**DOĞRUSAL PORGRAMLAMA DERSİ PROJE ÖDEVİ**

#include<stdio.h>

#include<stdlib.h>

#include<iostream>

#include<cmath>

#include<string.h>

**using** **namespace** std**;**

**int** main**(){**

**char** tip**[**4**];** *//Problemin tipini (max/min) tutar*

cout**<<"problem tipi (max or min): ";**

cin**>>**tip**;**

**int** N**;** *//değişken sayısı*

**int** m**;** *// kısıt sayısı*

cout**<<"\ndegisken sayisini giriniz: ";**

cin**>>**N**;**

cout**<<**endl**;**

**double** X**[**N**][**1**];** *//değişken değerlerinin optimal değerleri tutar*

**for(int** i**;** i**<**N**;** i**++){**X**[**i**][**0**]=**0**;}**

**char** ozx**[**N**];** *// değişkenlerin <0, >0 veya sınırsız olma özelliklerini tutar*

**for(int** i**=**0**;** i**<**N**;** i**++){**

cout**<<**i**+**1**<<". degisken sifirdan kucuk ise k, buyuk ise b, sinirsiz ise s harfini giriniz: ";**

cin**>>**ozx**[**i**];**

**}**

cout**<<**endl**;**

**double** C**[**1**][**N**];** *// amaç fonksiyonu katsayıları*

**for(int** i**=**0**;** i**<**N**;** i**++)**

**{**

cout**<<**i**+**1**<<". degiskenin amac fonksiyonundaki katsayisini giriniz: ";**

cin**>>**C**[**0**][**i**];**

**}**

cout**<<"\nKisit sayisini giriniz:";**

cin**>>**m**;**

cout**<<**endl**;**

**double** anamatris**[**m**][**N**];**

**for(int** i**=**0**;**i**<**m**;**i**++)**

**{**

**for(int** j**=**0**;**j**<**N**;**j**++)**

**{**

cout**<<**i**+**1**<<". kisitin "<<**j**+**1**<<". degiskenin katsayisini giriniz:";**

cin**>>**anamatris**[**i**][**j**];**

**}**

**}**

**int** n**=**N**;** *// gerekiyor....*

**for(int** i**=**0**;** i**<**n**;** i**++)**

**{**

**if(**ozx**[**i**]==**'s'**){**n**=**n**+**1**;}**

**}**

**double** matris**[**m**][**n**];** *//Değişkenlerden herhangi birinin <0 veya sınırsız olması durumunda oluşan yeni matris*

**double** c**[**1**][**n**];** *//Değişkenlerden herhangi birinin <0 veya sınırsız olması durumunda oluşan yeni amaç fonksiyon katsayıları*

**for(int** i**=**0**;** i**<**m**;** i**++)**

**{**

**for(int** j**=**0**;** j**<**n**;** j**++)**

**{**

matris**[**i**][**j**]=**0**;**

**}**

**}**

**int** y**=**0**;**

**for(int** i**=**0**;** i**<**n**;**i**++)**

**{**

**if(**ozx**[**i**-**y**]==**'k'**)**

**{**

c**[**0**][**i**]=(-**1**)\***C**[**0**][**i**-**y**];**

**for(int** j**=**0**;** j**<**m**;** j**++)**

**{**

matris**[**j**][**i**]=(-**1**)\***anamatris**[**j**][**i**-**y**];**

**}**

**}**

**if(**ozx**[**i**-**y**]==**'b'**)**

**{**

c**[**0**][**i**]=**C**[**0**][**i**-**y**];**

**for(int** j**=**0**;** j**<**m**;** j**++)**

**{**

matris**[**j**][**i**]=**anamatris**[**j**][**i**-**y**];**

**}**

**}**

**if(**ozx**[**i**-**y**]==**'s'**)**

**{**

c**[**0**][**i**]=**C**[**0**][**i**-**y**];**

c**[**0**][**i**+**1**]=(-**1**)\***C**[**0**][**i**-**y**];**

**for(int** j**=**0**;** j**<**m**;** j**++)**

**{**

matris**[**j**][**i**]=**anamatris**[**j**][**i**-**y**];**

matris**[**j**][**i**+**1**]=(-**1**)\***anamatris**[**j**][**i**-**y**];**

**}**

i**=**i**+**1**;**

y**=**y**+**1**;**

**}**

**}**

**for(int** i**=**0**;** i**<**N**;** i**++)**

**{**

**if(**ozx**[**i**]==**'s' **||** ozx**[**i**]==**'k'**)**

**{**

cout**<<**endl**<<"\n Güncellenen matris: \n";**

**for(int** i**=**0**;** i**<**m**;** i**++)**

**{**

**for(int** j**=**0**;** j**<**n**;** j**++)**

**{**

cout**<<" "<<**matris**[**i**][**j**];**

**}**

cout**<<**endl**;**

**}**

**}**

**}**

**double** RHS**[**m**+**1**][**1**];** *//sağ taraf sabitlerini tutan tek boyutlu matris*

RHS**[**0**][**0**]=**0**;**

cout**<<**endl**;**

**for(int** i**=**1**;** i**<**m**+**1**;** i**++)**

**{**

cout**<<**i**<<". kisit icin sag taraf sabitini giriniz:";**

cin**>>**RHS**[**i**][**0**];**

**}**

**char** sign**[**m**];** *//Kısıtların işaretlerini tutar*

**for(int** i**=**0**;** i**<**m**;** i**++)**

**{**

cout**<<**endl**<<**i**+**1**<<". kisitin isaretini giriniz ( < ise k, > ise b, = ise e):";**

cin**>>**sign**[**i**];**

**}**

**char** isrt**[**3**]={**'e'**,**'k'**,**'b'**};**

**for(int** i**=**1**;**i**<=**m**;**i**++)**

**{**

**if(**RHS**[**i**][**0**]<**0**){**RHS**[**i**][**0**]=(-**1**)\***RHS**[**i**][**0**];**

**if(**sign**[**i**-**1**]==**isrt**[**1**]){** sign**[**i**-**1**]=**isrt**[**2**];}**

**else** **if** **(**sign**[**i**-**1**]==**isrt**[**2**]){** sign**[**i**-**1**]=**isrt**[**1**];}**

**for(int** j**=**0**;**j**<**n**;**j**++){**matris**[**i**][**j**]=(-**1**)\***matris**[**i**][**j**];}}**

**}**

**int** sk**=**0**;** *//aylak (slack) değişken sayısı*

**for(int** i**=**0**;**i**<**m**;**i**++){if(**sign**[**i**]==**isrt**[**1**]){**sk**=**sk**+**1**;** **}}**

cout**<<"\nslack degisken sayisi: "<<**sk**<<**endl**;** *// sk: Kısıtlardaki < sayısı kadar aylak değişken eklenir. sk= aylak değişken sayısı*

**int** ak**=**0**;** *//artık değişken sayısı*

**for(int** i**=**0**;**i**<**m**;**i**++){if(**sign**[**i**]==**isrt**[**2**]){**ak**=**ak**+**1**;** **}}**

cout**<<"artik degisken sayisi: "<<**ak**<<**endl**;** *// sk: Kısıtlardaki > sayısı kadar artık değişken eklenir. ak= artık değişken sayısı*

**int** yk**=**0**;** *//yapay değişken sayısı*

**for(int** i**=**0**;**i**<**m**;**i**++){if(**sign**[**i**]==**isrt**[**0**]** **||** sign**[**i**]==**isrt**[**2**]){**yk**=**yk**+**1**;** **}}**

cout**<<"yapay degisken sayisi: "<<**yk**<<**endl**;** *// yk: Kısıtlardaki = ve > sayısı kadar yapay değişken eklenir. yk= yapay değişken sayısı*

**double** z**[**1**][**n**+**sk**+**ak**+**yk**];** *// z satırı*

**double** brm**[**m**][**sk**];**

**double** BFS0**[**m**][**n**+**sk**+**ak**+**yk**];**

**double** BFS1**[**m**+**1**][**n**+**sk**+**ak**+**yk**];**

**double** BFS**[**m**+**1**][**n**+**sk**+**ak**+**yk**+**1**];** *//basic feasible solution*

**double** M**=**100**;**

*//BFS0*

**for(int** i**=**0**;** i**<**m**;** i**++)**

**{**

**for(int** j**=**0**;** j**<**n**;** j**++)**

**{**

BFS0**[**i**][**j**]=**matris**[**i**][**j**];**

**}**

**}**

**for(int** i**=**0**;** i**<**m**;** i**++)**

**{**

**for(int** j**=**n**;** j**<**n**+**sk**+**ak**+**yk**;** j**++)**

**{**

BFS0**[**i**][**j**]=**0**;**

**}**

**}**

**int** q**=**0**;**

**for(int** i**=**0**;** i**<**m**;** i**++)**

**{**

**if(**sign**[**i**]==**isrt**[**0**])** **{**BFS0**[**i**][**n**+**i**+**q**]=**1**;}** *// =*

**else** **if(**sign**[**i**]==**isrt**[**1**]){** BFS0**[**i**][**n**+**i**+**q**]=**1**;** **}** *// <*

**else** **if(**sign**[**i**]==**isrt**[**2**])** **{**BFS0**[**i**][**n**+**i**+**q**]=-**1**;** BFS0**[**i**][**n**+**i**+**q**+**1**]=**1**;** q**=**q**+**1**;}** *// >*

**}**

*//z satırı*

**for(int** i**=**0**;** i**<**n**;** i**++)** z**[**0**][**i**]=(-**1**)\***c**[**0**][**i**];**

**for(int** i**=**n**;** i**<**n**+**ak**+**sk**+**yk**;** i**++)** z**[**0**][**i**]=**0**;**

q**=**0**;**

**if(!**strcmp**(**tip**,** **"min"))**

**{**

**for(int** i**=**0**;** i**<**m**;** i**++)**

**{**

**if(**sign**[**i**]==**isrt**[**0**])** z**[**0**][**n**+**i**+**q**]=-**M**;**

**else** **if(**sign**[**i**]==**isrt**[**2**]** **)** **{**z**[**0**][**n**+**i**+**1**+**q**]=-**M**;** q**=**q**+**1**;}**

**}**

**}**

q**=**0**;**

**if(!**strcmp**(**tip**,** **"max"))**

**{**

**for(int** i**=**0**;** i**<**m**;** i**++)**

**{**

**if(**sign**[**i**]==**isrt**[**0**])** z**[**0**][**n**+**i**+**q**]=**M**;**

**else** **if(**sign**[**i**]==**isrt**[**2**]** **)** **{**z**[**0**][**n**+**i**+**1**+**q**]=**M**;** q**=**q**+**1**;}**

**}**

**}**

*//BFS1*

**for(int** i**=**0**;** i**<**n**+**sk**+**ak**+**yk**;** i**++)** BFS1**[**0**][**i**]=**z**[**0**][**i**];**

**for(int** i**=**1**;** i**<**m**+**1**;** i**++)**

**{**

**for(int** j**=**0**;** j**<**n**+**sk**+**ak**+**yk**;** j**++)**

**{**

BFS1**[**i**][**j**]=**BFS0**[**i**-**1**][**j**];**

**}**

**}**

*//BFS*

**for(int** i**=**0**;** i**<**m**+**1**;** i**++)**

**{**

**for(int** j**=**0**;** j**<**n**+**sk**+**ak**+**yk**+**1**;** j**++)**

**{**

**if(**j**==**n**+**sk**+**ak**+**yk**)** BFS**[**i**][**j**]=**RHS**[**i**][**0**];**

**else** BFS**[**i**][**j**]=**BFS1**[**i**][**j**];**

**}**

**}**

**int** sat**[**yk**],** idx**[**yk**];** *// yapay değişkenin sütun numarasını tutar*

**int** k**=**0**;**

**for(int** i**=**0**;** i**<**n**+**sk**+**ak**+**yk**;** i**++)**

**{**

**if(**z**[**0**][**i**]==**M **||** z**[**0**][**i**]==-**M**)**

**{**

idx**[**k**]=**i**;**

**for(int** j**=**1**;** j**<**m**+**1**;** j**++)**

**{**

**if(**BFS**[**j**][**i**]==**1**){**sat**[**k**]=**j**;** k**=**k**+**1**;}**

**}**

**}**

**}**

**for(int** i**=**0**;** i**<**yk**;** i**++)** cout**<<**endl**<<" idx["<<**i**+**1**<<"]: "<<**idx**[**i**]+**1**<<**endl**;**

**for(int** i**=**0**;** i**<**yk**;** i**++)**

**{**

**for(int** j**=**0**;** j**<**n**+**sk**+**ak**+**yk**+**1**;** j**++)**

**{**

**if(!**strcmp**(**tip**,** **"min"))** BFS**[**0**][**j**]=**M**\***BFS**[**sat**[**i**]][**j**]+**BFS**[**0**][**j**];**

**else** **if(!**strcmp**(**tip**,** **"max"))**

**{**BFS**[**0**][**j**]=-**M**\***BFS**[**sat**[**i**]][**j**]+**BFS**[**0**][**j**]; }**

**}**

**}**

cout**<<"\n BFS: \n";**

**for(int** i**=**0**;** i**<**m**+**1**;** i**++)**

**{**

**for(int** j**=**0**;** j**<**n**+**sk**+**ak**+**yk**+**1**;** j**++)**

**{**

cout**<<" "<<**BFS**[**i**][**j**]<<" ";**

**}**

cout**<<**endl**;**

**}**

**bool** optimal**=false;**

*//optimallik şartı*

**if(!**strcmp**(**tip**,** **"max"))**

**{**

**int** k**=**0**;**

**for(int** i**=**0**;** i**<**n**+**sk**+**ak**+**yk**;** i**++)**

**{**

**if(**BFS**[**0**][**i**]<=**0**){**k**=**k**+**1**;}**

**}**

**if(**k**==**0**)**

**{**

cout**<<"\nmax problemi icin optimallik sarti saglanmistir.";** optimal**=true;**

cout**<<"\nX degerleri: ";**

**for(int** i**=**0**;** i**<**n**;** i**++){**cout**<<**X**[**i**][**0**]<<" ";}**

cout**<<"\nz: "<<**BFS**[**0**][**n**+**sk**+**ak**+**yk**]<<**endl**;**

**}**

**}**

**else** **if(!**strcmp**(**tip**,** **"min"))**

**{**

**int** k**=**0**;**

**for(int** i**=**0**;** i**<**n**+**sk**+**ak**+**yk**;** i**++)**

**{**

**if(**BFS**[**0**][**i**]>=**0**){**k**=**k**+**1**;}**

**}**

**if(**k**==**0**)**

**{**

cout**<<"\nmin problemi icin optimallik sarti saglanmistir.\n";** optimal**=true;**

cout**<<"\nX degerleri: ";**

**for(int** i**=**0**;** i**<**n**;** i**++){**cout**<<**X**[**i**][**0**]<<" ";}**

cout**<<"\nz: "<<**BFS**[**0**][**n**+**sk**+**ak**+**yk**]<<**endl**;**

**}**

**}**

*//Simplex iterasyon başlangıcı*

**int** iterasyon**=**0**;**

**double** optz**;** *//z optimal değeri*

**int** x**=-**1**;** *// anahtar sütun numarası*

y**=-**1**;** *//anahtar satır numarası*

**if(**optimal**==false)**

**{**

**do**

**{**

iterasyon**=**iterasyon**+**1**;**

cout**<<"\niterasyon: "<<**iterasyon**<<**endl**;**

*//Anahtar satır ve sütun*

**double** pcol**[**m**+**1**][**1**];**

**double** prow**[**1**][**n**+**sk**+**ak**+**yk**+**1**];**

**if(!**strcmp**(**tip**,** **"max"))**

**{**

**double** xx**=**0**;**

x**=-**1**;**

**for(int** i**=**0**;** i**<**n**+**sk**+**ak**+**yk**;** i**++)**

**{**

**if(**BFS**[**0**][**i**]<**0**)**

**{**

**if(**abs**(**BFS**[**0**][**i**])>**xx**)**

**{**

xx**=**abs**(**BFS**[**0**][**i**]);**

x**=**i**;** cout**<<"x= "<<**x**;**

**}**

**}**

**}**

**for(int** i**=**0**;** i**<**yk**;** i**++)**

**{**

**if(**x**==-**1 **||** x**==**idx**[**i**])** **{** x**=-**2**;}**

**}**

**if(**x**==-**2**){**cout**<<"\n\*\*\*Mumkun olmayan cozum!\*\*\*\n";** **break;}**

cout**<<"\nanahtar sutun: "<<**x**+**1**<<". sutun";**

**for(int** i**=**0**;** i**<**m**+**1**;** i**++)**

**{**

pcol**[**i**][**0**]=** BFS**[**i**][**x**];**

cout**<<"\n"<<**pcol**[**i**][**0**];**

**}**

**double** yy**=**99999**;**

y**=-**1**;**

**for(int** i**=**1**;** i**<**m**+**1**;** i**++)**

**{**

**if(**pcol**[**i**][**0**]>**0**)**

**{**

**if(**abs**(**BFS**[**i**][**n**+**sk**+**yk**+**ak**]/**pcol**[**i**][**0**])<=**yy**)**

**{**

yy**=**abs**(**BFS**[**i**][**n**+**sk**+**yk**+**ak**]/**pcol**[**i**][**0**]);**

y**=**i**;**

**}**

**}**

**}**

**if(**y**==-**1**){**cout**<<"\n\n\*\*\*Sinirsiz cozum!\*\*\*";** **break;}**

cout**<<"\n\nanahtar satir: "<<**y**+**1**<<". satir";**

**for(int** i**=**0**;** i**<**n**+**sk**+**ak**+**yk**+**1**;** i**++)**

**{**prow**[**0**][**i**]=** BFS**[**y**][**i**];}**

cout**<<"\nanahtar satir: ";**

**for(int** i**=**0**;** i**<**n**+**sk**+**ak**+**yk**+**1**;** i**++)**

**{**

cout**<<" "<<**prow**[**0**][**i**];**

**}**

cout**<<"\n\nSecilen hucre: "<<**BFS**[**y**][**x**]<<**endl**;**

**}**

**else** **if(!**strcmp**(**tip**,** **"min"))**

**{**

**double** xx**=**0**;**

x**=-**1**;**

**for(int** i**=**0**;** i**<**n**+**sk**+**ak**+**yk**;** i**++)**

**{**

**if(**BFS**[**0**][**i**]>**0**)**

**{**

**if(**BFS**[**0**][**i**]** **>**xx**){**xx**=**BFS**[**0**][**i**];** x**=**i**;}**

**}**

**}**

**for(int** i**=**0**;** i**<**yk**;** i**++)**

**{**

**if(**x**==-**1 **||** x**==**idx**[**i**])** **{** x**=-**2**;}**

**}**

**if(**x**==-**2**){**cout**<<"\n\*\*\*Mumkun olmayan cozum!\*\*\*\n";** **break;}**

**else** **{**cout**<<"\nanahtar sutun: "<<**x**+**1**<<". sutun";** **}**

**for(int** i**=**0**;** i**<**m**+**1**;** i**++)**

**{**

pcol**[**i**][**0**]=** BFS**[**i**][**x**];**

cout**<<"\n"<<**pcol**[**i**][**0**];**

**}**

**double** yy**=**99999**;**

y**=-**1**;**

**for(int** i**=**1**;** i**<**m**+**1**;** i**++)**

**{**

**if(**pcol**[**i**][**0**]>**0**)**

**{**

**if(**abs**(**BFS**[**i**][**n**+**sk**+**ak**+**yk**]/**pcol**[**i**][**0**])<=**yy**)**

**{**

yy**=**abs**(**BFS**[**i**][**n**+**sk**+**ak**+**yk**]/**pcol**[**i**][**0**]);**

y**=**i**;**

**}**

**}**

**}**

**if(**y**==-**1**){**cout**<<"\n\n\*\*\*Sinirsiz cozum!\*\*\*";** **break;}**

cout**<<"\n\nanahtar satir: "<<**y**+**1**<<". satir";**

**for(int** i**=**0**;** i**<**n**+**sk**+**ak**+**yk**+**1**;** i**++)**

**{**prow**[**0**][**i**]=** BFS**[**y**][**i**];}**

cout**<<"\nanahtar satir: ";**

**for(int** i**=**0**;** i**<**n**+**sk**+**ak**+**yk**+**1**;** i**++)**

**{**

cout**<<" "<<**prow**[**0**][**i**];**

**}**

cout**<<"\n\nSecilen hucre: "<<**BFS**[**y**][**x**]<<**endl**;**

**}**

*//Satır işlemleri*

**double** cell**=**BFS**[**y**][**x**];**

**double** k**[**m**+**1**][**1**];** *//satır katsayılarını tutan vektör*

**for(int** i**=**0**;** i**<**m**+**1**;** i**++)**

**{**

k**[**i**][**0**]=-**pcol**[**i**][**0**]/**cell**;**

**}**

*// katsayıları tüm matris ile çarp*

**for(int** i**=**0**;** i**<**m**+**1**;** i**++)**

**{**

**for(int** j**=**0**;** j**<**n**+**sk**+**ak**+**yk**+**1**;** j**++)**

**{**

BFS**[**i**][**j**]=**prow**[**0**][**j**]\***k**[**i**][**0**]+**BFS**[**i**][**j**];**

**}**

**}**

**for(int** i**=**0**;** i**<**n**+**sk**+**ak**+**yk**+**1**;** i**++)**

**{**

BFS**[**y**][**i**]=**prow**[**0**][**i**]/**cell**;**

**}**

cout**<<"\nmatris \n";**

**for(int** i**=**0**;** i**<**m**+**1**;** i**++)**

**{**

**for(int** j**=**0**;** j**<**n**+**sk**+**ak**+**yk**+**1**;** j**++)**

**{**

cout**<<" "<<**BFS**[**i**][**j**]<<" ";**

**}**

cout**<<**endl**;**

**}**

*//optimallik şartı*

**if(!**strcmp**(**tip**,** **"max"))**

**{**

**int** k**=**0**;**

**for(int** i**=**0**;** i**<**n**+**sk**+**ak**+**yk**;** i**++)**

**{**

**if(**BFS**[**0**][**i**]<**0**){**k**=**k**+**1**;}**

**}**

**if(**k**==**0**){** optimal**=true;}**

**}**

**else** **if(!**strcmp**(**tip**,** **"min"))**

**{**

**int** k**=**0**;**

**for(int** i**=**0**;** i**<**n**+**sk**+**ak**+**yk**;** i**++)**

**{**

**if(**BFS**[**0**][**i**]>**0**){**k**=**k**+**1**;}**

**}**

**if(**k**==**0**)** **{** optimal**=true;** **}**

**}**

**for(int** i**=**0**,**q**=**0**;** i**<**n**+**sk**+**ak**+**yk**;** i**++)**

**{**

**if(**BFS**[**0**][**i**]==**0**)** q**=**q**+**1**;**

**if(**q**>**m**){**cout**<<**endl**<<"\n\*\*\*Alternatif cozum var!\*\*\*\*\n";}**

**}**

**double** x**[**n**];** *//gerekiyor*

**if(**optimal**==true)**

**{**

**for(int** i**=**0**;** i**<**n**;** i**++)**

**{**

**int** bir**=**0**;**

**int** sfr**=**0**;**

**int** g**=**0**;** *//satır indisini tutar*

**for(int** j**=**0**;** j**<**m**+**1**;** j**++)**

**{**

**if(**BFS**[**j**][**i**]==**1**)** **{**bir**=**bir**+**1**;** g**=**j**;}**

**if(**abs**(**BFS**[**j**][**i**])<**0.0000001**){** sfr**=**sfr**+**1**;}**

**}**

**if(**bir**==**1 **&&** sfr**==**m**)** x**[**i**]=**BFS**[**g**][**n**+**sk**+**ak**+**yk**];**

**else** x**[**i**]=**0**;**

**}**

optz**=**BFS**[**0**][**n**+**sk**+**ak**+**yk**];**

**double** fonk**=**0**;**

**for(int** i**=**0**;** i**<**N**;** i**++)**

**{**

fonk**=** fonk **+** C**[**0**][**i**]\***x**[**i**];**

**}**

**if(**round**(**fonk**)!=**round**(**optz**))** **{** cout**<<"fonk= "<<**fonk**<<"optz= "<<**optz**<<"\n\*\*\*Mumkun olmayan cozum!\*\*\*\n ";** optimal**=false;** **break;** **}**

**for(int** i**=**0**,**q**=**0**;** i**<**N**;** i**++)**

**{**

**if(**ozx**[**i**]==**'k'**)** **{** X**[**i**][**0**]=(-**1**)\***x**[**i**+**q**];}**

**if(**ozx**[**i**]==**'s'**)** **{** X**[**i**][**0**]=**x**[**i**+**q**]-**x**[**i**+**q**+**1**];**q**=**q**+**1**;}**

**if(**ozx**[**i**]==**'b'**)** **{** X**[**i**][**0**]=**x**[**i**+**q**];** **}**

**}**

cout**<<"\n\n\n Optimal Cozum Bulunmustur: ";**

cout**<<"\n\nX degerleri:\n\n";**

**for(int** i**=**0**;** i**<**N**;** i**++){**cout**<<"X["<<**i**+**1**<<"]= "<<**X**[**i**][**0**]<<**endl**;}**

cout**<<"\nz: "<<**optz**<<**endl**;**

**}**

**}while(**optimal**==false);**

**}**

**return** 0**;**

**}**