

Soru 1

Soru1.c Soru2.c Soru3.c Soru4.c Soru5.c Soru6.c

```
1  #include<stdio.h>
2  int main(){
3      int i , k , sayacesli = 0 , sayacsessiz=0;
4      char sesli[] = {'A','E','I','U','O','a','e','i','o','u'} , sessiz[] = {'B','C','D','F',
5      char dizi[100];
6      printf("Lutfen metin giriniz.\n");
7      gets(dizi);
8
9      for(i=0 ; dizi[i]!='\0';i++){
10
11          for(k=0 ; sesli[k]!='\0' ; k++){
12              if(dizi[i]==sesli[k])
13                  sayacesli++;
14          }
15      }
16      for(i=0 ; dizi[i]!='\0';i++){
17
18          for(k=0 ; sessiz[k]!='\0' ; k++){
19              if(dizi[i]==sessiz[k])
20                  sayacsessiz++;
21          }
22      }
23      printf("Girdiginiz karakter dizisinde %d adet sesli harf vardır.",sayacesli);
24      printf("\nGirdiginiz karakter dizisinde %d adet sessiz harf vardır.",sayacsessiz);
25  }
```

3 – 5 : Uygulamamızda sesli ve sessiz harflerin sayısını bulacağımız için öncelikle sesli ve sessiz harflerin tanımlı olduğu bir dizi ardından da kullanıcıdan alacağımız kelimelerin koyulacağı diziyi oluşturduk.

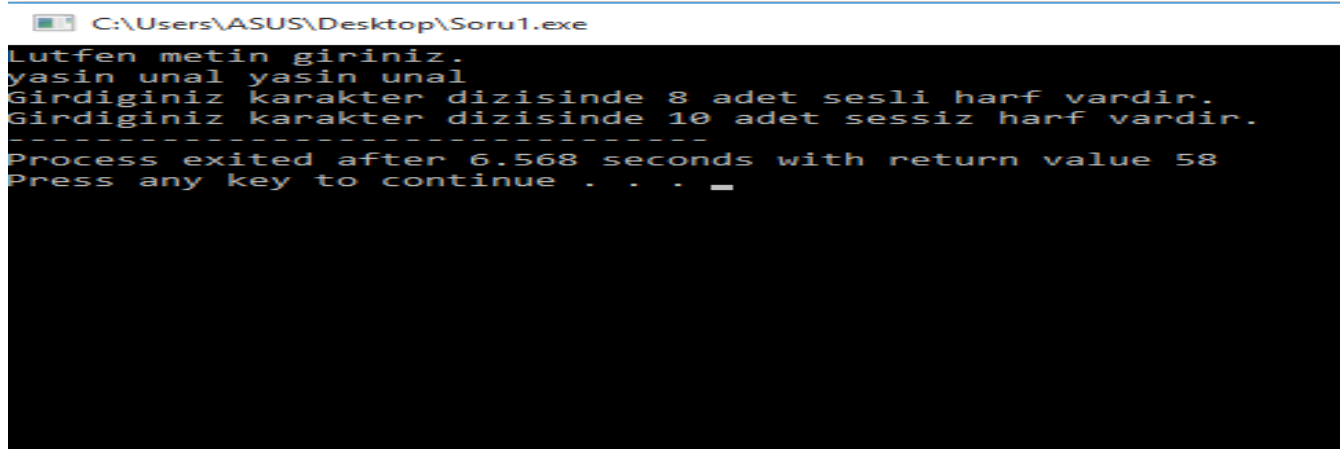
6 -7 : Kullanıcıdan kelimeleri aldık.

8-14 : Bir döngü oluşturarak kullanıcının girdiği tüm diziyi dolaştık. Program her dizi sonun \0 karakteri koyduğu için bu karaktere gelene kadar(yani dizi sonuna) diziyi dolaşacaktır.

Daha sonra döngü içine bir döngü daha koyarak önceden tanımladığımız sesli harfler dizisinde baştan sona dolaştım. Eğer girilen metnin ilk elemanı ile sesli dizisinin ilk elemanı eşitse sayacımız artacak. Değil ise sesli dizimizdeki tüm elemanlar tek tek denenecek(2.döngünün işi).Bulduğu eşleşmeleri “sayacesli” değişkenine atayacak. Sesli değişkeninin sonuna geldikten sonra girilen metnin ikinci elemanı içinde aynı işlemleri yapacak.

15-21 : Yine aynı şekilde sesli harfleri bulmak için yaptığımız işlemleri burada sessiz harfler için yapacağız. Bu sefer Sessiz harfler dizisinde dolaşacak ve eşleşenleri “sayacsessiz” isimli değişkende toplayacağız.

22-23 : Burada ise bulduğumuz değerleri ekrana yazdıracağız.



```
C:\Users\ASUS\Desktop\Soru1.exe
Lutfen metin giriniz.
yasin unal yasin unal
Girdiginiz karakter dizisinde 8 adet sesli harf vardır.
Girdiginiz karakter dizisinde 10 adet sessiz harf vardır.
-----
Process exited after 6.568 seconds with return value 58
Press any key to continue . . . _
```

Soru 2

```
Soru1.c Soru2.c Soru3.c Soru4.c Soru5.c Soru6.c
1  #include<stdio.h>
2  #include<math.h>
3  double doubleArray[] = {1.1,2.2,3.3,4.4,5.5,6.6,7.7,8.8,9.9,11.1,22.2,33.3,44.4,55.5,66.6,77.7,88.8,99.9 };
4  double calcAverage() , calcMedian() , calcStdDeviation() , calcVariance();
5  double diziboyutu = 18 , medyan , standartsapma , varyans;
6  int i =0 ;
7
```

1-2 : Uygulamamızda standart sapma hesaplayacağımız için kök almamız gerekecek. Bu yüzden math.h kütüphanesini ekliyoruz.

3-6 : Bize verilen diziyi ve hesaplayacağımız işlemleri tanımlıyoruz. Fonksiyon olanları parantez () ile gösteriyoruz. Noktalı sayılar olduğu için hepsini double olarak tanımladık. Dizimizin boyutu 18 olduğu için “diziboyutu” değişkenine 18 değerini atadık.

```
8  int main(){
9      printf("Dizinin aritmetik ortalamasi = %f\n",calcAverage());
10     printf("Dizinin medyani = %.2f\n",calcMedian());
11     printf("Dizinin varyansi = %f\n",calcVariance());
12     printf("Dizinin standart sapmasi = %f\n",calcStdDeviation());
13 }
```

8-13 : Bizden istenen işlemleri ekrana main fonksiyonu içinde ekrana basıyoruz. Printf komutu içine fonksiyonları değişken olarak atıyoruz.

```
14  double calcAverage(){
15      int i = 0;
16      double toplam = 0.0;
17      double ortalama = 0.0;
18
19      while(i<18){
23          ortalama = toplam / diziboyutu;
24          return ortalama ;
25      }
```

14 – 25 : calcAverage isimli fonksiyonumuz ile dizinin ortalamasını hesaplayacağız. Fonksiyon içinde toplam ve ortalama değişkenleri oluşturuyoruz. Bunların tipi double olmalı. Ardından while döngüsü ile tüm dizinin elemanlarını toplayıp dizi boyutuna bölüyoruz. Böylece ortalama değerimiz ortaya çıkıyor ve ortalama değerimizi döndürüyoruz.

```

26 double calcVariance(){
27     double deg1 = 0 ;
28     for(i=0;i<18;i++){
29         deg1 = deg1 + (doubleArray[i] - calcAverage()) * (doubleArray[i] - calcAverage());
30     }
31     varyans = deg1/17;
32     return varyans ;
33 }

```

26 – 33 : calcVariance isimli fonksiyonumuz ile dizinin varyansını hesaplayacağız. Öncelikle deg1 isimli bir değişken oluşturuyoruz. Ardından varyans için aşağıdaki formülü uyguluyoruz.

$$s^2 = \frac{\sum (x_i - \bar{x})^2}{n - 1}$$

s^2 = variance
 x_i = term in data set
 \bar{x} = Sample mean
 \sum = Sum
 n = Sample size

Her sayının ortalamaya olan uzaklığının karesini alıp bu değerleri topluyoruz. Bu toplanan değerleri dizi boyutunun bir eksiği olan 17 değerine bölüyoruz . Böylece varyansı buluyoruz.

```
34 double calcStdDeviation(){  
35     varyans = calcVariance();  
36     standartsapma = sqrt(varyans);  
37     return standartsapma;  
38  
39 }
```

34-39 : calcStdDeviation isimli fonksiyonumuz ile dizinin standart sapmasını hesaplayacağız. Bunu yapmak için tanımladığımız kütüphanemizden yararlanabiliriz. Çünkü varyansın karekökü standart sapmayı verecektir. Bu yüzden sqrt fonksiyonuna hesapladığımız varyans değerini parametre olarak vererek standart sapmayı hesaplıyoruz.

```
40 double calcMedian(){  
41     medyan = (doubleArray[17/2]+doubleArray[(17/2)+1])/2;  
42     return medyan ;  
43 }
```

40-43 : calcMedian isimli fonksiyonumuz ile dizinin medyanını yani ortanca değerini hesaplayacağız. Dizinin ortanca değeri eğer dizi boyutu tekse tam ortadaki sayı , çiftse ortadaki sayı ile bir sonraki sayının ortalamasıdır. Dizimizin boyutu 18 olduğu için 9 ve 10. elemanları bulmalıyız. Fakat dizi indisi 0 dan başladığı için orta değeri 18/2 değil 17/2 = 8 (int olduğu için) diyerek bulabiliriz. Bir sonraki değer içinde (17/2)+1 ile buluruz. Bu iki değer ortalaması ise bize medyanyı verecektir.

```
C:\Users\ASUS\Desktop\Soru2.exe
Dizinin aritmatik ortalamasi = 30.500000
Dizinin medyani = 10.50
Dizinin varyansi = 1100.894118
Dizinin standart sapmasi = 33.179724

-----
Process exited after 0.007571 seconds with return value 37
Press any key to continue . . .
```

Soru 3

```
1  #include <stdio.h>
2  #include<conio.h>
3  #include<string.h>
4  char charArray [] = {"A computer is a machine that can be instructed to carry out sequences of arithmetic or logical operations automati
5  char sesli[10] = {'A','E','I','U','O','a','e','i','o','u'} , sessiz[38] = {'B','C','D','F','G','H','J','K','L','M','N','P','R','S','T','
6  char karakter , maxsesli , maxsessiz ;
7
8  int i , k , sayac=1 , max1=0 , max2=0 , kelimesayisi=1 , sayackarakter = 0 , hecesayisi = 0;
9  int main(){
10
```

1-8 : Bize verilen diziye ve kullanacağımız değişkenleri tanımlıyoruz. Ayrıca strlen (dizi boyutu veren fonksiyon) kullanabilmemiz için string.h kütüphanesini eklememiz gerekiyor.

```

10
11 //Kullanıcı tarafından girilen harfin kaç kez geçtiğini veren kodlar.
12 printf("Lutfen bir harf giriniz:\n");
13 scanf("%c",&karakter);
14 for(i = 0 ; charArray[i] != '\0' ; i++ ){
15     if(charArray[i] == karakter )
16         sayackarakter++ ;
17 }
18 printf("Bu metinde %d adet %c var",sayackarakter,karakter);
19

```

11-13 : Kullanıcıdan bir harf alıyoruz ve bunu “karakter” isimli değişkenimize kaydediyoruz.

14-17 : Bir döngü oluşturup tüm dizi içinde kullanıcıdan alınan harfe eşit olan bir harf var mı diye kontrol ediyoruz. Eğer eşitlik sağlanırsa “sayackarakter” isimli değişkende bunu topluyoruz.

18 : En sonda ise bu toplamı bastırıyoruz.

```

20 //toplam harf sayısını veren kodlar
21
22 hecesayisi = strlen(charArray);
23 printf("\nBu metinde %d adet hece vardır.",hecesayisi);
24

```

21 – 23 : strlen(charArray) ile ‘charArray’ isimli dizimizin boyutunu buluyoruz. Bunu “hecesayisi ” isimli değişkenimize atıp bunu ekrana basıyoruz.


```

25 //En çok geçen sesli heceyi veren kodlar
26 for(i=0 ; i<10 ; i++){
27     for(k=0; charArray[k] !='\0' ; k++){
28         if (sesli[i] == charArray[k]){
29             sayac++;
30         }
31         else
32             continue;
33     }
34     if(sayac > max1){
35         max1 = sayac;
36         maxsesli = sesli[i];
37     }
38     sayac = 0;
39 }
40 printf("\nEn çok geçen sesli harf '%c' ve %d adet gecti",maxsesli,max1) ;
41

```

- 26-39 : En çok bulunan sesli harfi bulmak için sesli harfleri tanımladığımız dizi içerisindeki her harfi metin ile karşılaştıracğız. İlk for döngüsü ile sesli dizimiz içinde ikinci for döngümüz ile metin içerisinde dolaşyoruz. Yani sesli dizimizdeki ilk terim olan 'A' ile tüm metni karşılaştırıyoruz. Eşleşen değerleri sayaç değişkenimizde tutuyoruz. En büyük değeri ise max1 isimli değişkenimizde tutuyoruz. Daha sonra (34) max1 değişkenimiz ile sayacımızı karşılaştırıyoruz. Eğer sayaç değerimiz büyükse "max1" değerimize sayaç değerini atıyoruz(35). Harfi ise "maxsesli" değişkenimize atıyoruz(36). Metinde dolaştıktan ve döngüden çıkıp ikinci sesli harfimize geçmeden önce sayaç değerimizi sıfırlıyoruz(38).
- 40 : Son olarak döngü bittikten sonra oluşan maxsesli ve max değişkenlerimizi bastırıyoruz.

```

42 //En çok geçen sessiz harfi veren kodlar.
43
44 for(i=0 ; i<38 ; i++){
45     for(k=0; charArray[k] !='\0' ; k++){
46         if (sessiz[i] == charArray[k]){
47             sayac++;
48         }
49         else
50             continue;
51     }
52     if(sayac > max2){
53         max2 = sayac;
54         maxsessiz = sessiz[i];
55     }
56     sayac = 0;
57 }
58 printf("\nEn çok geçen sessiz harf '%c' ve %d adet gecti.",maxsessiz,max2);
59 }
60

```

44-58 : Bir önceki kısımda en çok geçen sesli harfi bulmak için yaptığımız işlemlerin aynısını bu sefer de sessiz harfleri bulmak için yapacağız. Fakat bu sefer doğal olarak sessiz harfleri tuttuğumuz dizi içerisinde gezinip verilen metin ile karşılaştıracakız ve karışmaması için farklı değişkenlerde tutacağız.

```

C:\Users\ASUS\Desktop\Soru3.exe
Lutfen bir harf giriniz:
a
Bu metinde 98 adet a var
Bu metinde 1495 adet hece vardır.
En çok geçen sesli harf 'e' ve 154 adet gecti
En çok geçen sessiz harf 'r' ve 92 adet gecti.
-----
Process exited after 1.736 seconds with return value 47
Press any key to continue . . .

```

Soru 4

```
Soru1.c Soru2.c Soru3.c Soru4.c Soru5.c Soru6.c
1  #include <stdio.h>
2  #include<conio.h>
3  #include<string.h>
4
5  char charArray [] = {"A computer is a machine that can be instructed to carry out sequences of arithmetic or logical operations automatically via
6  int i =0 , esle = 1 , sayac = 0 , heceboyutu ;
7
8  int main(){
9      printf("Kac karakter uzunluguna sahip kelimeleri bulmak istersiniz.");
10     scanf("%d",&heceboyutu);
11     for(i=0;charArray[i]!='\0';i++){
12         if(charArray[i] == ' ' && charArray[i+heceboyutu] == ' ' && charArray[i] != '.' && charArray[i] != '-' && charArray[i] != '(' && charArray
13             esle++;
14     }
15     printf("%d karakterlik %d adet kelime bulunmaktadır.",heceboyutu , esle);
16 }
```

1 – 3: Öncelikle kütüphanelerimizi tanımlıyoruz.

5 – 6: Daha sonra kullanacağımız değişkenleri tanımlıyoruz.

9 – 10: Kullanıcıdan kaç harfli kelimeler aradığı bilgisini alıyoruz ve bunu heceboyutu değişkenine atıyoruz.

11 : Ardından verilen metin üzerinde gezinecek bir döngü kuruyoruz.

12 : Metin üzerinde gezinirken boşlukları esas alıyoruz. Yani metin üzerindeki herhangi bir boşluk değerinden kullanıcının girdiği sayı kadar ilerideki terim de eğer boşluksa arada kalan bölge bize kelime uzunluğunu verir. Bu durumu sağlayanları ise esle değişkeninde toplayabiliriz.

Soru 5

```
Soru1.c Soru2.c Soru3.c [*] Soru4.c Soru5.c Soru6.c
1  #include<stdio.h>
2  int f(int x , int y);
3  int x , y , sonuc ;
4  int main(){
5      printf("X degerini giriniz:");
6      scanf("%d",&x);
7
8      printf("\nY degerini giriniz:");
9      scanf("%d",&y);
10
11     printf("%d^%d isleminin sonucu = %d ",x , y , f( x,  y) );
12 }
13 int f(x , y ){
14     int c = 1 ;
15     while(y>=1){
16
17         c=x*c;
18         y=y-1;
19     }
20     return c ;
21 }
22
23
```

1-3 : Uygulamada kullanacağımız değişkenleri tanımlıyoruz.

4-11 : x ve y değerlerini kullanıcıdan alıyoruz. Sonucu ekrana basmak için daha sonra tanımlayacağımız fonksiyonu çağırıyoruz.

13-20 : Üs hesaplamak için özyinelemeli fonksiyonu kullanmalıyız. Girilen y(üs) değeri 1' e eşitlenene kadar taban değerimiz olan x ile c yi çarpıyoruz. C değerini döndürüyoruz.

```
C:\Users\ASUS\Desktop\Soru5.exe
X degerini giriniz:2
Y degerini giriniz:5
2^5 isleminin sonucu = 32
-----
Process exited after 2.811 seconds with return value 26
Press any key to continue . . .
```

Soru 6

Soru1.c Soru2.c Soru3.c [*] Soru4.c Soru5.c [*] Soru6.c

```
1  #include<stdio.h>
2  int fibonacci(int );
3  int cagrilma(int);
4  int x;
5  int main(){
6      int sonuc , sonuccagrilma;
7
8      printf ("Fibonacci serisinin kacinci elemaninin hesaplanacagini giriniz:");
9      scanf("%d",&x);
10
11      sonuc = fibonacci(x);
12      printf ("\n%d. elemanin degeri : %d", x , sonuc);
13
14      sonuccagrilma = 2*cagrilma(x) - 2;
15      printf("\nSerinin %d. elemanini hesaplamak icin fonksiyon cagrilma sayisi : %d ",x , sonuccagrilma);
16
17 }
```

1-4 : Daha sonra kullanacağımız değişkenleri tanımlıyoruz.

6 : Main fonksiyonu içinde kullanacağımız değişkenleri yazıyoruz.

8 – 12 : Kullanıcıdan bir sayı girmesini istiyoruz ve bu sayıyı x değişkenine atayıp , bu sayı kadar adımdaki fibonacci Değerini hesaplamaya çalışıyoruz.

14 -15 : Fonksiyonun girilen değer kadar adımını hesaplarken kaç kez çağırıldığını hesaplama işlemini buradan çağırıyoruz.

```

18 int fibonacci(int x ){
19
20     int a , b;
21
22     if (x<=1)
23         return x;
24
25     else{
26         a = fibonacci( (x-1) ) ;
27         b = fibonacci( (x-2) ) ;
28     }
29     return ( a + b);
30 }

```

18 -30 : fibonacci isimli fonksiyonumuzun içine kullanıcıdan gelen x değerini atarak çalıştırıyoruz.

Fonksiyon içinde a ve b değişkenleri tanımlıyoruz(20). Ardından gelen x değerinin 1 den büyük olup olmadığına bakıyoruz. Eğer x 1'den küçükse fonksiyon x'i döndürecek(23).

Eğer fonksiyona gönderilen değer 1 den büyükse o zaman "a" değişkenimize fonksiyonun bir önceki değeri (x-1 için) b değişkenimize ise iki önceki değerini (x-2 için) atıyoruz. Böylece fonksiyon f(x) değerini bulmak için önce f(x-1) , sonra f(x-2) değerini buluyor . Sonra bu değerleri toplayıp döndürüyor.

$$f(x) = f(x-1) + f(x-2)$$

$$f(x-1) = f(x-2) + f(x-3)$$

. . .

$$f(2) = f(1) + f(0)$$

```

31 int cagrilma(x){
32     int k , l , g;
33
34     if(x<=1)
35         return 1 ;
36     else{
37         k = cagrilma( (x-1) );
38         l = cagrilma( (x-2) );
39     }
40
41     return (k + l );
42
43 }
44

```

31-43 : Aynı şekilde fonksiyon çalıştığında farklı değerler için fonksiyon çağırıyor. Fonksiyon çağırma sayısını Kontrol ettiğimizde özyinelemeli fonksiyona çok benzediğini görüyoruz.

2 için = 2 kez çağırılıyor

3 için = 4 kez çağırılıyor

4 için = 8 kez çağırılıyor

5 için = 14 kez çağırılıyor

6 için =24 kez çağırılıyor

Her değeri 2 ye bölersek 1 , 2 , 4 , 7 ve 12 sayılarını elde ederiz. Bu da fibonacci dizisine benziyor.

$$1 + 2 = 4 - 1$$

$$2 + 4 = 7 - 1$$



Tek fark 1 eksikliğine eşit olması. Bu yüzden 14.satırda sonucumuzu $2 * \text{cagrilma}(x) - 2$ değerine eşitliyoruz.

 C:\Users\ASUS\Desktop\Soru6.exe

Fibonacci serisinin kacinci elemaninin hesaplanacagini giriniz:10

10. elemanin degeri : 55

Serinin 10. elemanini hesaplamak icin fonksiyon cagrilma sayisi : 176

Process exited after 2.975 seconds with return value 71

Press any key to continue . . . 