#include <vcl.h> #include <stdlib.h> #include <stdio.h> #include <math.h> #include <vector.h> #include <string.h> #include <algorithm> #include <fstream.h> #include <iostream.h> #include <sstream.h> #pragma hdrstop #pragma argsused

vector<vector<int> > R,R1,R2,Riter,Rbest; //rotaların tanımlanması vector<double> Rt\_u, R1t\_u, R2t\_u, Ritert\_u, Rbestt\_u; vector<int> Rctrl, R1ctrl, R2ctrl, Riterctrl, Rbestctrl;

int TS,DS,kontrol,D1,D2; // dağıtım ve toplama sayıları

double XK,YK,ZK; //konteynır boyutlarının tanımlanması

double KK; //konteynır kapasitesinin tanımlanması

vector<double> depouzaklik; //Depo ile noktalar arasi uzaklık

vector<vector<double> > uzaklik; //uzaklık matrisinin tanımlanması

vector<int> Ttopla\_say; vector<vector<double> > TX,TY,TZ;

vector<int> Ddagit\_say; vector<vector<double> > DX,DY,DZ;

void TasarrufAlgorithm() { int i,j,k1,index,deger,pos,dene; vector<vector<double> > tasarruf, tmsf; vector<int> dagitim; vector<double> kapa; tasarruf.resize(3); for(i=0;i<DS;i++) dagitim.push\_back(i); for(i=0;i<DS-1;i++) { for(j=i+1;j<DS;j++) { kontrol=i; tasarruf[0].push\_back(i); kontrol=j; tasarruf[1].push\_back(j); kontrol=depouzaklik[i]+depouzaklik[j]uzaklik[i][j]; tasarruf[2].push\_back(depouzaklik[i]+depouzaklik[j]-uzaklik[i][j]); } } tmsf.resize(3,vector<double>(tasarruf[0].size())); tmsf=tasarruf; R.push\_back();

kapa.push\_back(KK); while(dagitim.size()!=0) { i=distance(tasarruf[2].begin(),max\_element(tasarruf[2].begin(),tasarruf[2 ].end())); k1=0; for(j=0;j<int(R.size());j++) { if(find(R[j].begin(),R[j].end(),tasarruf[0][i])!=R[j].end()) { index=j; k1=k1+1; deger=tasarruf[1][i]; pos=find(R[j].begin(),R[j].end(),tasarruf[0][i])-R[j].begin(); if(pos!=0 && pos!=int(R[j].size())) pos=pos+1; //&& ) //else if(pos== } if(find(R[j].begin(),R[j].end(),tasarruf[1][i])!=R[j].end()) { index=j; k1=k1+1; deger=tasarruf[0][i]; pos=find(R[j].begin(),R[j].end(),tasarruf[0][i])-R[j].begin(); if(pos!=0 && pos!=int(R[j].size())) pos=pos-1; //) } } if(k1==1) { if(Ddagit\_say[deger]<kapa[index]) { if(pos==int(R[j].size()-1)) R[index].push\_back(deger); else R[index].insert(R[index].begin()+pos,deger); tasarruf[2][i]=-10000; dagitim.erase(find(dagitim.begin(),dagitim.end(),deger)); kapa[index]=kapa[index]Ddagit\_say[deger]; } else { if(int(R.size())==1) { R.push\_back(); kapa.push\_back(KK); index=R.size()-1; if((Ddagit\_say[deger]<kapa[index])) { R[index].push\_back(deger);

tasarruf[2][i]=-10000; dagitim.erase(find(dagitim.begin(),dagitim.end(),deger)); kapa[index]=kapa[index]Ddagit\_say[deger]; } } else tasarruf[2][i]=-10000; } } else if(k1==0) { index=R.size()-1; if(Ddagit\_say[tasarruf[0][i]]+Ddagit\_say[tasarruf[1][i]]<=kapa[index] && R[index].size()==0) { //index=R.size()-1; R[index].push\_back(tasarruf[0][i]); R[index].push\_back(tasarruf[1][i]); tasarruf[2][i]=-10000; dagitim.erase(find(dagitim.begin(),dagitim.end(),tasarruf[0][i])); dagitim.erase(find(dagitim.begin(),dagitim.end(),tasarruf[1][i])); kapa[index]=kapa[index](Ddagit\_say[tasarruf[0][i]]+Ddagit\_say[tasarruf[1][i]]); } else tasarruf[2][i]=-10000; } else if(k1>1) tasarruf[2][i]=-10000; if(\*max\_element(tasarruf[2].begin(),tasarruf[2].end())==10000 && dagitim.size()>0) { R.push\_back(); kapa.push\_back(KK); tasarruf=tmsf; } } tasarruf.clear(); dagitim.clear(); kapa.clear(); }

void ONEINS() { int i,j,k,index1,index2,index3; double AYM,AYM1; vector<int> toplama; vector<double> A;

for(i=0;i<DS && Ttopla\_say[i]>0;i++) toplama.push\_back(DS+i);

A.resize(R.size(),0); Rt\_u.resize(R.size()); for(i=0;i<int(R.size());i++) {

Rt\_u[i]=depouzaklik[(R[i][0]<DS)\*R[i][0]+(R[i][0]>=DS)\*(R[i][0]DS)]+depouzaklik[(R[i][R[i].size()-1]<DS)\*R[i][R[i].size()1]+(R[i][R[i].size()-1]>=DS)\*(R[i][R[i].size()-1]-DS)]; for(j=0;j<int(R[i].size()-1);j++) { if(R[i][j+1]!=R[i][j]-DS && R[i][j]!=R[i][j+1]DS) Rt\_u[i]=Rt\_u[i]+uzaklik[(R[i][j]<DS)\*R[i][j]+(R[i][j]>=DS)\*(R[i][j]DS)][(R[i][j+1]<DS)\*R[i][j+1]+(R[i][j+1]>=DS)\*(R[i][j+1]-DS)]; } kontrol=Rt\_u[i]; }//Rota Uzunlukları hesaplandı (Depolar dahil) while(toplama.size()!=0) { AYM1=INT\_MAX; for(k=0;k<int(toplama.size());k++) { for(i=0;i<int(R.size());i++) { //RU[i].resize(R[i].alttur.size()+1); for(j=0;j<int(R[i].size());j++) { if(j==0) { } else if(j<int(R[i].size())) { if(R[i][j]>=DS && R[i][j1]>=DS) { A[i]=A[i]+Ttopla\_say[R[i][j]-DS]; } else if(R[i][j]>=DS) { A[i]=A[i]+Ttopla\_say[R[i][j]-DS]; } else if(R[i][j-1]>=DS) { A[i]=A[i]+Ddagit\_say[R[i][j]]; } else { A[i]=A[i]+Ddagit\_say[R[i][j]]; } } } kontrol=A[i]; } //Her a ve b arası rota uzunlukları ve A değerleri hesaplandi... for(i=0;i<int(R.size());i++) { for(j=0;j<int(R[i].size()+1);j++) { kontrol=R[i].size();

if(j==0) { //MANTIKSAL İŞLEM OLUYOR!!! AYM=depouzaklik[toplama[k]-DS]+uzaklik[toplama[k]DS][((R[i][j]>=DS)\*(R[i][j]-DS)+(R[i][j]<DS)\*(R[i][j]))]depouzaklik[((R[i][j]>=DS)\*(R[i][j]-DS)+(R[i][j]<DS)\*(R[i][j]))]double(1.5\*depouzaklik[toplama[k]-DS])+A[i]\*double((Rt\_u[i]/KK)); } else if(j<int(R[i].size())) { AYM=uzaklik[((R[i][j1]>=DS)\*(R[i][j-1]-DS)+(R[i][j-1]<DS)\*(R[i][j-1]))][toplama[k]DS]+uzaklik[toplama[k]-DS][((R[i][j]>=DS)\*(R[i][j]DS)+(R[i][j]<DS)\*(R[i][j]))]-uzaklik[((R[i][j-1]>=DS)\*(R[i][j-1]DS)+(R[i][j-1]<DS)\*(R[i][j-1]))][((R[i][j]>=DS)\*(R[i][j]DS)+(R[i][j]<DS)\*(R[i][j]))]-double(1.5\*depouzaklik[toplama[k]DS])+A[i]\*double((Rt\_u[i]/KK)); } else { AYM=depouzaklik[toplama[k]-DS]+uzaklik[((R[i][j-1]>=DS)\*(R[i][j-1]DS)+(R[i][j-1]<DS)\*(R[i][j-1]))][toplama[k]-DS]-depouzaklik[((R[i][j1]>=DS)\*(R[i][j-1]-DS)+(R[i][j-1]<DS)\*(R[i][j-1]))]double(1.5\*depouzaklik[toplama[k]-DS])+A[i]\*double((Rt\_u[i]/KK)); } if(AYM<=AYM1) { AYM1=AYM; index1=i; index2=j; index3=k; } if(j<int(R[i].size()-1) && A[i]!=0) { A[i]=A[i]((R[i][j+1]>=DS)\*(Ttopla\_say[R[i][j+1]DS])+(R[i][j+1]<DS)\*(Ddagit\_say[R[i][j+1]])); kontrol=A[i]; } } } } if(index2<int(R[index1].size())) { R[index1].insert(R[index1].begin()+index2,toplama[index3]); } else R[index1].push\_back(toplama[index3]); toplama.erase(toplama.begin()+index3); } toplama.clear(); A.clear(); }

void KYP() {

int i,j,k,l,m,count1,indis,pos; double XKY,YKY,ZKY, y; vector<double> y1; vector<vector<double> > KT, yer, sira; vector<int> ref; yer.resize(3); sira.resize(3); for(i=0;i<int(R.size());i++) { //İLK DEPODA BULUNUP KONTROL YAPILACAK YER... XKY=XK; YKY=YK; ZKY=ZK; for(j=0;j<int(R[i].size());j++) { if(R[i][j]<DS) { for(k=0;k<int(DX[R[i][j]].size());k++) { kontrol=DX[R[i][j]][k]; yer[0].push\_back(DX[R[i][j]][k]); kontrol=DY[R[i][j]][k]; yer[1].push\_back(DY[R[i][j]][k]); kontrol=DZ[R[i][j]][k]; yer[2].push\_back(DZ[R[i][j]][k]); } } } while(yer[0].size()!=0) { j=max\_element(yer[0].begin(),yer[0].end())yer[0].begin(); if(sira[0].size()==0) { kontrol=yer[0][j]; sira[0].push\_back(yer[0][j]); kontrol=yer[1][j]; sira[1].push\_back(yer[1][j]); kontrol=yer[2][j]; sira[2].push\_back(yer[2][j]); } else { kontrol=sira[0][sira[0].size()-1]; kontrol=yer[0][j]; if(find(sira[0].begin(),sira[0].end(),yer[0][j])!=sira[0].end()) { k=find(sira[0].begin(),sira[0].end(),yer[0][j])-sira[0].begin(); count1=count(sira[0].begin(),sira[0].end(),yer[0][j]); for(l=0;l<count1;l++) { kontrol=sira[1][k+l]; kontrol=yer[1][j]; if(sira[1][k+l]<yer[1][j]) { pos=k+l+1; }

else if(sira[1][k+l]>yer[1][j]) { pos=k+l; l=count1; } else { if(sira[2][k+l]<=yer[2][j]) { pos=k+l+1; } else { pos=k+l; l=count1; } } } if(pos==int(sira[0].size())) { kontrol=yer[0][j]; sira[0].push\_back(yer[0][j]); kontrol=yer[1][j]; sira[1].push\_back(yer[1][j]); kontrol=yer[2][j]; sira[2].push\_back(yer[2][j]); } else { sira[0].insert(sira[0].begin()+pos,yer[0][j]); sira[1].insert(sira[1].begin()+pos,yer[1][j]); sira[2].insert(sira[2].begin()+pos,yer[2][j]); kontrol=yer[0][j]; kontrol=yer[1][j]; kontrol=yer[2][j]; } } else { sira[0].push\_back(yer[0][j]); sira[1].push\_back(yer[1][j]); sira[2].push\_back(yer[2][j]); } } yer[0].erase(yer[0].begin()+j); yer[1].erase(yer[1].begin()+j); yer[2].erase(yer[2].begin()+j); } //KUTU YERLEŞTİRME BAŞLIYOR...

indis=0; while(indis==0) { YKY=YK; ZKY=ZK; KT.clear(); for(j=0;j<int(sira[0].size());j++) { if(KT.size()==0) { XKY=XKY-sira[0][j]; y=sira[0][j]; y1.push\_back(y); kontrol=YKY-sira[1][j]; KT.push\_back(); KT[0].push\_back(YKY-sira[1][j]); KT[0].push\_back(0); kontrol=ZKY-sira[2][j]; KT[0].push\_back(ZKY-sira[2][j]); kontrol=sira[0][j]; KT[0].push\_back(sira[0][j]); kontrol=sira[1][j]; KT[0].push\_back(sira[1][j]); kontrol=sira[2][j]; KT[0].push\_back(sira[2][j]); ref.push\_back(-1); YKY=YKY-sira[1][j]; sira[0][j]=-1; sira[1][j]=-1; sira[2][j]=-1; } else { for(k=0;k<int(KT.size());k++) { if(KT[k].size()<7) { if(sira[1][j]<=KT[k][0] && sira[2][j]<=ZKY) { kontrol=KT[k][0]-sira[1][j]; KT.push\_back(); KT[KT.size()-1].push\_back(KT[k][0]-sira[1][j]); kontrol=y1[k]; kontrol=y1[k]-sira[0][j]; KT[KT.size()-1].push\_back(y1[k]-sira[0][j]); kontrol=ZKY-sira[2][j]; KT[KT.size()-1].push\_back(ZKY-sira[2][j]); kontrol=sira[0][j]; KT[KT.size()-1].push\_back(sira[0][j]); kontrol=sira[1][j];

KT[KT.size()-1].push\_back(sira[1][j]); kontrol=sira[2][j]; KT[KT.size()-1].push\_back(sira[2][j]); ref.push\_back(k); y1.push\_back(y); KT[k][0]=0; sira[0][j]=-1; sira[1][j]=-1; sira[2][j]=-1; k=KT.size(); } else if(sira[0][j]<=KT[k][1] && sira[2][j]<=ZKY && sira[1][j]<=KT[k][4]) { kontrol=KT[k][0]+KT[k][4]-sira[1][j]; KT.push\_back(); KT[KT.size()-1].push\_back(KT[k][0]+KT[k][4]-sira[1][j]); kontrol=KT[k][1]-sira[0][j]; KT[KT.size()-1].push\_back(KT[k][1]-sira[0][j]); kontrol=ZKY-sira[2][j]; KT[KT.size()-1].push\_back(ZKY-sira[2][j]); kontrol=sira[0][j]; KT[KT.size()-1].push\_back(sira[0][j]); kontrol=sira[1][j]; KT[KT.size()-1].push\_back(sira[1][j]); kontrol=sira[2][j]; KT[KT.size()-1].push\_back(sira[2][j]); kontrol=KT[k][1]; y1.push\_back(KT[k][1]); kontrol=KT[k][4]; ref.push\_back(-1); KT[k][1]=0;

sira[0][j]=-1; sira[1][j]=-1; sira[2][j]=-1; k=KT.size(); } else if(sira[2][j]<=KT[k][2] && sira[1][j]<=KT[k][4]) { kontrol=KT[k][4]-sira[1][j]; KT.push\_back(); KT[KT.size()-1].push\_back(KT[k][4]-sira[1][j]); kontrol=KT[k][3]-sira[0][j]; KT[KT.size()-1].push\_back(KT[k][3]-sira[0][j]); kontrol=KT[k][2]-sira[2][j]; KT[KT.size()-1].push\_back(KT[k][2]-sira[2][j]); kontrol=sira[0][j]; KT[KT.size()-1].push\_back(sira[0][j]); kontrol=sira[1][j]; KT[KT.size()-1].push\_back(sira[1][j]); kontrol=sira[2][j]; KT[KT.size()-1].push\_back(sira[2][j]); KT[KT.size()-1].push\_back(k); KT[k][2]=0; kontrol=KT[k][3]; y1.push\_back(KT[k][3]); ref.push\_back(-1); sira[0][j]=-1; sira[1][j]=-1; sira[2][j]=-1; k=KT.size(); } } else {

if(sira[1][j]<=KT[k][0] && sira[2][j]<=KT[k][2]) { kontrol=KT[k][0]-sira[1][j]; KT.push\_back(); KT[KT.size()-1].push\_back(KT[k][0]-sira[1][j]); kontrol=y1[k]; kontrol=y1[k]-sira[0][j]; KT[KT.size()-1].push\_back(y1[k]-sira[0][j]); kontrol=KT[KT[k][6]][2]-sira[2][j]; KT[KT.size()-1].push\_back(KT[KT[k][6]][2]-sira[2][j]); kontrol=sira[0][j]; KT[KT.size()-1].push\_back(sira[0][j]); kontrol=sira[1][j]; KT[KT.size()-1].push\_back(sira[1][j]); kontrol=sira[2][j]; KT[KT.size()-1].push\_back(sira[2][j]); KT[KT.size()-1].push\_back(KT[k][6]); KT[k][0]=0; ref.push\_back(k); kontrol=KT[KT[k][6]][3]; y1.push\_back(KT[KT[k][6]][3]); sira[0][j]=-1; sira[1][j]=-1; sira[2][j]=-1; k=KT.size(); } else if(sira[0][j]<=KT[k][1] && sira[2][j]<=KT[k][5]+KT[k][2] && sira[1][j]<=KT[KT[k][6]][4]) { kontrol=KT[KT[k][6]][4]-sira[1][j]; KT.push\_back(); KT[KT.size()-1].push\_back(KT[KT[k][6]][4]-sira[1][j]); kontrol=KT[k][1]-sira[0][j];

KT[KT.size()-1].push\_back(KT[k][1]-sira[0][j]); kontrol=KT[KT[k][6]][2]-sira[2][j]; KT[KT.size()-1].push\_back(KT[KT[k][6]][2]-sira[2][j]); kontrol=sira[0][j]; KT[KT.size()-1].push\_back(sira[0][j]); kontrol=sira[1][j]; KT[KT.size()-1].push\_back(sira[1][j]); kontrol=sira[2][j]; KT[KT.size()-1].push\_back(sira[2][j]); KT[KT.size()-1].push\_back(KT[k][6]); kontrol=KT[k][1]; y1.push\_back(KT[k][1]); ref.push\_back(-1); KT[k][1]=0; sira[0][j]=-1; sira[1][j]=-1; sira[2][j]=-1; k=KT.size(); } else if(sira[2][j]<=KT[k][2] && sira[1][j]<=KT[k][4]) { kontrol=KT[k][4]-sira[1][j]; KT.push\_back(); KT[KT.size()-1].push\_back(KT[k][4]-sira[1][j]); kontrol=KT[k][3]-sira[0][j]; KT[KT.size()-1].push\_back(KT[k][3]-sira[0][j]); kontrol=KT[k][2]-sira[2][j]; KT[KT.size()-1].push\_back(KT[k][2]-sira[2][j]); kontrol=sira[0][j]; KT[KT.size()-1].push\_back(sira[0][j]); kontrol=sira[1][j];

KT[KT.size()-1].push\_back(sira[1][j]); kontrol=sira[2][j]; KT[KT.size()-1].push\_back(sira[2][j]); KT[KT.size()-1].push\_back(k); kontrol=KT[k][3]; y1.push\_back(KT[k][3]); ref.push\_back(-1); KT[k][2]=0; sira[0][j]=-1; sira[1][j]=-1; sira[2][j]=-1; k=KT.size(); } } } } } if(sira[0].size()>0) { if(XKY<sira[0][max\_element(sira[0].begin(),sira[0].end())sira[0].begin()]) { indis=indis+1; kontrol=0; Rctrl[i]=0; i=R.size(); } while(\*min\_element(sira[0].begin(),sira[0].end())==-1 && sira[0].size()>0) { j=find(sira[0].begin(),sira[0].end(),-1)-sira[0].begin(); sira[0].erase(sira[0].begin()+j); sira[1].erase(sira[1].begin()+j); sira[2].erase(sira[2].begin()+j); } } else { indis=indis+1; kontrol=1; Rctrl[i]=1; } } }

for(i=0;i<int(R.size());i++) { for(j=0;j<int(R[i].size()) && Rctrl[i]==1;j++) { //BURADA KONTROLE BAŞLIYORUZ... XKY=XK; YKY=YK; ZKY=ZK; yer.clear(); yer.resize(3); sira.clear(); sira.resize(3); if(R[i][j]>=DS) { for(l=0;l<int(TX[R[i][j]-DS].size());l++) { kontrol=TX[R[i][j]-DS][l]; yer[0].push\_back(TX[R[i][j]-DS][l]); kontrol=TY[R[i][j]-DS][l]; yer[1].push\_back(TY[R[i][j]-DS][l]); kontrol=TZ[R[i][j]-DS][l]; yer[2].push\_back(TZ[R[i][j]-DS][l]); } for(k=j+1;k<int(R[i].size());k++) { if(R[i][k]<DS && (R[i][j]-DS!=R[i][k] || k!=j+1)) { for(l=0;l<int(DX[R[i][k]].size());l++) { kontrol=DX[R[i][k]][l]; yer[0].push\_back(DX[R[i][k]][l]); kontrol=DY[R[i][k]][l]; yer[1].push\_back(DY[R[i][k]][l]); kontrol=DZ[R[i][k]][l]; yer[2].push\_back(DZ[R[i][k]][l]); } } } for(k=j-1;k>=0;k--) { if(R[i][k]>=DS) { for(l=0;l<int(TX[R[i][k]DS].size());l++) { kontrol=TX[R[i][k]-DS][l]; yer[0].push\_back(TX[R[i][k]-DS][l]); kontrol=TY[R[i][k]-DS][l]; yer[1].push\_back(TY[R[i][k]-DS][l]); kontrol=TZ[R[i][k]-DS][l];

yer[2].push\_back(TZ[R[i][k]-DS][l]); } } }

while(yer[0].size()!=0) { k=max\_element(yer[0].begin(),yer[0].end())-yer[0].begin(); if(sira[0].size()==0) { sira[0].push\_back(yer[0][k]); sira[1].push\_back(yer[1][k]); sira[2].push\_back(yer[2][k]); } else { if(find(sira[0].begin(),sira[0].end(),yer[0][k])!=sira[0].end()) { l=find(sira[0].begin(),sira[0].end(),yer[0][k])-sira[0].begin(); count1=count(sira[0].begin(),sira[0].end(),yer[0][k]); for(m=0;m<count1;m++) {

if(sira[1][l+m]<yer[1][k]) { pos=l+m+1; } else if(sira[1][l+m]>yer[1][k]) { pos=m+l; m=count1; } else { if(sira[2][l+m]<=yer[2][k]) { pos=l+m+1; } else { pos=l+m; m=count1; } } } if(pos==int(sira[0].size())) { kontrol=yer[0][k]; sira[0].push\_back(yer[0][k]);

kontrol=yer[1][k]; sira[1].push\_back(yer[1][k]); kontrol=yer[2][k]; sira[2].push\_back(yer[2][k]); } else { sira[0].insert(sira[0].begin()+pos,yer[0][k]); sira[1].insert(sira[1].begin()+pos,yer[1][k]); sira[2].insert(sira[2].begin()+pos,yer[2][k]); kontrol=yer[0][k]; kontrol=yer[1][k]; kontrol=yer[2][k]; } } else { sira[0].push\_back(yer[0][k]); sira[1].push\_back(yer[1][k]); sira[2].push\_back(yer[2][k]); } } yer[0].erase(yer[0].begin()+k); yer[1].erase(yer[1].begin()+k); yer[2].erase(yer[2].begin()+k); } //KUTU YERLEŞTİRME BAŞLIYOR... indis=0; while(indis==0) { YKY=YK; ZKY=ZK; KT.clear(); y1.clear(); ref.clear(); for(k=0;k<int(sira[0].size());k++) { if(KT.size()==0) { XKY=XKY-sira[0][k]; y=sira[0][k]; y1.push\_back(y); kontrol=YKY-sira[1][k]; KT.push\_back(); KT[0].push\_back(YKY-sira[1][k]); KT[0].push\_back(0); kontrol=ZKY-sira[2][k]; KT[0].push\_back(ZKYsira[2][k]);

KT[0].push\_back(sira[0][k]); KT[0].push\_back(sira[1][k]); KT[0].push\_back(sira[2][k]); ref.push\_back(-1); YKY=YKY-sira[1][k]; sira[0][k]=-1; sira[1][k]=-1; sira[2][k]=-1; } else { for(l=0;l<int(KT.size());l++) { if(KT[l].size()<7) { if(sira[1][k]<=KT[l][0] && sira[2][k]<=ZKY) { kontrol=KT[l][0]-sira[1][k]; KT.push\_back(); KT[KT.size()-1].push\_back(KT[l][0]-sira[1][k]); kontrol=y1[l]-sira[0][k]; KT[KT.size()-1].push\_back(y1[l]-sira[0][k]); kontrol=ZKY-sira[2][k]; KT[KT.size()-1].push\_back(ZKY-sira[2][k]); kontrol=sira[0][k]; KT[KT.size()-1].push\_back(sira[0][k]); kontrol=sira[1][k]; KT[KT.size()-1].push\_back(sira[1][k]); kontrol=sira[2][k]; KT[KT.size()-1].push\_back(sira[2][k]); ref.push\_back(l); y1.push\_back(y); KT[l][0]=0; sira[0][k]=-1; sira[1][k]=-1;

sira[2][k]=-1; l=KT.size(); } else if(sira[0][k]<=KT[l][1] && sira[2][k]<=ZKY && sira[1][k]<=KT[l][4]) { kontrol=KT[l][0]+KT[l][4]-sira[1][k]; KT.push\_back(); KT[KT.size()-1].push\_back(KT[l][0]+KT[l][4]-sira[1][k]); kontrol=KT[l][1]-sira[0][k]; KT[KT.size()-1].push\_back(KT[l][1]-sira[0][k]); kontrol=ZKY-sira[2][k]; KT[KT.size()-1].push\_back(ZKY-sira[2][k]); kontrol=sira[0][k]; KT[KT.size()-1].push\_back(sira[0][k]); kontrol=sira[1][k]; KT[KT.size()-1].push\_back(sira[1][k]); kontrol=sira[2][k]; KT[KT.size()-1].push\_back(sira[2][k]); kontrol=KT[l][1]; y1.push\_back(KT[l][1]); ref.push\_back(-1); KT[l][1]=0; sira[0][k]=-1; sira[1][k]=-1; sira[2][k]=-1; l=KT.size(); } else if(sira[2][k]<=KT[l][2] && sira[1][k]<=KT[l][4]) { kontrol=KT[l][4]-sira[1][k]; KT.push\_back(); KT[KT.size()-1].push\_back(KT[l][4]-sira[1][k]);

kontrol=KT[l][3]-sira[0][k]; KT[KT.size()-1].push\_back(KT[l][3]-sira[0][k]); kontrol=KT[l][2]-sira[2][k]; KT[KT.size()-1].push\_back(KT[l][2]-sira[2][k]); kontrol=sira[0][k]; KT[KT.size()-1].push\_back(sira[0][k]); kontrol=sira[1][k]; KT[KT.size()-1].push\_back(sira[1][k]); kontrol=sira[2][k]; KT[KT.size()-1].push\_back(sira[2][k]); KT[KT.size()-1].push\_back(l); kontrol=KT[l][3]; y1.push\_back(KT[l][3]); ref.push\_back(-1); KT[l][2]=0; sira[0][k]=-1; sira[1][k]=-1; sira[2][k]=-1; l=KT.size(); } } else { kontrol=KT[l][6]; if(sira[1][k]<=KT[l][0] && sira[2][k]<=KT[l][2]) { kontrol=KT[l][0]-sira[1][k]; KT.push\_back(); KT[KT.size()-1].push\_back(KT[l][0]-sira[1][k]); kontrol=y1[l]; kontrol=y1[l]-sira[0][k]; KT[KT.size()-1].push\_back(y1[l]-sira[0][k]); kontrol=KT[KT[l][6]][2]-sira[2][k];

KT[KT.size()-1].push\_back(KT[KT[l][6]][2]-sira[2][k]); kontrol=sira[0][k]; KT[KT.size()-1].push\_back(sira[0][k]); kontrol=sira[1][k]; KT[KT.size()-1].push\_back(sira[1][k]); kontrol=sira[2][k]; KT[KT.size()-1].push\_back(sira[2][k]); KT[KT.size()-1].push\_back(KT[l][6]); KT[l][0]=0; ref.push\_back(l); kontrol=KT[KT[l][6]][3]; y1.push\_back(KT[KT[l][6]][3]); sira[0][k]=-1; sira[1][k]=-1; sira[2][k]=-1; l=KT.size(); } else if(sira[0][k]<=KT[l][1] && sira[2][k]<=KT[l][5]+KT[l][2] && sira[1][k]<=KT[KT[l][6]][4]) { kontrol=KT[KT[l][6]][4]-sira[1][k]; KT.push\_back(); KT[KT.size()-1].push\_back(KT[KT[l][6]][4]-sira[1][k]); kontrol=KT[l][1]-sira[0][k]; KT[KT.size()-1].push\_back(KT[l][1]-sira[0][k]); kontrol=KT[KT[l][6]][2]-sira[2][k]; KT[KT.size()-1].push\_back(KT[KT[l][6]][2]-sira[2][k]); kontrol=sira[0][k]; KT[KT.size()-1].push\_back(sira[0][k]); kontrol=sira[1][k]; KT[KT.size()-1].push\_back(sira[1][k]); kontrol=sira[2][k]; KT[KT.size()-1].push\_back(sira[2][k]); KT[KT.size()-1].push\_back(KT[l][6]); y1.push\_back(KT[l][1]); ref.push\_back(-1); KT[l][1]=0; sira[0][k]=-1; sira[1][k]=-1; sira[2][k]=-1; l=KT.size(); } else if(sira[2][k]<=KT[l][2] && sira[1][k]<=KT[l][4]) { kontrol=KT[l][4]-sira[1][k]; KT.push\_back(); KT[KT.size()-1].push\_back(KT[l][4]-sira[1][k]); kontrol=KT[l][3]-sira[0][k]; KT[KT.size()-1].push\_back(KT[l][3]-sira[0][k]); kontrol=KT[l][2]-sira[2][k]; KT[KT.size()-1].push\_back(KT[l][2]-sira[2][k]); kontrol=sira[0][k]; KT[KT.size()-1].push\_back(sira[0][k]); kontrol=sira[1][k]; KT[KT.size()-1].push\_back(sira[1][k]); kontrol=sira[2][k]; KT[KT.size()-1].push\_back(sira[2][k]); KT[KT.size()-1].push\_back(l); kontrol=KT[l][3]; y1.push\_back(KT[l][3]); ref.push\_back(-1); KT[l][2]=0; sira[0][k]=-1;

sira[1][k]=-1; sira[2][k]=-1; l=KT.size(); } } } } } if(sira[0].size()>0) { if(XKY<sira[0][max\_element(sira[0].begin(),sira[0].end())sira[0].begin()]) { indis=indis+1; kontrol=0; Rctrl[i]=0; i=R.size(); } while(\*min\_element(sira[0].begin(),sira[0].end())==-1 && sira[0].size()>0) { k=find(sira[0].begin(),sira[0].end(),-1)-sira[0].begin(); sira[0].erase(sira[0].begin()+k); sira[1].erase(sira[1].begin()+k); sira[2].erase(sira[2].begin()+k); } } else { indis=indis+1; kontrol=1; Rctrl[i]=1; } } } } } yer.clear(); sira.clear(); KT.clear(); }

int main(int argc, char\* argv[]) { int i1,j1,k1,l1,i2,j2,i3,i4,sum,STS,LTS,LTF,TK; double x1,y1,z1,TM,TM1,TM2,TMiter,TMbest,TMiter2; string str; vector<vector<double> > STabuList, LTabuList; vector<vector<vector<int> > > T2,T1; vector<vector<double> > T2t\_u, T1t\_u; vector<vector<int> > T2ctrl, T1ctrl;

vector<int> K; ifstream in; ofstream out;

out.precision(10); //Parametrelerin Okutulması in.open("C:\\KDTARP\\parametre.txt"); out.open("C:\\KDTARP\\sonuc.txt"); if(!in.is\_open()) { cout<<"PROBLEM DOSYASI OKUNAMADI!!!!..."<<endl; system("EXIT"); } else cout<<"PROBLEM DOSYASI OKUNUYOR.....";

while(!in.eof()) { in>>str>>D1; in>>str>>D2; in>>str>>STS; in>>str>>LTS; in>>str>>LTF; in>>str; while(str!="Toplama\_Miktarları") { in>>str; stringstream convert(str); convert>>i2; if(str!="Toplama\_Miktarları") Ddagit\_say.push\_back(i2); } str.clear(); DS=Ddagit\_say.size(); Ttopla\_say.resize(DS); for(i1=0;i1<DS;i1++) in>>Ttopla\_say[i1]; in>>str; for(i1=0;i1<DS+1;i1++) in>>str; depouzaklik.resize(DS); in>>str>>str; for(i1=0;i1<DS;i1++) in>>depouzaklik[i1]; uzaklik.resize(DS, vector<double>(DS)); for(i1=0;i1<DS;i1++) { in>>str>>str; for(i2=0;i2<DS;i2++) { in>>uzaklik[i1][i2]; } } in>>str>>str>>str>>str; in>>KK>>XK>>YK>>ZK; in>>str>>str>>str>>str>>i1; DX.resize(DS); DY.resize(DS); DZ.resize(DS); while(str!="Toplama\_Kutu\_Müşteri\_No") { in>>x1>>y1>>z1; DX[i1-1].push\_back(x1); DY[i1-1].push\_back(y1);

DZ[i1-1].push\_back(z1); in>>str; stringstream convert(str); convert>>i1; } str.clear(); in>>str>>str>>str; TX.resize(DS); TY.resize(DS); TZ.resize(DS); while(!in.eof()) { in>>i1>>x1>>y1>>z1; TX[i1-1].push\_back(x1); TY[i1-1].push\_back(y1); TZ[i1-1].push\_back(z1); } } in.close(); cout<<"BITTI!!!"<<endl<<endl; cout<<"BASLANGIC COZUMU DERLENIYOR....."; //Tasarruf Algoritması \*/ TasarrufAlgorithm();

kontrol=R.size(); R1t\_u.resize(R.size()); R2t\_u.resize(R.size()); Ritert\_u.resize(R.size()); Rbestt\_u.resize(R.size()); Rctrl.resize(R.size()); R1ctrl.resize(R.size()); R2ctrl.resize(R.size()); Riterctrl.resize(R.size()); Rbestctrl.resize(R.size()); //1-INS algoritması ONEINS();

for(i1=0;i1<int(R.size());i1++) { Rt\_u[i1]=depouzaklik[(R[i1][0]<DS)\*R[i1][0]+(R[i1][0]>=DS)\*(R[i1][0]DS)]+depouzaklik[(R[i1][R[i1].size()-1]<DS)\*R[i1][R[i1].size()1]+(R[i1][R[i1].size()-1]>=DS)\*(R[i1][R[i1].size()-1]-DS)]; for(j1=0;j1<int(R[i1].size()-1);j1++) { if(R[i1][j1+1]!=R[i1][j1]-DS && R[i1][j1]!=R[i1][j1+1]-DS) Rt\_u[i1]=Rt\_u[i1]+uzaklik[(R[i1][j1]<DS)\*R[i1][j1]+(R[i1][j1]>=DS)\*(R[i1] [j1]-DS)][(R[i1][j1+1]<DS)\*R[i1][j1+1]+(R[i1][j1+1]>=DS)\*(R[i1][j1+1]DS)]; } kontrol=Rt\_u[i1]; } KYP(); cout<<"BITTI!!!"<<endl<<endl; //TABUYA GİRME ZAMANI....... sum=0; for(i1=0;i1<int(R.size()-1);i1++) { for(i2=i1+1;i2<int(R.size());i2++) sum=sum+R[i1].size()\*R[i2].size(); } if(D1>sum+1) {

cout<<"DIS DONGU DEGERI "<<sum+1<<" DEN BUYUK OLAMAZ!!!!"<<endl; system("EXIT"); } cout<<"TABU CALISIYOR....."; T1.resize(sum+1); T1t\_u.resize(sum+1); T1[0].resize(R.size()); T1t\_u[0].resize(R.size()); R1.resize(R.size()); R2.resize(R.size()); Riter.resize(R.size()); Rbest.resize(R.size()); K.resize(R.size(),0); for(i1=0;i1<int(R.size());i1++) { i2=R[i1].size(); R1[i1].resize(i2); R2[i1].resize(i2); Riter[i1].resize(i2); Rbest[i1].resize(i2); T1[0][i1].resize(i2); T1[0][i1]=R[i1]; T1t\_u[0][i1]=Rt\_u[i1];

} Riter=R; Rbest=R; R1=R; R2=R; TM=0; for(i1=0;i1<int(R.size());i1++) { TM=TM+Rt\_u[i1]; } TMbest=TM; TMiter=TM; sum=1; for(i1=R.size()-1;i1>0;i1--) { for(i2=i1-1;i2>=0;i2--) { for(j1=0;j1<int(R[i2].size());j1++) { for(j2=0;j2<int(R[i1].size());j2++) { T1[sum].resize(R.size()); T1t\_u[sum].resize(R.size()); for(k1=0;k1<int(T1[sum].size());k1++) T1[sum][k1].resize(R[k1].size()); T1[sum]=R1; swap(T1[sum][i1][j2],T1[sum][i2][j1]); for(k1=0;k1<int(T1[sum].size());k1++) { kontrol=T1[sum][k1][0]; kontrol=T1[sum][k1][T1[sum][k1].size()-1]; T1t\_u[sum][k1]=depouzaklik[(T1[sum][k1][0]<DS)\*T1[sum][k1][0]+(T1[sum][k1 ][0]>=DS)\*(T1[sum][k1][0]DS)]+depouzaklik[(T1[sum][k1][T1[sum][k1].size()-

1]<DS)\*T1[sum][k1][T1[sum][k1].size()-1]+(T1[sum][k1][T1[sum][k1].size()1]>=DS)\*(T1[sum][k1][T1[sum][k1].size()-1]-DS)]; for(l1=0;l1<int(T1[sum][k1].size()-1);l1++) { if(T1[sum][k1][l1]!=T1[sum][k1][l1+1]-DS && T1[sum][k1][l1+1]!=T1[sum][k1][l1]-DS) T1t\_u[sum][k1]=T1t\_u[sum][k1]+uzaklik[(T1[sum][k1][l1]<DS)\*T1[sum][k1][l1 ]+(T1[sum][k1][l1]>=DS)\*(T1[sum][k1][l1]DS)][(T1[sum][k1][l1+1]<DS)\*T1[sum][k1][l1+1]+(T1[sum][k1][l1+1]>=DS)\*(T1 [sum][k1][l1+1]-DS)]; } //Rota Uzunlukları hesaplandı (Depolar dahil) } sum+=1; } } } }

if(R1.size()==1) D1=1; for(i1=0;i1<D1;i1++) { out<<i1<<'\t'; Riter=T1[i1]; TMiter=0; for(j1=0;j1<int(T1t\_u[i1].size());j1++) TMiter=TMiter+T1t\_u[i1][j1]; for(i2=0;i2<D2;i2++) { if(i2!=0) out<<'\t'<<i2<<'\t'; else out<<i2<<'\t'; R2=Riter; sum=1; for(j1=0;j1<int(R2.size());j1++) { for(j2=0;j2<int(R2[j1].size()-1);j2++) { out<<R2[j1][j2]+1<<"-"; } out<<R2[j1][j2]+1<<'\t'; } out<<TMiter<<'\t'; T2.clear(); T2t\_u.clear(); T2.resize(R.size()); T2t\_u.resize(R.size()); for(j1=0;j1<int(R.size());j1++) { T2[j1].resize(R[j1].size()+1); T2t\_u[j1].resize(R[j1].size()+1); for(j2=0;j2<int(R[j1].size());j2++) { T2[j1][j2].resize(R[j1].size()); if(j2==0) T2[j1][j2]=R2[j1]; else { T2[j1][j2]=R2[j1];

swap(T2[j1][j2][j21],T2[j1][j2][j2]); } } T2[j1][j2].resize(R[j1].size()); T2[j1][j2]=R2[j1]; T2[j1][j2].insert(T2[j1][j2].begin()+1,T2[j1][j2][j2-1]); T2[j1][j2].erase(T2[j1][j2].begin()+j2); sum=sum\*(R[j1].size()+1); for(j2=0;j2<int(R[j1].size()+1);j2++) { T2t\_u[j1][j2]=depouzaklik[(T2[j1][j2][0]<DS)\*T2[j1][j2][0]+(T2[j1][j2][0] >=DS)\*(T2[j1][j2][0]-DS)]+depouzaklik[(T2[j1][j2][T2[j1][j2].size()1]<DS)\*T2[j1][j2][T2[j1][j2].size()-1]+(T2[j1][j2][T2[j1][j2].size()1]>=DS)\*(T2[j1][j2][T2[j1][j2].size()-1]-DS)]; for(k1=0;k1<int(T2[j1][j2].size()-1);k1++) { if(T2[j1][j2][k1]!=T2[j1][j2][k1+1]-DS && T2[j1][j2][k1+1]!=T2[j1][j2][k1]-DS) T2t\_u[j1][j2]=T2t\_u[j1][j2]+uzaklik[(T2[j1][j2][k1]<DS)\*T2[j1][j2][k1]+(T 2[j1][j2][k1]>=DS)\*(T2[j1][j2][k1]DS)][(T2[j1][j2][k1+1]<DS)\*T2[j1][j2][k1+1]+(T2[j1][j2][k1+1]>=DS)\*(T2[j1 ][j2][k1+1]-DS)]; } }//Rota Uzunlukları hesaplandı (Depolar dahil) } TK=1; TM1=INT\_MAX; TM2=INT\_MAX; TMiter2=0; for(j1=1;j1<sum;j1++) { TM=0; out<<j1<<'\t'; kontrol=R.size(); for(j2=0;j2<int(R.size());j2++) { R[j2]=T2[j2][K[j2]]; if(j2==int(R.size()-1) && j1==1) { R[j2]=T2[j2][1]; TM=TM+T2t\_u[j2][1]; } else {

TM=TM+T2t\_u[j2][K[j2]]; } } KYP(); int i5=0; for(j2=0;j2<int(R.size());j2++) { i3=0;

for(k1=0;k1<int(R[j2].size()1);k1++) { l1=R[j2][k1]; out<<R[j2][k1]+1<<"-"; kontrol=0; for(l1=k1+1;l1<int(R[j2].size());l1++) { if(R[j2][l1]<DS) kontrol=kontrol+(Ddagit\_say[R[j2][l1]]); } for(l1=k1;l1>=0;l1--) { if(R[j2][l1]>=DS) kontrol=kontrol+(Ttopla\_say[R[j2][l1]-DS]); } if(kontrol<KK && i3==0) { i3=0; i5=0; } else { i3+=1; i5+=1; } } l1=R[j2][k1]; out<<R[j2][k1]+1<<'\t'; if(i3!=0) out<<"Hayır"<<'\t'; else out<<"Evet"<<'\t'; if(Rctrl[j2]==0) out<<"Hayır"<<'\t'; else out<<"Evet"<<'\t'; } i4=0; for(j2=0;j2<int(R.size());j2++) { if(Rctrl[j2]==1 && i4==0) i4=0; else i4+=1; } kontrol=0; for(j2=0;j2<int(R.size());j2++) { for(k1=0;k1<int(R[j2].size()1);k1++) { for(l1=0;l1<int(STabuList.size());l1++) { if(R[j2][k1]==STabuList[l1][0] && R[j2][k1+1]==STabuList[l1][1]) { kontrol+=1; if(STabuList[l1][2]<LTF) STabuList[l1][2]+=1;

} } for(l1=0;l1<int(LTabuList.size());l1++) { if(R[j2][k1]==LTabuList[l1][0] && R[j2][k1+1]==LTabuList[l1][1]) { kontrol+=1; } } } } if(TM<TMiter && i5==0 && i4==0) { if(kontrol==0) { Riter=R; TMiter=TM; TK=0; } } else if(TM>TMiter && TM<TM2 && i5==0 && i4==0) { if(kontrol==0) { R1=R; TM1=TM; TM2=TM; } } else if(TM==TMiter && i5==0 && i4==0 && TMiter2==0) { if(kontrol==0) { Riter=R; TMiter=TM; TK=0; TMiter2=1; } } if(kontrol==0) out<<"Hayır"<<'\t'; else out<<"Evet"<<'\t'; for(j2=0;j2<int(R.size());j2++) { if(j2==int(R.size()-1) && j1==1) out<<T2t\_u[j2][1]<<'\t'; else out<<T2t\_u[j2][K[j2]]<<'\t'; } if(R.size()!=1) out<<TM<<'\t'; if(TM<TMbest && i5==0 && i4==0) { Rbest=R; TMbest=TM; Riter=R; TMiter=TM;

} if(j1==1) K[K.size()-1]+=2; else K[K.size()-1]+=1; for(j2=R.size()-1;j2>0;j2--) { if(K[j2]==int(R[j2].size())+1) { K[j2]=0; K[j2-1]+=1; } } out<<TMbest<<'\t'; if(K[0]==int(R[0].size())+1) K[0]=0; for(k1=0;k1<int(STabuList.size());k1++) out<<"("<<STabuList[k1][0]+1<<","<<STabuList[k1][1]+1<<")"<<","; out<<'\t'; for(k1=0;k1<int(LTabuList.size());k1++) out<<"("<<LTabuList[k1][0]+1<<","<<LTabuList[k1][1]+1<<")"<<","; out<<'\n'; out<<'\t'<<'\t'; for(j2=0;j2<int(R.size());j2++) { if(R.size()>1) out<<'\n'; for(k1=0;k1<int(R[j2].size());k1++) { if(R[j2][k1]>9) out<<" "; else out<<" "; } } out<<'\t'<<'\t'; } if(TK==1 && TM1<INT\_MAX) { Riter=R1; TMiter=TM1; } for(j1=0;j1<int(R.size());j1++) { for(j2=0;j2<int(R[j1].size());j2++) { if(R2[j1][j2]!=Riter[j1][j2]) { if(Riter[j1][j2]==R2[j1][R2[j1].size()-1] && j2==1) { for(k1=0,i3=0;k1<int(STabuList.size());k1++) { if(STabuList[k1][0]==Riter[j1][j2] && STabuList[k1][1]==R2[j1][0]) { i3+=1; } } if(i3==0)

{ STabuList.push\_back(); STabuList[STabuList.size()-1].resize(4); STabuList[STabuList.size()-1][0]=Riter[j1][j2]; STabuList[STabuList.size()-1][1]=R2[j1][0]; STabuList[STabuList.size()-1][2]=1; j2=R[j1].size(); } } else { for(k1=0,i3=0;k1<int(STabuList.size());k1++) { if(STabuList[k1][0]==R2[j1][j2] && STabuList[k1][1]==Riter[j1][j2]) { i3+=1; } } if(i3==0) { STabuList.push\_back(); STabuList[STabuList.size()-1].resize(4); STabuList[STabuList.size()-1][0]=R2[j1][j2]; STabuList[STabuList.size()-1][1]=Riter[j1][j2]; STabuList[STabuList.size()-1][2]=1; j2=R[j1].size(); } } } } } for(l1=0;l1<int(STabuList.size());l1++) { kontrol=0; for(k1=0;k1<int(LTabuList.size());k1++) {

if(LTabuList[k1][0]==STabuList[l1][0] && LTabuList[k1][1]==STabuList[l1][1]) { STabuList[l1][3]=1; } else { if(STabuList[l1][3]!=1) STabuList[l1][3]=0; } } } for(k1=0;k1<int(STabuList.size());k1++) { if(STabuList[k1][2]==LTF && STabuList[k1][3]==0) { LTabuList.push\_back(); LTabuList[LTabuList.size()-1].resize(2); LTabuList[LTabuList.size()-1][0]=STabuList[k1][0]; LTabuList[LTabuList.size()1][1]=STabuList[k1][1]; } } if(int(STabuList.size())>STS) { for(k1=0;k1<int(STabuList.size()-1);k1++) { STabuList[k1]=STabuList[k1+1]; } STabuList.erase(STabuList.begin()+(STabuList.size()-1)); } kontrol=LTabuList.size(); if(int(LTabuList.size())>LTS) { for(k1=0;k1<int(LTabuList.size()-1);k1++) { LTabuList[k1]=LTabuList[k1+1]; } LTabuList.erase(LTabuList.begin()+(LTabuList.size()-1)); } out<<'\n'; } } out.close(); K.clear(); R.clear(); R1.clear(); R2.clear(); Riter.clear(); Rbest.clear(); cout<<"BITTI!!!!!!!"<<endl<<endl;; system("PAUSE"); return 0;

}