

BMM 111

Bilgisayar Programlama-I

6. Uygulama

Dr. Öğr. Üyesi Mustafa İSTANBULLU

Çukurova Üniversitesi
Mühendislik Fakültesi
Biyomedikal Müh. Böl.

E-mail: mm.istanbullu@gmail.com

Not: Slaytlar, kaynakça bölümünde verilen listeden faydalanılarak hazırlanmıştır.

Döngülerde Yapılan Temel Hatalar

- Bu bölümde son olarak programlarda sıklıkla yapılan hatalardan ve yanlış yapılandırılmış döngülerden bahsetmek istiyoruz.

'{' ve '}' Ayraçlarının Unutulması

- Programcı, $f(x) = x^2 + 1$ fonksiyonunu her x değeri için 0.0 dan 2.0'a kadar 0.1 adımlarla hesaplamak ve tablo olarak ekranda görmek istemektedir.
- Programımız aşağıdaki gibi olursa,

```
printf("  x    f(x)\n");  
for (x=0.0, x<=2.0; x=x+0.1)  
    f=x*x+1;  
printf("\n%5.2lf%5.2lf", x, f);
```

- Çıktı aşağıdaki gibidir:

x	f(x)
2.00	5.00

- Bu programda her **x** değerine karşı **f** değerleri döngü ile doğru hesaplanırsa da, çıktıda sadece **x**'in ve **f**'in son bulunan değerleri görüntülenir.
- Programda doğru girintili yazım kullanılmasına rağmen **'{'** ve **'}'** ayraçları unutulmuştur. Dolayısıyla döngü gövdesi sadece **'f=x*x+1;'** komutundan oluşmaktadır.
- Doğru program parçası aşağıda verilmiştir.

```
printf("  x    f(x)\n");  
for (x=0.0, x<=2.0; x=x+0.1)  
{  
    f=x*x+1;  
    printf("\n%5.2lf%5.2lf",x,f);  
}
```

`;' Hatalı Kullanımı

- Bu örnekte, 5'den 100'e kadar ve 5'in katları olan tamsayıların toplamını $5+10+15+ \dots +95+100$ bulan ve ekranda görüntüleyen program parçasını yazalım.

```
toplam=0;
for (i=5;i<=100; i=i+5);
    toplam+=i;
printf("Toplam=%d",toplam) ;
```

- Bu program yürütüldüğünde çıktı aşağıdaki gibidir.

Toplam=105

- Bu sonuç istenen eylemi yansıtmamaktadır. İlk bakışta program doğru gibi görünse de ufak bir hata nedeniyle yanlış sonuç elde edilmiştir.
- Bunun nedeni **for** komutunun sonundaki yanlışlıkla konulmuş olan **;** işaretidir.
- Böylece döngünün gövdesinde hiçbir komut yer almamaktadır.
- Program üzerindeki **;** işaretinin kaldırılmasıyla doğru çıktı elde edilecektir.

'=' Hatalı Kullanımı

- Atama operatörü "=" genellikle, ilişkisel operatörü "==" ile çok karıştırılmaktadır.
- Aşağıdaki örnekte yine 5'den 100'e kadar ve 5'in katları olan tamsayıların toplamını bulalım.

```
toplam=0;
i=5;
do
{
    toplam+=i;
    i+=5;
}
while(i=100);
printf("Toplam=%d",toplam);
```

- Bu programda **while**'in koşul ifadesinde "**i=100**" yer almaktadır.
- Bu bir atama ifadesi olup **100** değeri **i** değişkenine atanır. Dolayısıyla, ifadenin değeri sıfırdan farklı olan **100**'dür.
- Yani, hep doğru sonucu veren bir ifadedir. Hiçbir zaman sıfır değerini almayacak ve sonsuz bir döngü elde edilecektir.
- Doğru sonuç almak için atama komutu yerine 'eşit mi?' sorusunu sormamızı sağlayan **'=='** kullanılmalıdır. Dolayısıyla, '**i=100**' koşul ifadesi '**i==100**' ifadesi olarak değiştirilmelidir.

Uygulama 1:

10 reel sayıyı girdi olarak alan ve pozitif sayıların toplamını bulan C kodunu yazınız.

Uygulama 1:

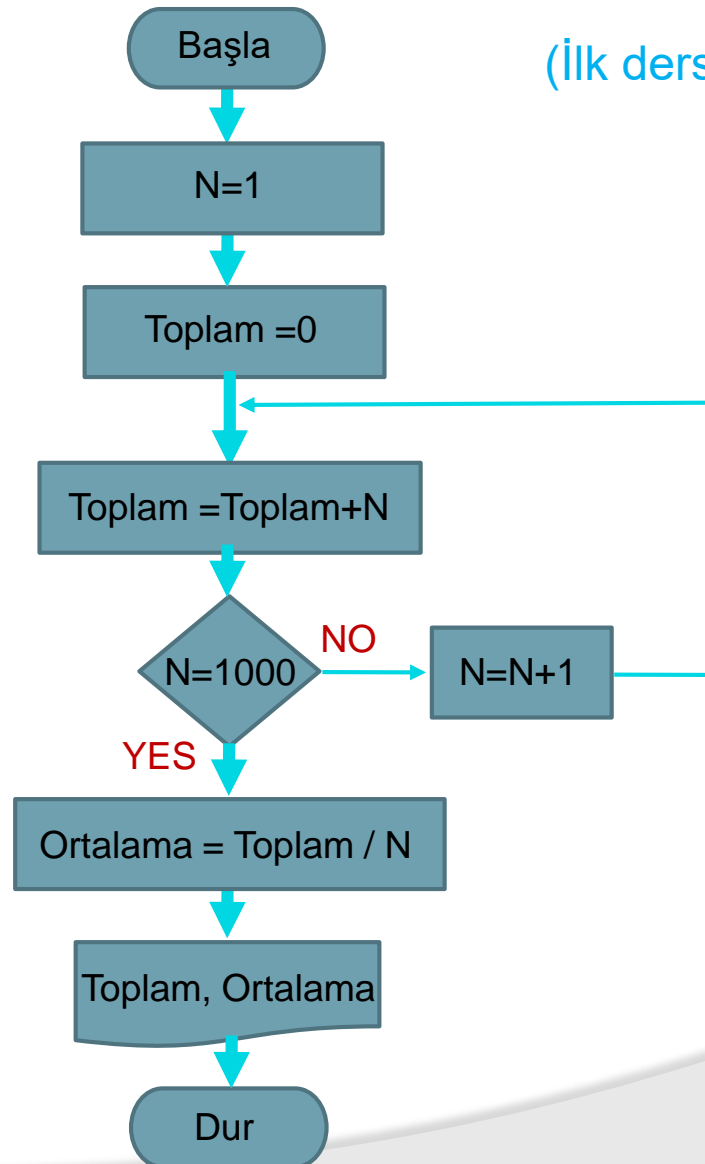
```
#include <stdio.h>      /*Hafta6_1.c */
int main(void)
{
    int i;
    float sayi,toplam;
    i=1;
    toplam=0.0;
    while(i<=10)
    {
        /*reel sayinin girilmesi*/
        printf("%d.sayiyi giriniz:",i);
        scanf("%f",&sayi);
        /*sayinin pozitif olup olmadiginin kontrolu*/
        if(sayi>0)
            toplam=toplam+sayi;
        i=i+1;
    } /*while komutunun sonu*/
    printf("Pozitif sayilarin toplami:%5.2f",toplam);
    system("pause");
    return(0);
}
```

Uygulama 2: 1'den 1000'e kadar olan tam sayıların toplamlarını ve ortalamasını bulan bir program yazılmak istenmektedir.

- a) Gereken akış diyagramını çiziniz. (İlk derslerde yapıldı)
- b) Akış diyagramını uygulayan programı C dilinde yazınız.

Uygulama 2:

(İlk derslerde yapıldı).



6. Bölüm

Döngü Uygulamaları

Uygulama 2:

```
⦿ #include <stdio.h> /*Hafta6_2.c */
⦿ /* Bu program 1 den 1000 kadar olan tam sayıların toplamını ve ortalamasını alıp ekrana yazar. */
⦿ int main(void)
⦿ {
⦿     int i;
⦿     long toplam=0; /* long integer*/
⦿     double ortalama;

⦿     for(i=1; i<=1000;i++)
⦿     {
⦿         toplam += i; /* toplam=toplam+i */
⦿     } ;
⦿     printf("1 den 1000 kadar olan sayıların toplamı: %d \n",toplam);
⦿     printf("\n");
⦿     ortalama = toplam / 1000.0;
⦿     printf("1 den 1000 kadar olan sayıların ortalaması: %lf \n",ortalama);
⦿     printf("\n");
⦿     system("pause");
⦿     return 0;
⦿ }
```

Uygulama 3: $f(x)$ fonksiyonu, x 'in çeşitli değerlerine göre aşağıdaki şekilde değişmektedir. x değeri, 0 ile 10 arasında 0.5 aralıklarla arttığına göre her bir x değeri için $f(x)$ fonksiyonunu hesaplayan, x ile $f(x)$ değerlerini ekrana yazan bir program için :

- a) Akış şemasını çiziniz. (İlk derslerde yapıldı)
- b) Programı C dilinde yazınız.

$$0 \leq x \leq 2$$

$$f(x) = x$$

$$2 < x \leq 3$$

$$f(x) = x^3 - x^2 - 2$$

$$3 < x \leq 4$$

$$f(x) = x^2 - 2x + 13$$

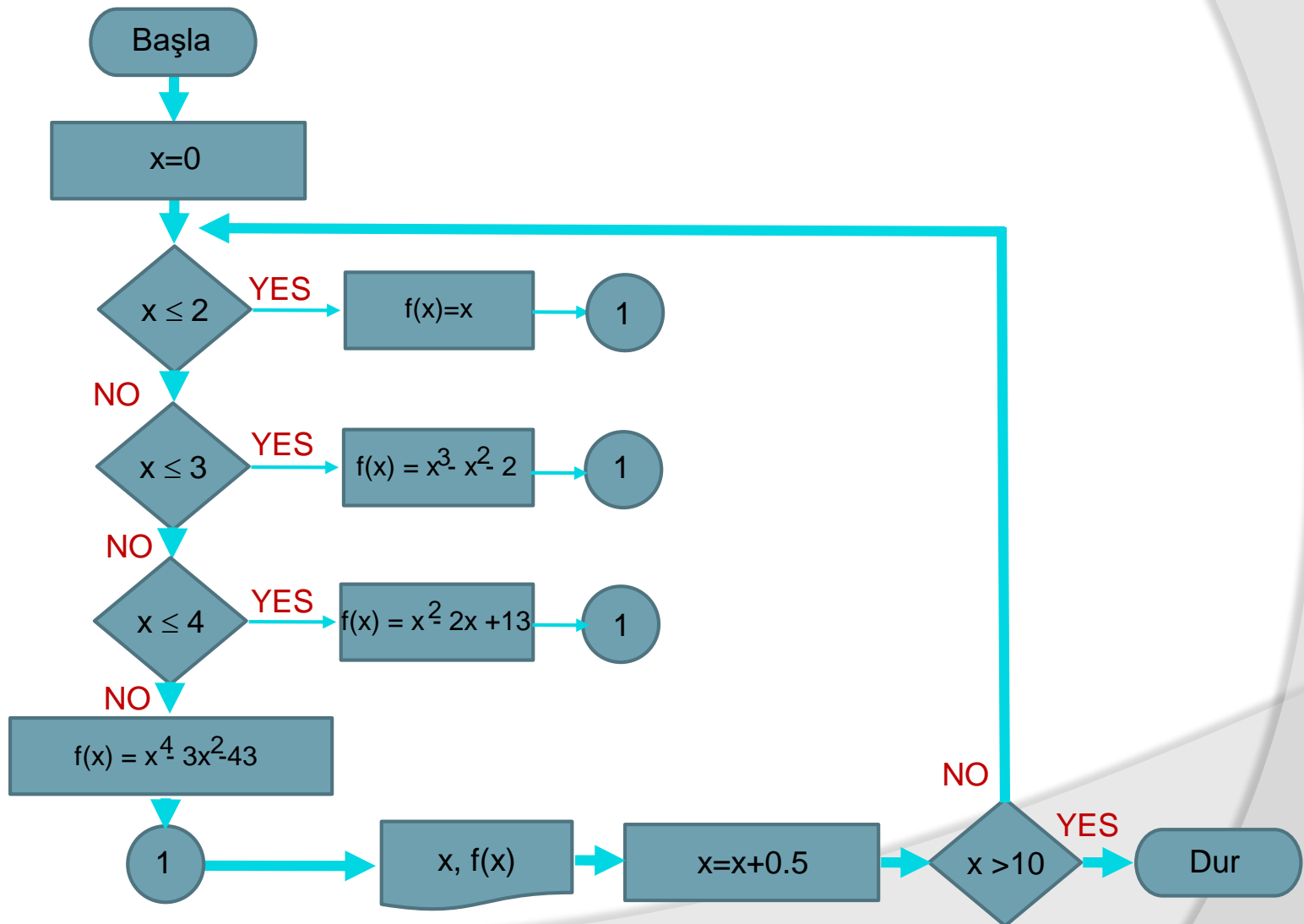
$$4 < x$$

$$f(x) = x^4 - 3x^2 - 43$$

6. Bölüm

Döngü Uygulamaları

(İlk derslerde yapıldı).



```

○ #include <stdio.h> /*Hafta6_3.c*/
○ #define step 0.5
○ int main(void)
○ {
○     float x=0;
○     double fx=0;
○     printf("This program calculates output of a piewise function" );
○     printf("\n");
○     while (x<=10.0)
○     {
○         if ((x<=2.0) && (x>=0.0)) fx=x;
○         if ((x<=3.0) && (x>2.0)) fx=(x*x*x-x*x-2);
○         if ((x<=4.0) && (x>3.0)) fx=(x*x-2*x+13);
○         if (x> 4.0) fx=(x*x*x*x-3*x*x-43);
○         printf(" Argument: %.2f   Function Value: %.2lf",x,fx);
○         printf("\n");
○         x=x+step;
○     }
○     system("pause");
○     return 0;
○ }
```

Uygulama 4 : e^x fonksiyonunun seri olarak açılımı aşağıda verilmiştir.

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \dots$$

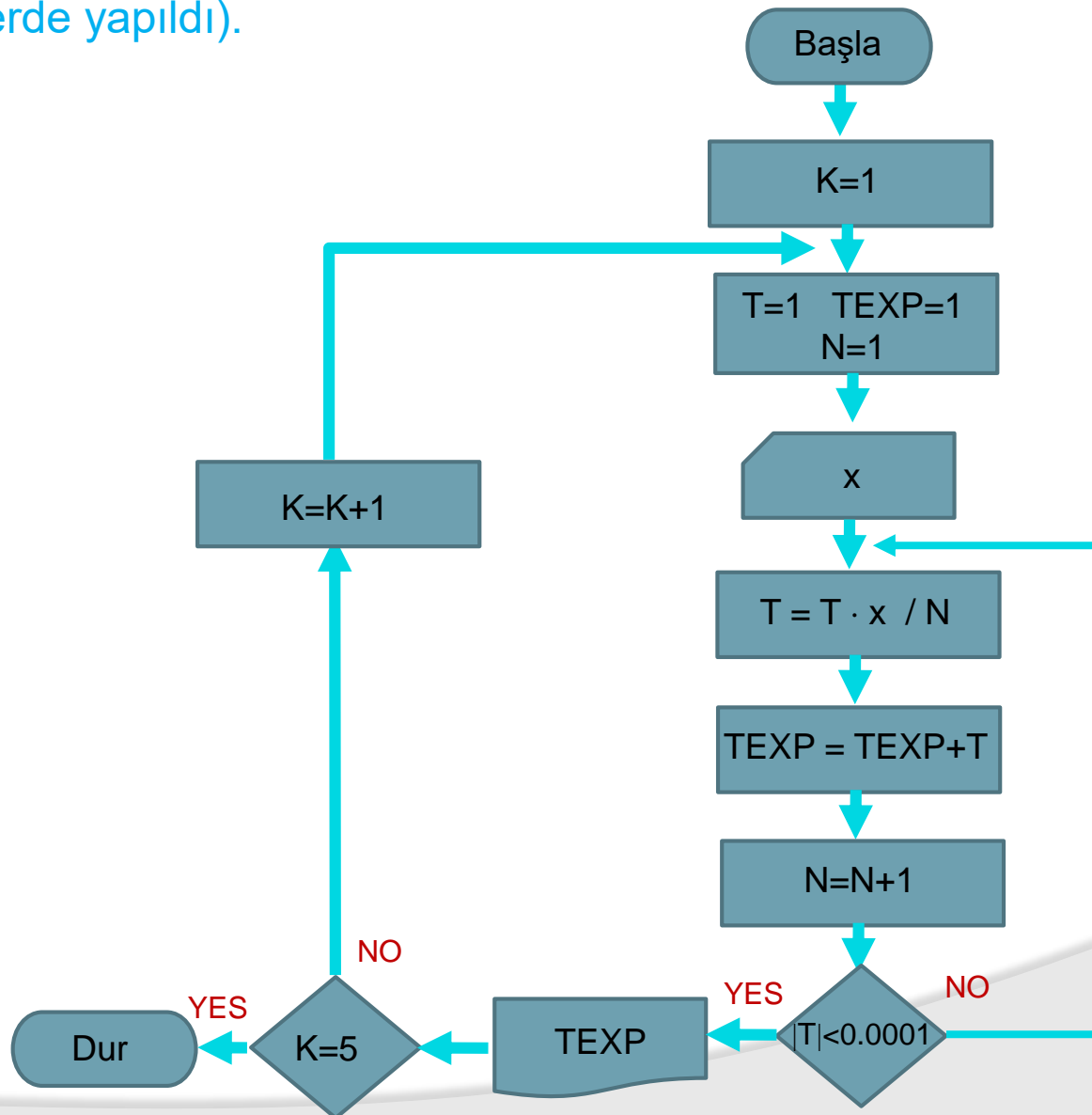
x 'in değişik 5 değeri için e fonksiyonunu hesaplayan bir program akış şeması çiziniz (İlk derslerde yapıldı). Gereken C programını yazınız. Serinin son teriminin mutlak değeri 0.0001'den küçük olana kadar seriyi hesaplamaya devam ettiriniz.

Not: Bu örnek için x 'in değerlerinin, $-1 < x < 1$ aralığında olduğunu kabul ediniz.

6. Bölüm

Döngü Uygulamaları

(İlk derslerde yapıldı).



6. Bölüm

Döngü Uygulamaları

Uygulama 4:

```

⦿ #include <stdio.h>    /*Hafta6_4.c*/
⦿ int main(void)
⦿ {
⦿     int k,n;  float x;  double T, Tex;
⦿     for (k=1;k<=5;k++)
⦿     {
⦿         T=1.0;    n=1;    Tex=1.0;
⦿         do /* Enforcing user to enter correct range of variable */
⦿         {
⦿             printf("Enter the %d. argument , (between -1 and 1) \n",k);
⦿             scanf("%f",&x);
⦿             } while( (x>1) || (x<-1));
⦿         printf(" x:%f \n",x);
⦿         while ( (T > 0.00001))
⦿         {
⦿             T=T*x/n;
⦿             Tex=Tex+T;
⦿             n++;
⦿         } ;
⦿         printf(" Exp(%3.2f):%lf",x, Tex);
⦿         printf("\n");
⦿     }
⦿     system("PAUSE"); return 0;
⦿ }
```


6. Bölüm

Döngü Uygulamaları

Uygulama 5:

```
#include <stdio.h>
#define PI 3.141592654
int main(void)
{
    int cevap; double a,r,alan;
    do
    {
        printf("\n 1. Kare"); /*MENÜNÜN GÖRÜNTÜLENMESİ*/
        printf("\n 2. Daire");
        printf("\n 3. Bitti");
        printf("\n\nSeceneginizi belirtiniz:");
        scanf("%d",&cevap);
        switch(cevap)
        {
            case 1: /*Karenin alaninin hesaplanmasi*/
                printf("\nKarenin kenar uzunlugunu giriniz:");
                scanf("%lf",&a);
                alan=a*a;
                printf("Karenin alani:%.2f\n",alan);
                break;
            case 2: /*Dairenin alaninin hesaplanmasi*/
                printf("\nDairenin yaricap uzunlugunu giriniz:");
                scanf("%lf",&r);
                alan=PI*r*r;
                printf("Dairenin alani:%.2f\n",alan);
            }
        } while(cevap!=3);
        system("pause");
        return(0);
    }
```

6. Bölüm

Döngü Uygulamaları

```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

```
    int pos=0,neg=0,sifir=0,n,sayi,i;
```

```
    printf("Kac tane sayi gireceksiniz:");
```

```
    scanf("%d",&n);
```

```
    /*Döngü içinde n tane sayının girilmesi ve işlenmesi*/
```

```
    for(i=1;i<=n;i=i+1)
```

```
    {
```

```
        printf("%d. sayi: ",i);
```

```
        scanf("%d",&sayi);
```

```
        /* Girilen sayının pozitif, negatif veya sıfır olup olmadığının belirlenmesi*/
```

```
        if(sayi>0)
```

```
            pos++;
```

```
        else if(sayi<0)
```

```
            neg++;
```

```
        else sifir++;
```

```
    }
```

```
    /*Sonuçların gösterilmesi*/
```

```
    printf("\n%d adet pozitif sayi.\n",pos);
```

```
    printf("%d adet negatif sayi.\n",neg);
```

```
    printf("%d adet sıfır sayısı.\n",sifir);
```

```
    system("pause");
```

```
    return(0);
```

```
}
```

Uygulama 6:

6. Bölüm

Döngü Uygulamaları

Uygulama 7:

```
# include <stdio.h>
```

```
// Bu program iç içe çift "for" döngüsü kullanım örneğidir. C.V.Baysal
```

```
// Sayıları Tablo şeklinde yazdırır.
```

```
int main(void)
```

```
{
```

```
    int i,j,k,x;
```

```
    printf("Bir tamsayı giriniz:");
```

```
    scanf("%d", &x);
```

```
    printf("\n\t\t");
```

```
    for(k=1;k<=x;++k)
```

```
        printf("%2d.sutun",j);
```

```
    printf("\n");
```

```
    for (i=1;i<=x;++i)
```

```
    {
```

```
        printf("\n%2d.satir :t",i);
```

```
        for(j=0;j<x;++j)
```

```
            printf("%6d\t",j-i);
```

```
        printf("\n");
```

```
    }
```

```
    getch();
```

```
    return (0);
```

```
}
```

KAYNAKÇA

- Prof.Dr. İbrahim DEVELİ, Bilgisayar Programlama Ders Notları, Erciyes Üniv. Elektrik-Elektronik Müh. Böl.
- H.Turgut UYAR, Programlamaya Giriş Ders Notları,İTÜ, 2004.
- Fedon KADİFELİ,Standart C Programlama Dili, (Tercüme),2000.
- Doç. Dr. Soner ÇELİKKOL, Programlamaya Giriş ve Algoritmalar, Murathan Yayınevi, TRABZON; 2009
- N. Ercil Çağıltay ve ark., C DERSİ PROGRAMLAMAYA GİRİŞ, Ada Matbaacılık, ANKARA; 2009.
- Çeşitli kişilerin internette paylaşımına açtığı notlardan faydalanılmıştır.