l:=m;
  solution[cnt]:=l;
  operation;
end{m<tabu}
  else if m > tabu[cnt,cnt1] then
begin
  n:=m-tabu[cnt,cnt1];
  cnt1:=cnt1+1;
  l:=n;
  solution[cnt]:=l;
  operation;
end;{solution_n}
end;{l>tabu}
end;{for_cnt}
end;{with}
end;{main}
```

```
Function swap3(d,g : integer) : longint;
begin
  solve[ran2] := solution[ran2];
  solution[ran2] := solution[ran1];
  solution[ran1] := solve[ran2];
  writeln('iteration ',num);
  tabupattern2(planfi,oprfi);
  for i := 1 to part do
begin
  solution[i] := sol[i];
end;
  if aspiration[d] > sum then
begin
  more := more+1;
  if more = g then more := 1;
  tabupos[d] := 0;
  tabulist[d] := 0;
  aspiration[d] := 9999999;
  tabulist[more] := solution[ran2];
  tabupos[more] := ran2;
  aspiration[more] := sum;
  if sum < minimum then minimum := sum;
end else
begin
end;  end;

Procedure swap(var fi : fitype; var fi1 : fitype1);
var cnt2,cnt3,next,mach1,size    : integer;
    zum,times,newtimes,minimum1,maximum   :  longint;
    ch,key : char;
begin
  clrscr;
  writeln('Neighborhood List :Swap pairwise interchange');
  writeln;
  write('Select Tabu size [3,5,7] ; '); readln(size);
  prepare(planfi);
  prepare1(oprfi);
```

```
  read(fi,planrec);
  read(fi1,oprrec);
  with planrec do
  with oprrec do
 begin
  writeln('N : Enter new seed');
  writeln('O : Use old seed');
  sum:=0;  maximum:=0;  zum;=0;  minimum1:=0;
  part := partnum;
  for cnt := 1 to partnum do
 begin
  newtimes:=0;
  for cnt1 := 1 to plannum[cnt] do
 begin
  times:=1;
  for cnt2 := 1 to oprnum[cnt,cnt1] do
 begin
  next :=0;
  repeat
  next := next+1;
  findword(oprna[cnt,cnt1,cnt2],opname[next]);
  until oprna[cnt,cnt1,cnt2]=opname[next];
  mach1 := machnum[next];{Total plan for part,plan}
  times := times*mach1;
end;{for_cnt2}
  tabu[cnt,cnt1] := times;
  newtimes := newtimes+tabu[cnt,cnt1];
end;{for_cnt1}
  method[cnt] := newtimes;
{  repeat}
  if method[cnt]>maximum then maximum:=method[cnt];
{  until cnt=partnum;}
end;{for_cnt}
  readln(key);
  if key = 'O'  then
begin
  writeln('Iteration 0');
  tabupattern(planfi,oprfi);
```

```
  minimum := sum;
  write('Sequence : ');
  for i := 1 to partnum do
begin
  write(sol[i],',');
end;
  writeln;
  writeln('Makespan : ',sum,'   Best : ',minimum);
end else
begin
  for i := 1 to partnum do
begin
  write('Enter Seed ; '); readln(sol[i]);
end; write('Enter minimum : ');  readln(minimum);  end;
  {SWAP PAIRWISE}
  num:=1;  more:=0;
  for i := 1 to partnum do
begin
  solution[i] := sol[i];
end;
repeat
  repeat
  randomize;
  ran1 := random(partnum);
  until ran1 >0;
  repeat
  randomize;
  ran2 := random(partnum);
  until (ran2>0) and (ran2<>ran1) and (solution[ran1]<=method[ran2])
  and (solution[ran2]<=method[ran1]);
  case size of
  3 : begin
    if (solution[ran1] = tabulist[1]) and (ran2 = tabupos[1]) then
begin
  swap3(1,4);
end else
  if (solution[ran1] = tabulist[2]) and (ran2 = tabupos[2]) then
begin
```

```
  swap3(2,4);
end else
  if (solution[ran1] = tabulist[3]) and (ran2 = tabupos[3]) then
begin
  swap3(3,4);
end else
begin
  solve[ran2] := solution[ran2];
  solution[ran2] := solution[ran1];
  solution[ran1] := solve[ran2];
  writeln('Iteration ',num);
  tabupattern2(planfi,oprfi);
  for l := 1 to partnum do
 begin
  solution[l] := sol[i];
 end;
  more := more+1;
  if more = 4 then more := 1;
  tabulist[more] := solution[ran2];
  tabupos[more] := ran2;
  aspiration[more] := sum;
  if sum < minimum then minimum := sum;
 end;
  minimum1 := minimum-minimum1;
  if minimum1 = 0 then begin minimum1 := minimum; zum := zum+1; end
  else begin minimum1 := minimum; zum:=0; end;
  write('Sequence : ');
  for l := 1 to partnum do
 begin
  write(sol[l],',');
 end;
  writeln;
  writeln('T((i),j)  Makespan   Tabulist');
  write(sol[ran2]:2,ran2:3,sum:9,':6);
  for m := 1 to 3 do
 begin
  write(tabulist[m],',');
 end;
```

```
  writeln;
  writeln('Best : ',minimum);
end;
  5 : begin
  if (solution[ran1] = tabulist[1]) and (ran2 = tabupos[1]) then
begin
  swap3(1,6);
end else
  if (solution[ran1] = tabulist[2]) and (ran2 = tabupos[2]) then
begin
  swap3(2,6);
end else
  if (solution[ran1] = tabulist[3]) and (ran2 = tabupos[3]) then
begin
  swap3(3,6);
end else
  if (solution[ran1] = tabulist[4]) and (ran2 = tabupos[4]) then
begin
  swap3(4,6);
end else
  if (solution[ran1] = tabulist[5]) and (ran2 = tabupos[5]) then
begin
  swap3(5,6);
end else
begin
  solve[ran2] := solution[ran2];
  solution[ran2] := solution[ran1];
  solution[ran1] := solve[ran2];
  writeln('Iteration ',num);
  tabupattern2(planfi,oprfi);
  for i := 1 to partnum do
begin
  solution[i] := sol[i];
end;
  more := more+1;
  if more = 6 then more := 1;
  tabulist[more] := solution[ran2];
  tabupos[more] := ran2;
```

```
 aspiration[more] := sum;
 if sum < minimum then minimum := sum;
end;
 minimum1 := minimum-minimum1;
 if minimum1 = 0 then begin minimum1 := minimum; zum := zum+1; end
 else begin minimum1 := minimum; zum:=0; end;
 write('Sequence : ');
 for l := 1 to partnum do
begin
 write(sol[l],',');
end;
 writeln;
 writeln('T((i),j)  Makespan  Tabulist');
 write(sol[ran2]:2,ran2:3,sum:9,'':6);
 for m := 1 to 5 do
begin
 write(tabulist[m],',');
end;
 writeln;
 writeln('Best : ',minimum);
end;
 7 : begin
 if (solution[ran1] = tabulist[1]) and (ran2 = tabupos[1]) then
begin
 swap3(1,8);
end else
 if (solution[ran1] = tabulist[2]) and (ran2 = tabupos[2]) then
begin
 swap3(2,8);
end else
 if (solution[ran1] = tabulist[3]) and (ran2 = tabupos[3]) then
begin
 swap3(3,8);
end else
 if (solution[ran1] = tabulist[4]) and (ran2 = tabupos[4]) then
begin
 swap3(4,8);
end else
```

```
 If (solution[ran1] = tabulist[5]) and (ran2 = tabupos[5]) then
begin
  swap3(5,8);
end else
  If (solution[ran1] = tabulist[6]) and (ran2 = tabupos[6]) then
begin
  swap3(6,8);
end else
  If (solution[ran1] = tabulist[7]) and (ran2 = tabupos[7]) then
begin
  swap3(7,8);
end else
begin
  solve[ran2] := solution[ran2];
  solution[ran2] := solution[ran1];
  solution[ran1] := solve[ran2];
  writeln('Iteration ',num);
  tabupattern2(planfl,oprfi);
  for I := 1 to partnum do
begin
  solution[I] := sol[I];
end;
  more := more+1;
  If more = 8 then more := 1;
  tabulist[more] := solution[ran2];
  tabupos[more] := ran2;
  aspiration[more] := sum;
  If sum < minimum then minimum := sum;
end;
  minimum1 := minimum-minimum1;
  If minimum1 = 0 then begin minimum1 := minimum; zum := zum+1; end
  else begin minimum1 := minimum; zum:=0; end;
  write('Sequence : ');
  for I := 1 to partnum do
begin
  write(sol[I],',');
end;
  writeln;
```

```
 writeln('T((i),j)  Makespan   Tabulist');
 write(sol[ran2]:2,ran2:3,sum:9,'':6);
 for m := 1 to 7 do
begin
 write(tabulist[m],',');
end;
 writeln;
 writeln('Best : ',minimum);
end  else exit; end;
 {if num mod 20 = 0 then ch := readkey;}
 num := num+1;
 until (num=1001) or (zum=10);
end;{with_opr}
end;{main}

Procedure Tabupattern3(var fi : fitype; var fi1 : fitype1);
begin
 with planrec do
 with oprrec do
begin
 sum:=0;
 for ont := 1 to partnum do
begin
 repeat
 randomize;
 i := random(method[cnt]);
 until i > 0;
 cnt1:=1;
 sol[cnt] := i;
 if i <= tabu[cnt,cnt1] then
begin
 solution[cnt]:=i;
 operation;
end{i<tabu}
 else if i > tabu[cnt,cnt1] then
begin
 m:=i-tabu[cnt,cnt1];
 cnt1:=cnt1+1;
```

```
 if m <= tabu[cnt,cnt1] then
begin
 l:=m;
 solution[cnt]:=l;
 operation;
end{m<tabu}
 else if m > tabu[cnt,cnt1] then
begin
 n:=m-tabu[cnt,cnt1];
 cnt1:=cnt1+1;
 l:=n;
 solution[cnt]:=l;
 operation;
end;{solution_n}
end;{l>tabu}
end;{for_cnt}
end;{with}
end;{main}

Procedure Tabupattern4(var fl : fltype; var fl1 : fltype1);
begin
  with planrec do
  with oprrec do
begin
  sum:=0;
  for cnt := 1 to partnum do
begin
  sol[cnt] := solution[cnt];
  l := solution[cnt];
  cnt1:=1;
  if l <= tabu[cnt,cnt1] then
begin
  solution[cnt]:=l;
  operation;
end{l<tabu}
  else if l > tabu[cnt,cnt1] then
begin
  m:=l-tabu[cnt,cnt1];
```

```
  cnt1:=cnt1+1;
  if m <= tabu[cnt,cnt1] then
begin
  l:=m;
  solution[cnt]:=l;
  operation;
end{m<tabu}
  else if m > tabu[cnt,cnt1] then
begin
  n:=m-tabu[cnt,cnt1];
  cnt1:=cnt1+1;
  l:=n;
  solution[cnt]:=l;
  operation;
end;{solution_n}
end;{l>tabu}
end;{for_cnt}
end;{with}
end;{main}

Function insert3(e,f : integer) : longint;
var  j,tem : longint;
begin
  with planrec do
  with oprrec do
begin
  for m := 1 to partnum do
begin
  tempo[m] := solution[m];
end;
  tempo[1] := tempo[ran1];
  l := ran1;
  l := l-1;
  if l=0 then
begin
end  else
begin
  for j := 1 to l do
```

```
begin
  tempo[j+1] := solution[j];
end;
end;{else}
  count1 := 1;
{  if (count1=ran1) or {count1=ran1-1) or (count1<>ran2) then
begin  writeln('Iteration ',num);
end else}
  if (ran1>=3) and (ran2=1) then
begin
  for m := 1 to partnum do
begin
  solution[m] := tempo[m];
end;
  writeln('Iteration ',num);
  tabupattern4(planfi,oprfi);
  for l := 1 to partnum do
begin
  solution[l] := sol[l];
end;
  if aspiration[e] > sum then
begin
  more := more+1;
  if more = f then more := 1;
  tabulist[e] := 0;
  aspiration[e] := 999999;
  tabulist[more] := tabu1;
  aspiration[more] := sum;
  if sum < minimum then minimum := sum;
end  else
begin  end; end;
  for l := 1 to partnum-1 do
begin
  count1 := count1+1;
  if (count1=ran1) or (count1=ran1-1) or (count1<>ran2) then
begin
  tem := tempo[i+1];
  tempo[i+1] := tempo[i];
```

```
  tempo[i] := tem;
 end else
 begin
  tem := tempo[i+1];
  tempo[i+1] := tempo[i];
  tempo[i] := tem;
  for m := 1 to partnum do
 begin
  solution[m] := tempo[m];
 end;
  writeln('iteration ',num);
  tabupattern4(planfi,oprfi);
  for i := 1 to partnum do
 begin
  solution[i] := sol[i];
 end;
  if aspiration[e] > sum then
 begin
  more := more+1;
  if more = f then more := 1;
  tabulist[e] := 0;
  aspiration[e] := 999999;
  tabulist[more] := tabu1;
  aspiration[more] := sum;
  if sum < minimum then minimum := sum;
 end else begin  end; end; end; end; end;

Function insert4(g,h : integer) : longint;
var  j,tem : longint;
begin
  with planrec do
  with oprrec do
 begin
  for m := 1 to partnum do
 begin
  tempo[m] := solution[m];
 end;
  tempo[1] := tempo[ran1];
```

```
 l := ran1;
 l := l-1;
 if l=0 then
begin
end  else
begin
  for j := 1 to l do
begin
  tempo[j+1] := solution[j];
end;
end;{else}
  count1 := 1;
{  if (count1=ran1) or (count1=ran1-1) or (count1<>ran2) then
begin
end else}
  if (ran1>=3) and (ran2=1) then
begin
  for m := 1 to partnum do
begin
  solution[m] := tempo[m];
end;
  writeln('Iteration ',num);
  tabupattern4(planfi,oprfi);
  for l := 1 to partnum do
begin
  solution[l] := sol[l];
end;
  more := more+1;
  if more = h then more := 1;
  tabulist[more] := tabu1;
  aspiration[more] := sum;
  if sum < minimum then minimum := sum;
end;
  for i := 1 to partnum-1 do
begin
  count1 := count1+1;
  if (count1=ran1) or (count1=ran1-1) or (count1<>ran2) then
begin
```

```
    tem := tempo[i+1];
    tempo[i+1] := tempo[i];
    tempo[i] := tem;
  end else
  begin
    tem := tempo[i+1];
    tempo[i+1] := tempo[i];
    tempo[i] := tem;
    for m := 1 to partnum do
  begin
    solution[m] := tempo[m];
  end;
    writeln('Iteration ',num);
    tabupattern4(planfi,oprfi);
    for l := 1 to partnum do
  begin
    solution[l] := sol[l];
  end;
    more := more+1;
    if more = h then more := 1;
    tabulist[more] := tabu1;
    aspiration[more] := sum;
    if sum < minimum then minimum := sum;
end; end; end; end;

Procedure Insertion(var fi : fitype; var fi1 : fitype1);
var j,cnt2,next,mach1,size1      : integer;
    times,newtimes,tem,maximum   : longint;
    key1 : char;
begin
  clrscr;
  writeln('Neighborhood List :Insertion Interchange');
  writeln;
  write('Select Tabu size [3,5,7] ; '); readln(size1);
  prepare(planfi);
  prepare1(oprfi);
  read(fi,planrec);
  read(fi1,oprrec);
```

```
  with planrec do
  with oprrec do
begin
  writeln('N: Enter new seed');
  writeln('O: Use old seed');
  sum:=0;  maximum:=0;  zum1:=0;  minimum2:=0;
  for cnt := 1 to partnum do
begin
  newtimes:=0;
  for cnt1 := 1 to plannum[cnt] do
begin
  times:=1;
  for cnt2 := 1 to oprnum[cnt,cnt1] do
begin
  next :=0;
  repeat
  next := next+1;
  findword(oprna[cnt,cnt1,cnt2],oprname[next]);
  until oprna[cnt,cnt1,cnt2]=oprname[next];
  mach1 := machnum[next];{Total plan for part,plan}
  times := times*mach1;
end;{for_cnt2}
  tabu[cnt,cnt1] := times;
  newtimes := newtimes+tabu[cnt,cnt1];
end;{for_cnt1}
  method[cnt] := newtimes;
{  repeat}
  if method[cnt]>maximum then maximum:=method[cnt];
{  until cnt=partnum;}
end;{for_cnt}
  readln(key1);
  if key1 = 'O' then
begin
  writeln('Iteration 0');
  tabupattern3(planfi,oprfi);
  minimum := sum;
  write('Sequence : ');
  for i := 1 to partnum do
```

```
begin
  write(sol[i],',');
end;
  writeln;
  writeln('Makespan : ',sum,'   Best : ',minimum);
end else
begin
  for l := 1 to partnum do
begin
  write('Enter seed : ');
  readln(sol[l]);
end;
  write('Enter minimum : ');
  readln(minimum);
end;
  {INSERTION}
  num:=1;  more:=0;
  for l := 1 to partnum do
begin
  solution[l] := sol[l];
end;
repeat
  repeat
  randomize;
  ran1 := random(partnum);
  until ran1 >0;
  repeat
  randomize;
  ran2 := random(partnum);
  until (ran2>0) and (ran2<>ran1) and (ran2<>ran1-1) and (solution[ran1]<=method[ran2])
  and (solution[ran2]<=method[ran1]);
  tabu1 := solution[ran1];
  case size1 of
  3 : begin
  if tabulist[1] = solution[ran1] then
begin
  insert3(1,4);
end else
```

```
 if tabulist[2] = solution[ran1] then
begin
 insert3(2,4);
end else
 if tabulist[3] = solution[ran1] then
begin
 insert3(3,4);
end else
begin
 insert4(3,4);
end;
 minimum2 := minimum-minimum2;
 if minimum2=0 then begin minimum2 := minimum;  zum1 := zum1+1; end
 else begin minimum2 := minimum;  zum1:=0; end;
 write('Sequence : ');
 for i := 1 to partnum do
begin
 write(sol[i],',');
end;
 writeln;
 writeln('T(i)    Makespan   Tabulist');
 write(tabu1:2,sum:12,':6);
 for j := 1 to 3 do
begin
 write(tabulist[j],',');
end;
 writeln;
 writeln('Best : ',minimum);
end;
 5 : begin
 if tabulist[1] = solution[ran1] then
begin
 insert3(1,6);
end else
 if tabulist[2] = solution[ran1] then
begin
 insert3(2,6);
end else
```

```
  if tabulist[3] = solution[ran1] then
begin
  insert3(3,6);
end else
  if tabulist[4] = solution[ran1] then
begin
  insert3(4,6);
end else
  if tabulist[5] = solution[ran1] then
begin
  insert3(5,6);
end else
begin
  insert4(5,6);
end;
  minimum2 := minimum-minimum2;
  if minimum2=0 then begin minimum2 := minimum;  zum1 := zum1+1; end
  else begin minimum2 := minimum;  zum1:=0; end;
  write('Sequence : ');
  for I := 1 to partnum do
begin
  write(sol[I],',');
end;
  writeln;
  writeln('T(i)    Makespan   Tabulist');
  write(tabu1:2,sum:12,'':6);
  for J := 1 to 5 do
begin
  write(tabulist[J],',');
end;
  writeln;
  writeln('Best : ',minimum); end;
  7 : begin
  if tabulist[1] = solution[ran1] then
begin
  insert3(1,8);
end else
  if tabulist[2] = solution[ran1] then
```

```
begin
  insert3(2,8);
end else
  if tabulist[3] = solution[ran1] then
begin
  insert3(3,8);
end else
  if tabulist[4] = solution[ran1] then
begin
  insert3(4,8);
end else
  if tabulist[5] = solution[ran1] then
begin
  insert3(5,8);
end else
  if tabulist[6] = solution[ran1] then
begin
  insert3(6,8);
end else
  if tabulist[7] = solution[ran1] then
begin
  insert3(7,8);
end else
begin
  insert4(7,8);
end;
  minimum2 := minimum-minimum2;
  if minimum2=0 then begin minimum2 := minimum;  zum1 := zum1+1; end
  else begin minimum2 := minimum;  zum1:=0; end;
  write('Sequence : ');
  for i := 1 to partnum do
begin
  write(sol[i],',');
end;
  writeln;
  writeln('T(i)    Makespan   Tabulist');
  write(tabu1:2,sum:12,':8);
  for j := 1 to 7 do
```

```
begin
  write(tabulist[j],',');
end;
  writeln;
  writeln('Best : ',minimum);
  end; else exit end;
  num := num+1;
  until (num = 1001) or (zum1 = 10);
end;{with_opr}
end;{main}

Procedure Tabusearch;
var ch2 : char;
begin
  clrscr;
  writeln;  writeln;
  writeln('Type of Tabu Search ');
  writeln('S : Swap Pairwise ');
  writeln('I : Insertion ');
  writeln('Q : Quit ');
  writeln;
  write('Select type of Search : ');
  repeat ch2 := upcase(readkey) until ch2 in ['S','I','Q'];
      case ch2 of
       'S' : Swap(planfi,oprfi);
       'I' : Insertion(planfi,oprfi);
       'Q' : begin clrscr; Exit; end;
      end;
end;


Procedure Displsy(var fi : fitype);
var num1,num2,num3 : integer;
begin
  prepare(planfi);
  read(fi,planrec);
  with planrec do
begin
```

```
 clrscr;
 writeln('Part      Inventory Cost      Plan      Operation');
 for num1 := 1 to partnum do
 for num2 := 1 to plannum[num1] do
 for num3 := 1 to oprnum[num1,num2] do
 writeln(partna[num1],'      ',invcost[num1],'      ',planna[num1,num2],'      ',oprna
[num1,num2,num3]);
end;
end;

Procedure Display1(var fi1 : fitype1);
var num4,num5 : integer;
begin
  prepare1(oprfi);
  read(fi1,oprrec);
  with oprrec do
begin
  clrscr;
  writeln('Operation  Machine  Machine Cost  Setup Cost  Production Cost');
  for num4 := 1 to oprnumm do
  for num5 := 1 to machnum[num4] do
  writeln('  ',oprname[num4],'      ',machna[num4,num5],'      ',machcost[num4,num5]
  ,'      ',setcost[num4,num5],'      ',prodcost[num4,num5]);
end;
end;

Procedure getoption(var ch1 : char);
begin
  writeln;  writeln;
  writeln('E : Enter Part ');
  writeln('O : Enter Operation ');
  writeln('D : Display Part ');
  writeln('K : Display Operation ');
  writeln('T : Tabu Search ');
  writeln('Q : Quit ');
  writeln; write('Please Select  ');
  repeat ch1 := upcase(readkey) until ch1 in ['E','D','K','O','T','Q'];
  writeln(ch1);
```

```
end;

var sel : char;
begin
  clrscr;
  repeat
    getoption(sel);
    case sel of
      'E' : Enpart;
      'O' : Enopr(oprfi);
      'D' : Display(planfi);
      'K' : Display1(oprfi);
      'T' : Tabusearch;
    end;
  until sel = 'Q'
end.
```

ประวัติผู้เขียน

นายประสิทธิ์ สวราชย์ เกิดเมื่อวันที่ 29 เมษายน 2513 ที่จังหวัดกรุงเทพฯ สำเร็จการศึกษาปริญญาตรี วิทยาศาสตรบัณฑิต สาขาฟิสิกส์อุตสาหกรรมและอุปกรณ์การแพทย์ สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ หลังจากนั้นได้เข้าทำงานที่บริษัท ออโตเมชั่นเซอร์วิส จำกัด และปัจจุบันทำงานที่บริษัท บีทีมิตรแลนด์ จำกัด ได้เข้าศึกษาต่อหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต สาขาวิศวกรรมอุตสาหการ จุฬาลงกรณ์มหาวิทยาลัยในปีการศึกษา 2538