

İçindekiler

Analiz	2
Tasarım	3
Matrisin Alınması	3
Matrisin Aranması	3
Puanın Tablolanması	3
Verilerin Yazdırılması	3
Programcı Kataloğu	4
Fonksiyonlar ve Prototipleri	4
int main(void) fonksiyonu	4
int yeniVektor(void) fonksiyonu	6
void matrisYazdir(int matris[BOYUT][BOYUT]) fonksiyonu	6
int puanYerlestir(int[][BOYUT],int,int,int,int) fonksiyonu	7
void yonYazdir(int yon) fonksiyonu	7
Kullanılan Veri Yapıları	8
int matris[BOYUT][BOYUT]	8
int matrisPuan[BOYUT][BOYUT]	8
int vektorEleman[BOYUT]	8
Kaynak Kodu	9
Kullanıcı Kataloğu	20
Programın Kullanım Kılavuzu	20
Programdaki Kısıtlamalar	20

Analiz

Bizden 20x20 büyüklüğünde bir matris içerisinde; uzunluğunu kullanıcının belirlediği, değerleri ise yine kullanıcının girdiği ve bir puan değeri olan vektörü aramamız isteniyor.

Matris üzerinde bulunan vektör olduğunda, bu vektörün puanına göre kullanıcının aldığı puanın hesaplanması gerekiyor. Eğer bulunan iki vektörün kesiştiği bir veya daha fazla nokta varsa, puanlama sırasında puanı büyük olan vektörün puanı toplam puana katılıyor. Aranacak vektör matris içinde 8 farklı şekilde bulunabiliyor. Yatay(doğu,batı), dikey(kuzey,güney), diagonal(4 ara yön).

Her vektör aramadan sonra kullanıcıya yeni vektör aramak isteyip istemediği sorulması gerekiyor.

Her veri alma sonrasında yazdırılacaklar;

- Vektörün matris içindeki başlangıç konumu
- Vektörün yönü
- Vektörün uzunluğu

Tüm işlem tamamlandıktan sonra ise ekrana kullanıcının puanının yazdırılması gerekiyor.

Tasarım

Matrisin Alınması

20x20 büyüklüğündeki matris, programla aynı klasör içindeki “matris.txt” dosyasından alınmalıdır. Dosya içindeki sayılar tamsayı olarak saklanacağından ve her değer arasında boşluk bulunacağından, integer tipinde tanımlanmış 2boyutlu bir dizi olan matrisimizin içine iki tane 20lik for döngüsü içindeki fscanf ile değerlerimizi atayabiliriz.

Matrisin Aranması

İlk olarak kullanıcıdan vektörün uzunluğu, vektörün elemanlarının değerleri ve vektörün puanı kullanıcıdan alınmalıdır. Alınan vektörün ilk değerine matris içerisinde ulaşılan kadar tarama yapılmalıdır. Eğer vektörün ilk değeri matris içerisindeki bir değere eşitse, bu değer etrafındaki değer vektörün 2.değeri olup olmadığına bakılmalıdır. Eğer 2.değere bakılırken bir eşitliğe rastlanmışsa, rastlanan bu yönde vektörün tamamını bulana kadar bakılmaya devam edilmelidir.

Eğer tüm işlemler sonucunda bir vektör bulunamamışsa ekrana aranan vektörün matris içinde bulunmadığı yazılmalıdır. Eğer bulunmuşsa, vektörün puanı tabloya konmalıdır.

Puanın Tablolanması

Bu tablolaştırma için fonksiyon kullanmak mantıklı olacaktır. Tablolaştırma için kullandığımız fonksiyon, 20x20'lik bir matris içine daha önceden bulduğumuz vektörün bulunduğu konuma gelecek şekilde her hücreye vektörün puanını doldurmalıdır. Bunun için vektörün yönü, vektörün uzunluğu, vektörün başlangıç konumu fonksiyon tarafından alınmalıdır.

Verilerin Yazdırılması

C dilinde dizi kullanırken, kullandığımız bu dizi [0][0] ilk elemanı ile başladığı için 0'dan başlayan iki for döngüsü için yazdırılacak konum (i,j) yerine (i+1,j+1) olmalıdır.

Vektörün yönü 1-8 arası sayısal değerlerle tutulup, yazdırma kısmında switch-case kullanılarak metin haline dönüştürülebilir. Vektörün boyu ise kullanıcının girdiği değer olacaktır.

Programcı Kataloğu

Fonksiyonlar ve Prototipleri

int main(void) fonksiyonu

Programın büyük kısmının gerçekleştiği bölüm main() fonksiyonu. Değişkenlerin tanımlandığı kısımda *matrisPtr adresi FILE olarak atandı.

```
if( (matrisPtr=fopen("matris.txt","r"))==NULL)
    printf("Dosya acilamadi");/
else
{
```

Yukarıdaki kodla dosyanın düzgün açılması sağlandı. Eğer matris.txt dosyası bulunamamışsa, program kullanıcıya “Dosya açilamadi” uyarısı gönderiyor. Eğer dosya düzgün açılmışsa ilk olarak aşağıdaki for döngüsü çalışıyor.

```
for(i=0;i<BOYUT;i++)
{
    //matrise değer alırken içinde bulunduğumuz satır
    for(j=0;j<BOYUT;j++)
    {
        //matrise değer alırken içinde bulunduğumuz sütun
        fscanf(matrisPtr,"%d",&matris[i][j]);
    }
}
```

Bu for döngüsü matris.txt dosyası içindeki tamsayı değerlerini teker teker alarak sırayla matrisin içine atıyor. Matris içindeki “i” değeri içinde bulunduğu satırı, “j” değeri içinde bulunduğu sütunu gösteriyor. Sonrasında fonksiyon do-while döngüsü içine giriyor. Bu döngüden yeni arama yapılmak istenmediği takdirde çıkılabiliniyor.

```
do
{
    //Kullanıcı arama yapmak istemeyene kadar dönen döngü
    //İşlemler bu aralıkta yapılıyor.
}while(dongudenCik==1);
```

İşlemlerin yapıldığı aralığa bakacak olursak, ilk kısımda kullanıcıdan arayacağı vektör alınıyor.

```
matrisYazdir(matris); //matris.txt ekrana yazdırılıyor
printf("Vektördeki eleman sayisini girin:");
scanf("%d",&vektorElemanSayisi);
printf("Vektördeki elemanlar sırayla istenecektir\n");
for(i=0;i<vektorElemanSayisi;i++)
{
    printf("Vektörün %d.elemanini girin:",i+1);
    scanf("%d",&vektorEleman[i]);
}
printf("Vektörün puanini girin:");
scanf("%d",&vektorPuani);
```

Daha sonra bu vektöre göre for döngüsüne giriliyor.

```
for(i=0;i<BOYUT;i++)//Yatayda ve dikeyde vektör araması
{
    //Bu döngüde satırdayım
    1 for(j=0;j<=BOYUT-vektorElemanSayisi;j++)
    {
        //Bu döngüde sütundayım
        for(k=0;k<vektorElemanSayisi;k++){
            .. }
            ..}
    2 for(j=BOYUT-1;j>=vektorElemanSayisi-1;j--){
        //Bu döngüde sütundayım
        for(k=0;k<vektorElemanSayisi;k++){
            .. }
            ..}
    }
```

Yukarıdaki döngülerde, 1 numaralı kısımda kuzeydoğu, doğu, güneydoğu ve güney yönlerinde arama yapılırken, 2 numaralı kısımda tam tersi yönlerle bakılmaktadır.

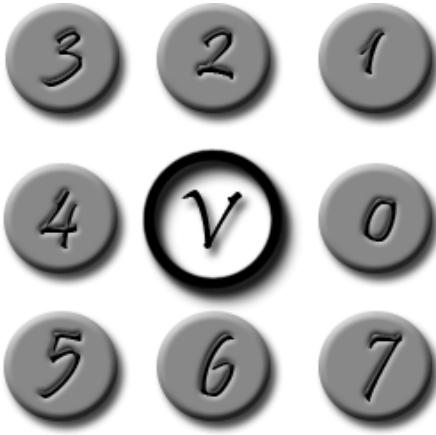
1 fonksiyonu içinde yapılanlara bakacak olursak,

```
if(vektorEleman[k]!=matris[i][j+k]) yatay_bulunamadi=1;//Doğuya kontrol
if(vektorEleman[k]!=matris[j+k][i]) dusey_bulunamadi=1;//Güneye kontrol
if((i+k)<BOYUT)
{
    //Bu ifle matris değerleri dışında arama yapılmasını engelliyorum
    if(vektorEleman[k]!=matris[i+k][j+k]) diagonal_bulunamadi=1;//Güneydoğuya kontrol
}
else diagonal_bulunamadi=1;//eğer vektör matris dışına çıkmışsa vektör bulunamadı
if((i-k)>=0)
{
    //Bu ifle matris değerleri dışında arama yapılmasını engelliyorum
    if(vektorEleman[k]!=matris[i-k][j+k]) ters_diagonal_bulunamadi=1;//Kuzeydoğuya kontrol
}
else ters_diagonal_bulunamadi=1;//eğer vektör matris dışına çıkmışsa vektör bulunamadı

/*Hiçbir eşitlik sağlanamamışsa aşağıdaki if ile döngüden çıkıyorum*/
if
(yatay_bulunamadi==1&&dusey_bulunamadi==1&&diagonal_bulunamadi==1&&ters_diagonal_bulunamadi==1) break;
```

İçinde bulunulduğu i.satır ve j.sütuna göre tek tek arama yapılmaktadır. Eğer ilk elemanda doğruluk sağlanmışsa, diğer elemanların eşitliğini de kontrol etmek için aranan diğer vektörün elemanı ilgili yönü bulduracak şekilde matris içinde bakılan değeri gösterecek şekilde i veya j'nin yanına eklenir.

```
if(yatay_bulunamadi==0){
    vektorunKonumu=i*BOYUT+j;
    vektorunYonu=0;
    vektorBulundu=1;
}
if(dusey_bulunamadi==0){
    vektorunKonumu=i*BOYUT+j;
    vektorunYonu=6;
    vektorBulundu=1;
}
.....
```



Konum;

$i*BOYUT+j$ olarak belirlenmiştir. Bu, matris içindeki (0,0) noktasından ne kadar ilerlendiğini göstermektedir.

İşlemlerde kullanılan 8 yönün değeriye sol taraftaki şemada gösterilmiştir.

Eğer vektör matris içinde bulunamamışsa ekrana vektörün matris içinde bulunamadığı yazdırılır. Eğer bulunmuşsa, vektörün puanı `puanYerlestir()`

fonksiyonuyla puan tutan 20x20'lik matris içine yazdırılır. Yeni arama yapmayı sorgulamak için `yeniVektor()` fonksiyonu çağırılır. İyi bir programcı olmak istendiğinden, fonksiyonun sonunda `fclose(matrisPtr)` denerek dosya kapatılır.

int yeniVektor(void) fonksiyonu

```
//Kullanıcı E derse yeniden vektör soruyorum H derse programdan çıkıyorum
char evetHayir;
printf("Yeni bir vektör girmek ister misiniz?(E/H):");
fflush(stdin);
scanf("%c",&evetHayir);
if(evetHayir=='E' || evetHayir=='e') return 1;
else if(evetHayir=='H' || evetHayir=='h') return 0;
else return yeniVektor();
}
```

Fonksiyon kullanıcıya yeniden vektör çizmek isteyip istemediğini sorar. Eğer kullanıcı E veya e girerse 1 değeri döner ve ana döngüdeki do-while koşulu sağlanmış olur. Kullanıcı H veya h değeri girdiğinde ise döngüden çıkılır.

Kullanıcının bu değerler dışında girdiği değerlerde ise döngü kendini çağırır. Bu sayede kullanıcı doğru değeri girene kadar beklenmiş olunur.

void matrisYazdir(int matris[BOYUT][BOYUT]) fonksiyonu

```
{
    int i,j;
    printf("matris.txt dosyasi icindeki %dx%d matrisimiz:\n",BOYUT,BOYUT);
    for(i=0;i<BOYUT;i++){
        for(j=0;j<BOYUT;j++){
            printf("%3d",matris[i][j]);
        }//BOYUT*BOYUT büyüklüğündeki matrisimi yazdırıyorum.
        printf("\n");
    }
}
```

Fonksiyon,20*20 büyüklüğündeki matrisi yazdırmaya yarar.

int puanYerlestir(int [][]BOYUT,int,int,int,int) fonksiyonu

matrisPuan[][][BOYUT], vektorPuanı, vektorunKonumu, vektorunYonu, vektorunElemanSayisi değişkenlerini ana fonksiyondan alarak çalışır. Eğer matris içinde bir vektör bulunmuşsa bu fonksiyon çalışır. Örneğin “kuzeybatı” yönü için puan yerleştirme koduna bakalım.

```
case 3:
    for(i=0;i<vektorunElemanSayisi;i++)
        {//Kuzeybatı yönünde vektörün uzunluğu kadar

    if(puan>matrisPuan[vektorunKonumu/BOYUT-i][vektorunKonumu%BOYUT-i]){
    matrisPuan[vektorunKonumu/BOYUT-i][vektorunKonumu%BOYUT-i]=puan;
        eklenen_puan+=puan;}
    }break;
```

vektorunKonumu/BOYUT=içinde bulunulan satır

vektorunKonumu%BOYUT=içinde bulunulan sütun

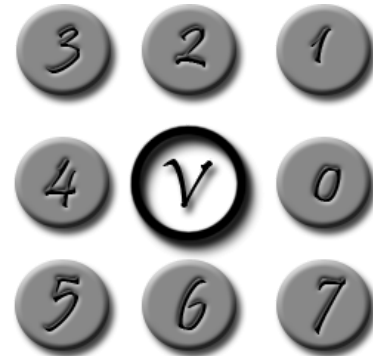
i=0 durumunda vektörün başlangıç konumundayız. Satır ve sütundan 1 çıkardığımızda, sol üst yani kuzeybatı yönünde hareket etmiş oluruz. Bu işlem vektörün uzunluğu kadar devam eder. Ve vektörün asıl matriste bulunduğu konumda matrisPuan üzerine vektörün puanı yazılır. İçerideki if ile de o an elde olan puan değerinin matrisPuan üzerinde bulunan değerden büyük olup olmadığı kontrol ediliyor.

[-i][-i]	[-i][s]	[-i][+i]
[s][-i]	V	[s][+i]
[+i][-i]	[+i][s]	[+i][+i]

Yandaki tabloda vektörün yönüne göre kullanılan +,- ler gösterilmiştir.

void yonYazdir(int yon) fonksiyonu

```
switch (yon)
{
    case 0: printf("Dogu");break;
    case 1:
    printf("Kuzeydogu");break;
    case 2: printf("Kuzey");break;
    case 3:
    printf("Kuzeybati");break;
    case 4: printf("Bati");break;
    case 5:
    printf("Guneybati");break;
    case 6: printf("Guney");break;
    case 7:
    printf("Guneydogu");break;
}
```



Sağdaki şemaya göre yön yazdıran fonksiyon.

Kullanılan Veri Yapıları

int matris[BOYUT][BOYUT]

matris.txt dosyasından aldığımız tamsayı değerlerini tutan iki boyutlu bir dizidir. Programın başında `#define BOYUT 20` denildiğinden, dizimizin toplam 400 tamsayı kapasitesi vardır.

int matrisPuan[BOYUT][BOYUT]

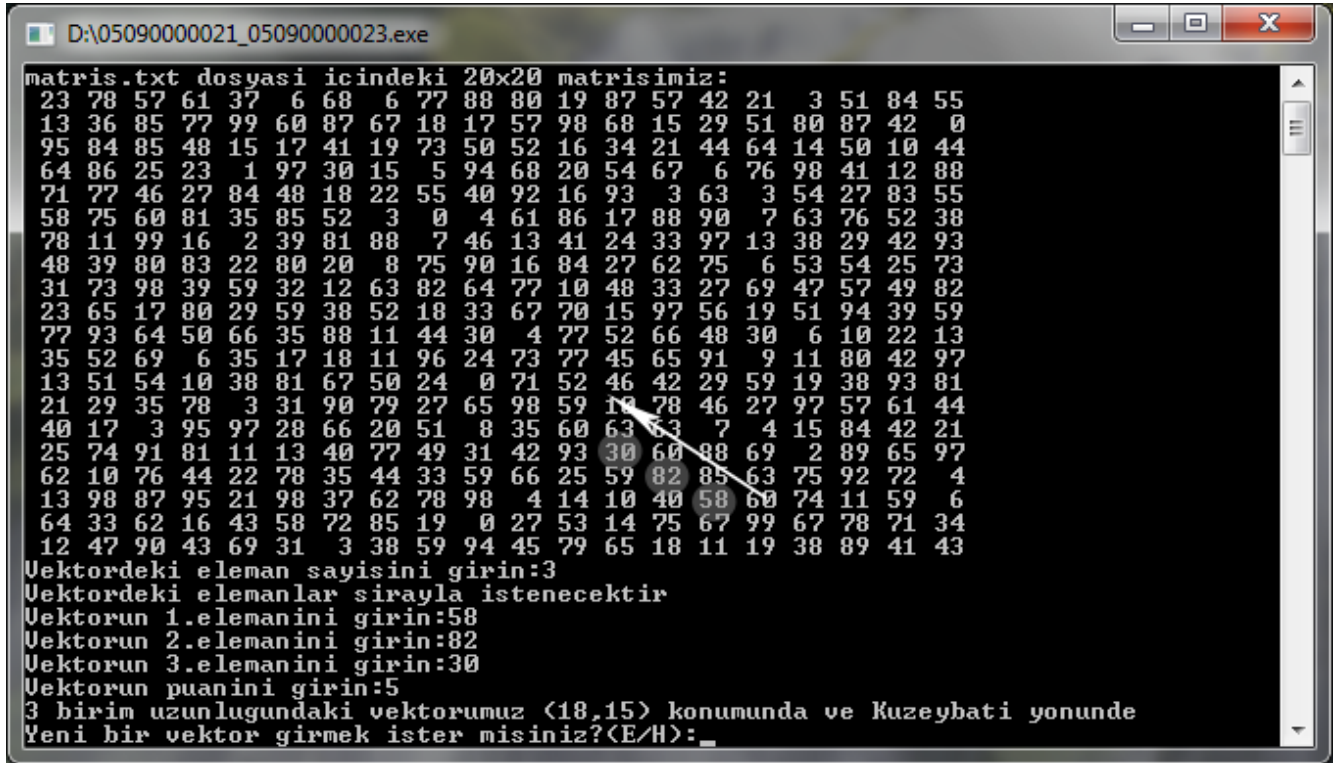
Puanlama işlemi yapıldıktan sonra puan tablosunu içinde tutan iki boyutlu dizidir. 400 tamsayı kapasitesi vardır.

int vektorEleman[BOYUT]

Her vektör aramanın başında vektörün uzunluğu kadarki kısmı değiştirilen dizidir. Vektörün elemanlarını içinde tutar. Maksimum matrisin bir kenarı kadar elemanı olabileceğinden `BOYUT=20` tamsayı değeri tutar.

Kullanıcı Kataloğu

Programın Kullanım Kılavuzu



```
D:\05090000021_05090000023.exe
matris.txt dosyasi icindeki 20x20 matrisimiz:
23 78 57 61 37 6 68 6 77 88 80 19 87 57 42 21 3 51 84 55
13 36 85 77 99 60 87 67 18 17 57 98 68 15 29 51 80 87 42 0
95 84 85 48 15 17 41 19 73 50 52 16 34 21 44 64 14 50 10 44
64 86 25 23 1 97 30 15 5 94 68 20 54 67 6 76 98 41 12 88
71 77 46 27 84 48 18 22 55 40 92 16 93 3 63 3 54 27 83 55
58 75 60 81 35 85 52 3 0 4 61 86 17 88 90 7 63 76 52 38
78 11 99 16 2 39 81 88 7 46 13 41 24 33 97 13 38 29 42 93
48 39 80 83 22 80 20 8 75 90 16 84 27 62 75 6 53 54 25 73
31 73 98 39 59 32 12 63 82 64 77 10 48 33 27 69 47 57 49 82
23 65 17 80 29 59 38 52 18 33 67 70 15 97 56 19 51 94 39 59
77 93 64 50 66 35 88 11 44 30 4 77 52 66 48 30 6 10 22 13
35 52 69 6 35 17 18 11 96 24 73 77 45 65 91 9 11 80 42 97
13 51 54 10 38 81 67 50 24 0 71 52 46 42 29 59 19 38 93 81
21 29 35 78 3 31 90 79 27 65 98 59 10 78 46 27 97 57 61 44
40 17 3 95 97 28 66 20 51 8 35 60 63 63 7 4 15 84 42 21
25 74 91 81 11 13 40 77 49 31 42 93 30 60 88 69 2 89 65 97
62 10 76 44 22 78 35 44 33 59 66 25 59 82 85 63 75 92 72 4
13 98 87 95 21 98 37 62 78 98 4 14 10 40 58 60 74 11 59 6
64 33 62 16 43 58 72 85 19 0 27 53 14 75 67 99 67 78 71 34
12 47 90 43 69 31 3 38 59 94 45 79 65 18 11 19 38 89 41 43
Vektördeki eleman sayisini girin:3
Vektördeki elemanlar sirayla istenecektir
Vektörün 1.elemanini girin:58
Vektörün 2.elemanini girin:82
Vektörün 3.elemanini girin:30
Vektörün puanini girin:5
3 birim uzunlugundaki vektörümüz <18,15> konumunda ve Kuzeybatı yonunde
Yeni bir vektör girmek ister misiniz?(E/H):_
```

1. 05090000021_05090000023.exe dosyasını çalıştırın.
2. Arayacağınız vektörün eleman sayısını girin.
3. Sırayla vektörünüzün elemanlarını girin.
4. Vektörün puanını girin.
5. Eğer yeniden arama yapmak istiyorsanız gelen soruya E veya e, istemiyorsanız H veya h cevabı girin.

Bunların dışında,

- Veri girdileri “evet-hayır” soruları dışında sayısal olmalıdır.
- Vektördeki eleman sayısı en fazla 20 olmalıdır

Programdaki Kısıtlamalar

- Aynı vektörden iki tane olmasına karşın bunu belirleyen bir işlem yapılmamıştır. Bulunan vektör, her zaman sol üste en yakın olmalıdır.
- Sayısal girilmesi gereken yerlerde başka değerler girerse program sonsuz döngüye girmekte