

18.6.2020

The logo for C Programming is centered on the page. It features a large, stylized white letter 'C' inside a blue hexagon. The hexagon has a 3D effect with a darker blue shadow. The entire logo is set against a solid blue rectangular background. A horizontal grey bar with a black segment on the left passes over the logo, serving as a backdrop for the title text.

C PROGRAMLAMA & ÖRNEKLER

YAZAR: | Doğan AY

İçindekiler

C programlama dilinin genel yapısı, veri tipleri değişkenler.....6

1. [Değişken Bildirimi](#)
2. [Aritmetik Operatörler](#)
3. [Atama Operatörler](#)
4. [İlişkisel Operatörler](#)
5. [Mantıksal Operatörler](#)

Örnekler:

- [1_1_ornek.c](#)
- [1_2_ornek.c](#)
- [1_3_ornek.c](#)

Program denetim ve döngü yapıları11

1. [İf Else İf Else](#)
2. [For](#)
3. [While](#)
4. [Switch Case](#)
5. [Do While](#)

Örnekler:

- [2_1_ornek.c](#)
- [2_2_ornek.c](#)
- [2_3_ornek.c](#)
- [2_4_ornek.c](#)
- [2_5_ornek.c](#)

Önişlemci komutları.....15

Örnekler:

- [3_1_ornek.c](#)
- [3_2_ornek.c](#)
- [3_3_ornek.c](#)

Diziler.....17

1. Tek Boyutlu Diziler
2. [Çift Boyutlu Diziler](#)
3. [Çok Boyutlu Diziler](#)
4. [String Dizileri](#)

Örnekler:

- [4_1_ornek.c](#)
- [4_2_ornek.c](#)
- [4_3_ornek.c](#)
- [4_4_ornek.c](#)

Fonksiyonlar.....21

1. [Fonksiyon Tanımı](#)
2. [Fonksiyon Tipleri](#)
3. [Fonksiyon Tanımlama Kuralları](#)
4. [İtaratif Fonksiyonlar](#)
5. [Rekürsif Fonksiyonlar](#)
6. [Fonksiyonlarda Diziler](#)

Örnekler:

- [5 1 ornek.c](#)
- [5 2 ornek.c](#)
- [5 3 ornek.c](#)
- [5 4 ornek.c](#)
- [5 5 ornek.c](#)
- [5 6 ornek.c](#)
- [5 7 ornek.c](#)

Göstericiler(pointer)..... 32

1. Pointer Bildirimi
2. Adres Operatörü
3. [İçerik indirection operatörü](#)
4. [Pointer Dizi](#)
5. [Pointerda fonksiyonlar](#)

Örnekler:

- [6 1 ornek.c](#)
- [6 2 ornek.c](#)
- [6 3 ornek.c](#)
- [6 4 ornek.c](#)

Dinamik bellek yönetimi.....38

- Dinamik Dizilerde Fonksiyon

Örnekler:

- [7 1 ornek.c](#)
- [7 2 ornek.c](#)

Karakter Dizileri (Strings)..... 41

- Bazı Katar Fonksiyonları

Örnekler:

- [8 1 ornek.c](#)
- [8 2 ornek.c](#)
-

Matematiksel İşlemler.....45

Örnekler:

- [9 1 ornek.c](#)
- [9 2 ornek.c](#)

Yapılar Structür.....49

1. [Bit Alanı Field](#)
2. [Struct'ların Fonksiyonlarla Kullanımı](#)

Örnekler:

- [10_1_ornek.c](#)
- [10_2_ornek.c](#)
- [10_3_ornek.c](#)
- [10_4_ornek.c](#)
- [10_5_ornek.c](#)
- [10_6_ornek.c](#)

Dosya Yönetimi.....57

1. Dosya Açma Modları
2. [Dosyalama Fonksiyonları](#)
3. [Binary Dosya Fonksiyonları](#)
4. [Metin ve Binary dosyalama farkları](#)

Örnekler:

- [11_1_ornek.c](#)
- [11_2_ornek.c](#)
- [11_3_ornek.c](#)
- [11_4_ornek.c](#)
- [11_5_ornek.c](#)
- [11_6_ornek.c](#)
- [11_7_ornek.c](#)
- [11_8_ornek.c](#)

Değerlendirme.....73

Ünite Değerlendirme Cevapları:

- [cevap1.c](#)
- [cevap2.c](#)
- [cevap3.c](#)

[Allegro Kütüphanesi İle 2d Oyun.....89](#)

[Yazar Hakkında bilgi.....91](#)

[Kaynakça.....92](#)

Önsöz:

C programlama dili günümüzde en çok kullanılan dillerden biridir. Bilgisayar mühendisliği 1.sınıf öğrencileri ve liselerde bu ders müfredatta yer alır .C programlama dilinin amacı programlama mantığını öğretmek ve kavratmaktır. Bu programlama dili ile ortaya somut şeylerde çıkarılabilir. C dili ile driver, grafik ve oyun, gömülü sistemler, işletim sistemi, programlama dili (PHP), hesaplama platformları yapılabilir. Bu işlemleri düzey olduğu için ben kitabımda daha basit düzey yani üniversitelerde ve liselerde ki müfredatı kadarıyla anlatmaya çalıştım daha çok örneklerde verdim ve örneklerde bu konuyla ilgili açıklama satırları ekledim örneklere ulaşmak için kaynakça bölümüne github hesabımı ekliyorum buradan örneklerin. Unutmayalım ki bir programlama dili ancak çok örnek yaparak akılda kalıcı olabilir.

1. C Programlama Dilinin Genel

Yapısı, Veri

TİPLERİ, DEĞİŞKENLER, OPERATÖRLER

C dili makine diline yakın bir dildir. Hem donanıma hitap eder, hem de uygulama geliştirmede kullanırız. C dili büyük harf küçük harf duyarlılığı olan bir dildir. Her komutun bittiği yere “;” konulur. Bu onun syntax kuralıdır. İleride göreceğimiz if, else if, else, for, while, switch(degiskenadi) gibi blokların başlangıcında ve bitişinde “{ ve }” kullanılır. Kodlar bir ana fonksiyon içinde derlenir. Kodlara yorum satırı eklemek ya da açıklama yapmak için “//tek satırlık bir yorum için” ya da “/* yorumunuz yazılır */” kullanılır. Standart bir c iskeleti yapımı:

```
#include <stdio.h> //standart giriş çıkış işlemleri yapılır printf scanf gibi fonksiyonlar çalıştırılır.  
int main(void) //Ana iskelet şekli bu haldedir kodlar bu iskeletin içine yazılır.  
{  
printf(“Merhaba Karaman”); //burada printf ekrana birşey yazdırmak için kullanılır  
return 0; //return fonksiyonu çağrıldığı yere geri dönülmesini sağlar eğer return 0; dan sonra  
kod eklenirse bu kod devre dışı kalır yazılmamış gibi olur.  
}
```

Veriler kullanıcadan alımlı ya da daha önceden tanımlı olabilir.

Kullanıcıdan alırken verileri <stdio.h> yani standart kütüphane fonksiyonlarından scanf(); komutunu kullanırız içine bazı değerler koyarız. Ekrana birşey yazdırmak için ise printf(“”) ; //fonksiyonunu kullanırız.

Tanımlı ya da klavyeden girilmiş bir veriyi yazdırmak için ise;
printf(“veri türü integer için =%d”,degisken_ismi);

Kullanımı:

scanf(“%d”,°isken_ismi); //burada &isareti belleğin geçici bir şekilde hafızada tutulmasını sağlar daha sonra işleyeceğimiz pointerlar bu alanı işaret eder ve bu sayede bellekte yer ayırabileceğiz. %d işareti ise veri türüne aittir. Her veri türüne göre bu değişir Veri tipi tablosunda bunlara değineceğiz.

[Belgenin Başı](#)

Veri tipi tablosu bkz.

Veri tipi	Karşılığı	Değer olarak almak	Türkçe Karşılığı	Kullanıcı Tanımlaması
Tam sayılar	(integer) int	%d	Sayısal	int a =5;
Short int	short int	%hd	Kısa tam sayı	short int=-4;
Long int	long int	%ld	Uzun tam sayı	long int=34;
Unsigned int	int	%u	İşaretsiz tam sayı Pozitifdir daima.	int=442; //pozitif değer alır
Karakterler	Char,string	%c,%s	Karakter değeri	char harf='a';
Mantıksal	(Boolean) bool		Yanlış ya da doğru	bool key =true;
Kesirli Sayılar	float,double	%f,%lf	Ondalıklı sayılar	float sayı=10.4;

Değişken Bildirimi(Declaration):

Bir veriye anabellekte bir yer ayırma eylemidir. Ayrılan yere bir veri yazılsın ya da yazılmasın değişken etkin kaldığı sürece, ayrılan anabellek adresine başka bir şey yazılmaz. Değişken adı, aslında bellekte o adresi işaret eden bir işaretçidir (pointer).

Değişken Tanımlama Kuralları:

- Değişken sayı ile başlamaz. Örnek= int 232=23; //yanlış tanımlama
- Değişken başlangıcında ya da bitişinde “_” kullanılabilir. Örnek=int _yas=23; char harf_ =’a’;
- Değişken isimleri büyük harfle başlayabilir.
- C dili komutları değişken adı olarak verilemez.Örneğin (if,not,else)
- Değişken ismi birden çok kelimeden oluşuyorsa arasına boşluk konulmaz.Örnek=futbolcu adı;
- İçinde özel karakterler (“#,\$,&,ö,ş,ı”v.b) kullanılmaz.

1_1_ornek.c)Veri tiplerine değişkenler atayın ve hafızada kapladığı yeri ekranda gösteriniz bunun için sizeof parametresinden yararlanın.

```
#include <stdio.h>
int main()
{
    int sayi1=12; //burada veri tiplerimize değişkenler atayarak onlara sayı verdik
    double sayi2=10.43;
    float sayi3=13.23;
    char harf='b'; //char veri tipine karakter verdik
    unsigned int sayi5=32;
    printf("intin sayının ekranda yazımı=%d ve hafızada kapladığı yer=%d\n",sayi1,sizeof(int)); //sayıların
    ekranda yazımını yaptık ve onların hafızada kapladığı yerleri bulduk sizeof parametresi ile
    printf("doublein sayının ekranda yazımı=%f ve hafızada kapladığı yer=%d\n",sayi2,sizeof(double));
```

```
printf("float sayinin ekranda yazimi=%f ve hafizada kapladigi yer=%d\n",sayi3,sizeof(float));
printf("unsigned sayinin ekranda yazimi=%d ve hafizada kapladigi yer=%d\n",sayi5,sizeof(unsigned));
printf("Char degerin ekranda yazimi=%c ve hafizada kapladigi yer=%d\n",harf,sizeof(char));
return 0;
}Belgenin Başı
```

Operatörler:

C dilinde operatörler 5 e ayrılır.

- Aritmetik Operatörler,
- Atama Operatörleri,
- İlişkisel Operatörler,
- Mantıksal Operatörler,
- Bitsel operatörler.

1.Aritmatik Operatörler:Matematiksel işlemleri yapmak için kullanırız.

Aritmetik İşlem	Kullanımı	Sonucu
+ (Toplama)	int x=5 ,int y=7;x+y;	x+y=12
- (Çıkarma)	int x=5 ,int y=7;x-y;	x-y=-2
* (Carpma)	int x=5 ,int y=7;x*y;	x*y=35
/ (Bolme)	int x=35 ,int y=7;x/y;	x/y=5
% (Mod)	int x =35,int y=7;x%y;	x%y=0

1_2_ornek.c 2 sayiya 5 (toplama,cikarma,carpma,bolme,mod) islemi yapan program yazınız.”35,7” Sayılarini kullanın.

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

```
int main()
{
    int sayi1=35,sayi2=7;
    int topla,cikar,bol,carp,modu;
    topla=35+7;
    cikar=35-7;
    carp=35*7;
    bol=35/7;
    modu=35%7;
    printf("Toplami=%d\n",topla);
    printf("Cikarma=%d\n",cikar);
    printf("Bolme=%d\n",bol);
    printf("Carp=%d\n",carp);
    printf("Kalani=%d\n",modu);
    return 0;
}
```

[Belgenin Başı](#)

2.Atama Operatörleri:[Belgenin Başı](#)

Değişkenlerimize bir değer atamak için kullanılır.Atama işlemimiz sağdan sola olur.

Örneğin `int a=234;`

Operatör	Kullanımı	Açıklaması
=	<code>int a=6;</code>	A değişkenine 6 değerini atarız.
+=	<code>a+=23;</code>	A değişkeninin değerini 23 ekleyerek artırırız.
-=	<code>a-=12;</code>	A değişkenin değerini 23 eksilterek azaltırız.
=	<code>a=10;</code>	A değişkenin değerini 10 ile çarparak atarız.
/=	<code>a/=2;</code>	A değişkenin değerini 2'ye bölerek atarız.
%=	<code>a%=5;</code>	A değişkenin değerini 5.dereceden modunu alırız.

1_3_ornek.c)Tanımlı bir değişkene atama operatörlerini kullanarak değer atayan,ekleyen,çıkartan,çarpan,bölen program yazınız.

```
#include <stdio.h>
int main()
{
    int a;
    printf("Atama yapalım=%d",a=5);
    printf("\nArtırma yapalım=%d",a+=12);
    printf("\nEksiltme yapalım=%d",a-=1);
    printf("\nÇarpma yapalım=%d",a*=3);
    printf("\nBölme yapalım=%d",a/=2);
    printf("\nKalani bulalım mod=%d",a%=5);
    return 0;
}
```

3.İlişkisel Operatörler:[Belgenin Başı](#)

Bu operator 2 tane değişken arasındaki ilişkiyi ifade etmek için kullanılır.

Operatör	Kullanımı	Açıklama
>	<code>x>y</code>	X değişkeni y'den büyüktür.
<	<code>x<y</code>	X değişkeni y'den küçüktür.
>=	<code>x>=y</code>	X değişkeni y'den büyük

		veya eşittir.
<=	x<=y	X değişkeni y'den küçük ya da eşittir.
==	x==y	X değişkeni y'ye eşittir.
!=	x!=y	X değişkeni y'ye eşit değildir.

4.Mantıksal Operatörler:[Belgenin Başı](#)

Operatör	Kullanımı	Acıklama
&& (and) ve kapısı	if(a==1 && b==2)	Sayıların doğruluk değeri true yani 1 değeri döndürüyorsa aynı anda sonuç 1'dir.Değilse 0'dır
(or) veya kapısı	if(a==1 b==2)	Doğruluk değerlerinden 2 sinden 1'nin 1 olması sonucu 1 yapar.
!=(not) kapısı	if(!(a==1 && b==2))	Değilidir.

2.PROGRAM DENETİM VE DÖNGÜ YAPILARI:

2.a)İf Else if Else: [Belgenin Başı](#)

İf else if else bir işin değişik yönlerinden birini seçmek için kullanılır.İf doğru olduğu durumda.Else if değilse ,else ise hiçbirisi değilse döndürülür. { yay ayraç ile başlar } ile bitirilir.Tek satırlık bir kod varsa yay ayraça gerek yoktur.

2_1_ornek.c)Girilen hava sıcaklığına göre havanın sıcak olup olmadığını bulan program yazınız.(sicaklik>20 hava sıcaktır ve sicaklik>5 den büyükse ılık,hiçbiri değilse soğuktur)

```
#include <stdio.h>
int main()
{
    int sicaklik;
    printf("Hava sicakligini giriniz=");
    scanf("%d",&sicaklik);
    if(sicaklik>20)
    {
        printf("Hava sicaktır");
    }
    else if(sicaklik>5 && sicaklik<20)
    {
        printf("Hava ılıktır");
    }
    else
    {
        printf("Hava soguktur");
    }
    return 0;
}
```

2.b)For Döngüsü: [Belgenin Başı](#)

Döngü yapmak için kullanılır.Belirli bir sınırdan başlayarak belirli değerlerde artarak istenen sınıra gelene kadar bir işlemi yapmayı sağlar.Komut yapıları { yay ayraç içinde ifade edilir }.

For döngüsü yapısı

```
for(başlangıç,bitiş,artış değeri){
komut yapıları;
}
```

2_2_ornek.c)For döngüsü kullanarak E harfini yazan yıldızları yapınız.

```
#include <stdio.h>
int main()
{
    int i,j;
```

```

for(i=0;i<7;i++) //ilk for döngümüz sutun kısmını ifade eder
{
    for(j=0;j<7;j++) //ikinci for döngümüz satiri ifade eder
    {
        if(i==0 || i==3 || i==6) //eger satirimiz 0 indiste 3 indiste 6 indiste ise şart sağlanmış
olur bu şartımız e harfinin başlangıcını ortasını ve son noktasını belli eder.
        {
            printf("*"); //satir kismimiz j=0 dan başlayarak 7 ye kadar yıldızlarla dolar
        }
    }
    printf("*\n"); //eger if şartı sağlanmaz ise döngüye girilmez her sütüne 1 adet yıldız koyarak
bir alt satıra geçme işlemi yapılır "\n" ile.
}
return 0;
}

```

2.c)While Döngüsü:[Belgenin Başı](#)

While döngüsü şart sağlanırsa eğer sürekli ” { komut yapıları } “ içinde bulunan kod yapıları çalışır.While döngüsünde unutulmaması gerek kısım while içine tek koşul yazılır.Artırma işlemi döngünün içinde yapılır.

while(koşul)

```

{
komutlar;
}

```

2_3_ornek.c)F harfini yıldızlarla yapan kodu yazınız.

```

#include <stdio.h>
int main()
{
    int i=0,j=0; //while döngüsünde başlangıç değerlerini önceden atamalıyız
    while(i<5) //sütun kısmımızı belli eden döngü 5 sütünlük yıldız yapıcak
    {
        j=0; //j miz en son 5 olduğu için 2. ifademizin sağlanması için tekrar 0 dan başlatıyoruz
        while(j<5) //j<5 küçük olduğunda ve if şartı sağlanırsa döngü çalışacak
        {
            if(i==0 || i==2) // if şartı sağlanması için satirimiz 0 indiste ve 2 indiste olmalı
            printf("*");
            j++; //artma işlemi j<5 den küçük olana kadar devam edip i yi artırıp döngü tekrar
            devam edecek
        }
        i++;
        printf("*\n");
    }
    return 0; }

```

2.d)Switch –Case:[Belgenin Başı](#)

If deyimi gibidir.Birden çok seçenek arasında işlem yapmamızı sağlar.Kontrol deyimidir.

Kullanımı:

```

switch(ifade){
case ifade:

```

```

komutlar;
break; //döngüyü kırmamızı sağlar
case ifade2:
komutlar;
break;
default: //hiçbiri sağlanmazsa case ifadelerimizden bu döngüye girer
break;
}

```

2_4_ornek.c) Klavyeden girilen bir ayın baş harfi ile başlayan o ayları sıralayan program yazınız.

```

#include <stdio.h>
int main()
{
    char secenek;
    printf("Hangi harfle baslayan aylari gormek istersiniz buyuk harf giriniz=");
    scanf("%c",&secenek);
    switch(secenek) //secenek isleminde alinan degerin coklu secimler parametremizde varmı kontrol etmemize yarar
    {
        case 'A': //case ifadesinde karakter bir secim yapıldıysa secilen karakterin harfini 'A' gosteririz
            printf("A harfi ile baslayanlar=\nAralık\nAgustos");
            break; //dongu kırmak için kullanılırız
        case 'E':
            printf("E harfi ile baslayanlar=\nEylul\nEkim");
            break;
        case 'M':
            printf("M harfi ile baslayanlar=\nMart\nMayıs");
            break;
        case 'O':
            printf("O harfi ile baslayanlar=\nOcak");
            break;
        case 'N':
            printf("N harfi ile baslayanlar=\nNisan");
            break;
        case 'S':
            printf("S harfi ile baslayanlar=\nSubat");
            break;
        case 'K':
            printf("K harfi ile baslayanlar=\nKasım");
            break;
        case 'H':
            printf("H harfi ile baslayanlar=\nHaziran");
            break;
        case 'T':
            printf("T harfi ile baslayanlar=\nTemmuz");
            break;
        default:
            printf("Hatali bir harf tuslamasi");
    }
    return 0;}

```

2.e)Do-While Döngüsü:[Belgenin Başı](#)

While döngüsü gibi şart sağlanana kadar döngü devam ettirilir. While'dan farkı döngü sorgulama işlemi yaptırılmadan önce 1 kez döner ardından sorgulama işlemi yapılır istenen sağlanırsa döngü tekrar çalıştırılır. Döngünün içinde sayacımızı kendimiz artırmalıyız. Sayacımızı döngüden önce tanımlamalıyız döngünün içinde tanımlarsak eğer her dönmesinde sayac tanımladığımız değeri alır.

Kullanımı

```
do{  
komut yapıları;  
}while(ifade);
```

2_5_ornek.c Klavyeden alınan 2 sayı ile seçilen 4 işlemten birini yapan ve h harfine basıldığında döngüyü sonlandıran program yazınız.

```
#include <stdio.h>  
int main()  
{  
char harf;  
int toplama,cikarma,carpma,sayi1,sayi2;  
float bolme;  
printf("Islem yapmak istediginiz sayilari giriniz=");  
scanf("%d %d",&sayi1,&sayi2);  
do{  
printf("\nHangi islemi yapmak istersiniz\nToplama='+' \nCikarma='-'  
\nBolme='/' \nCarpma='*' \nCikmak icin='h' basiniz");  
scanf(" %c",&harf);  
if(harf=='+'){  
toplama=sayi1+sayi2;  
printf("\nSayilarin toplami=%d",toplama);  
}  
else if(harf=='-'){  
cikarma=sayi1-sayi2;  
printf("\nSayilarin toplami=%d",cikarma);  
}  
else if(harf=='/'){  
if(sayi1>0 && sayi2) //sayilarmiz 0 dan büyükse bolme islemi yapilir  
{  
bolme=(float)sayi1/(float)sayi2; //aldigimiz sayilari float ifadeye yani ondalığa cevirdik çünkü  
bölme işlemi sonucumuz ondalikli cikabilir  
printf("\nSayilarin bolumu=%f",bolme);  
}  
else{  
printf("\nSayilarin bolumu=0"); //eger sayilardan herhangi biri 0 sa 0 sonucu döndürülür  
}  
}  
else if(harf=='*'){  
carpma=sayi1*sayi2;  
printf("\nSayilarin toplami=%d",carpma);  
}  
}while(harf!='h'); //burada h ye basılmadığı sürece döngü devam etmesi sağlanır not kapısı ile  
printf("Cikis yapildi");  
return 0; }
```

3.Ön İşlemci Komutları (Makrolar)

Preprocessor [Belgenin Başı](#)

Sembolik sabitlerde olduğu gibi makrolar da kaynak kodun herhangi bir yerinde tanımlanabilir. Tanımladıkları yerden dosya sonuna kadar olan bölge içerisinde etkili olurlar.

3.a) #define: Bir deger tanımlamak için kullanılır.

#define pi 3.14 // gibi

3.b) #include: Bu işlemden sonra dosya ismi,kütüphane,daha önceden yazılmış bir C programının ismi yazılabilir.

#include <stdio.h> //standart giriş çıkış fonksiyonlarının bulunduğu kütüphaneyi çağırdık.

3.c) #if: If komutu ile aynı mantıkla çalışır. Verilen şart sağlanıyorsa komutun içerdiği işlem gerçekleşir. { } kullanılmadığı için #endif komutu ile bitirilir.

3.D) #ELSE: If şartı sağlanmazsa kullanılır,hiçbiri değilse anlamı vardır en sonuna #endif konur.

3.e) #elif:If şartı sağlanmazsa kullanılır.

3_1_ornek.c)Saati 13 verilen zaman diliminin ogleden önce mi sonra mı olduğunu gosteren ön işlemci komutunu yazınız.

```
#include <stdio.h>
#define hour 13
int main()
{
    #if hour>12
        printf("Saat ogleden sonradir");
    #elif hour<12
        printf("Saat ogleden oncedir");
    #else
        printf("Saat hatalidir");
    #endif
    return 0;
}
```

3.f) #ifdef: Daha önceden tanımlanmış olup olmadığını kontrol eder.Özellikle derleyicinin kendisinin tanımlandığı makrolar olup olmadığını test etmek için kullanılır.

#define komutu ile birşeyler tanımlanmış mı onu sorgular.

3.g) #error: Bu ön işlemci komutu çalıştırılacak olursa derleme durur ve belirtilen bir hata mesajı görüntülenir.

3.h) #undef: Önceden yapılan bir makro tanımlama işlemini iptal eder.

3_2_ornek.c) Tanimli sicaklik degerinin olup olmadigini gösterip sonra iptal eden ön işlemci komutunu yaziniz.

```
#include <stdio.h>

#define sicaklik 30 //önce tanimliyoruz

#undef sicaklik //iptal edicez

int main(void){
    #if defined MAX
    #error sicaklik tanimli
    printf("Sicaklik tanimli\n");
    #else
    printf("Sicaklik tanimli degil\n");
    #endif
    #if !defined sicaklik2
    printf("Sicaklik 2 tanimli degil\n");
    #else
    printf("Sicaklik 2 tanimli\n");
    #endif
    return 0;
}
```

3.j) #line: C'de koddaki satır bilgisi `__LINE__` makrosunda tutulur. #line ön işlemci komutu ise bu bilgileri değiştirmeye yarar.

3_3_ornek.c)

```
#include <stdio.h>

int main(void){
    printf("%s\n", __DATE__); //gün ay yıl bilgisi tutar
    printf("%s\n", __TIME__); //saat dakika saniye bilgisi tutar
    printf("%d\n", __STDC__); //C standartlarının sağlanıp sağlanmadığı kontrol edilir sağlanırsa =1
    sağlanmazsa=0
    return 0;
}
```

[Belgenin Başı](#)

4.Diziler:

Bellekte sürekli bir biçimde bulunan aynı türden nesnelerden oluşan kümeye dizi denir.

Kullanımı:

Değişken türü diziismi[boyut]; // diye tanımlanır boyut sayısal bir ifadedir tek boyutlu bir dizi yaratmak için kullanılır.

- Dizinin içerisinde bulunan elemanlar 0.indisten başlayarak bulunur.

int dizi[0];

Dizinin elemanlarının bellekte yer tutması;

.....	FC04
dizi[0]	FC05
dizi[1]	FC06
dizi[2]	FC07
dizi[3]	FC08
dizi[4]	FC09
.	FC0F

[Belgenin Başı](#)

4_1_ornek.c) Tek boyutlu tanımlı bir dizi ile klavyeden girilen diziyi karşılaştırıp bulduğunda bulunan indisi gösterip 10 kere eleman bulma denemesi yapan programı yazınız.

```
#include <stdio.h>
#define n 10
int main(){
    int i,j;
    int buldun=0; //her eleman bulunduğunda buldun değeri +1 artar
    int tanimlidizi[n]={1,5,21,34,123,64,675,23,34,85}; //tanımlı olan dizimiz
    int girilendizi[n]; //kullanıcıdan alacağımız dizimiz
    for(i=0;i<10;i++){ //dizinin boyutu 10 olduğu için kullanıcı 10 a kadar eleman girişi alıyoruz
        printf("%d.Dizinin elemanini giriniz=",i+1); //Diziler 0 indisten başladığı için ekrana dizinin
        1.elemanını yazdırmak için i+1 yapıyoruz
        scanf("%d",&girilendizi[i]);
        for(j=0;j<10;j++) //tanimli dizinin her elemanı döndürülür
        {
            if(tanimlidizi[j]==girilendizi[i]) //tanimlidizinin içinde kullanıcının girdiği deger var mı kontrol
            edilir eger varsa
            {
```

```

        printf("Tebrikler dizinin elemanini buldunuz\n bulunduğunuz indis=%d\n",j); //bulunan
dizinin elemanı tanımladığı de kaçinci indiste olduğu gösterilir
        buldun+=1; //bulunduğunda artar
        break;
    }
}
}
printf("%d.tane eleman buldun",buldun); //kaç tane eleman bulunduğu yazılır
return 0;
}

```

4.a)Çift Boyutlu Diziler:[Belgenin Başı](#)

X ve y düzleminde yani satır ve sütunlar aracılığıyla dizi oluşturulur.

Tanımlanması:

Veritürü diziismi[satirboyutu][sütunboyutu];

	Sütun 0	Sütun 1	Sütun 2
Satır 0	Diziismi[0][0]	Diziismi[0][1]	Diziismi[0][2]
Satır 1	Diziismi[1][0]	Diziismi[1][1]	Diziismi[1][2]
Satır 2	Diziismi[2][0]	Diziismi[2][1]	Diziismi[2][2]

Şeklinde bir 2 boyutlu dizi tanımlanmıştır.Satırda ya da sutunda indisi bulmak için [satirindisi][sutunindisi] gösterilir.

4_2_ornek.c)Tofas,Citroen,Pejo araba modellerinin aylık satış miktarı sırasıyla 123,423,234 dür.Bu firmaların satış miktarı ve markalarını ekrana yazdıran program yazınız.

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

```
int main()
```

```
{
```

```
    int i,j;
```

```
    char arabalar[3][10]={"Tofas","Citroen","Pejo"}; //her araba modeli 1 satırda tutulur 10 sütunda yer
```

```
    alır
```

```
    int satis[3]={123,423,234}; //arabaların satış miktarı 3 satırda tutulur çünkü 3 adet araba vardır
```

```
    for(i=0;i<3;i++)
```

```
    {
```

```
        printf("%s satış miktarı=%d\n",arabalar[i],satis[i]);
```

```
    }
```

```
    return 0;}
```

[Belgenin Başı](#)

4.b)Çok boyutlu diziler:[Belgenin Başı](#)

Çok boyutlu diziler x,y,z koordinatları içinde oluşturulur.Genellikle 3 for döngüsü kullanılır.

Tanimi:

Veritürü diziismi[boyut1][boyut2][boyut3];

4_3_ornek.c)3 okulda bulunan her 3 öğrencinin 2 notunu klavyeden alan ve sonra ekrana yazdıran programı yazınız.

```
#include <stdio.h>
int main()
{
    int i,j,a;
    int ogrenci[3][9][18];
    for(i=0;i<3;i++) //üç okulu hitap eder
    {
        for(j=0;j<3;j++) //her okuldaki 3 öğrencidir
        {
            for(a=0;a<2;a++) //her okuldaki 3 öğrencinin 2 tane sınav notudur
            {
                printf("%d.Okuldaki %d.Ogrencinin=%d notu giriniz=\n",i+1,j+1,a+1);
                scanf("%d",&ogrenci[i][j][a]); // her bir dizinin içerisi dolduktan sonra
                diğetine geçer
            }
        }
    }
    for(i=0;i<3;i++)
    {
        for(j=0;j<3;j++)
        {
            for(a=0;a<2;a++)
            {
                printf("%d.Okuldaki %d.Ogrencinin=%d notu
                giriniz=%d\n",i+1,j+1,a+1,ogrenci[i][j][a]);
            }
        }
    }
    return 0;
}
```

4.C)KARAKTER DİZİLERİ:[Belgenin Başı](#)

C’de yazı ve mesaj gibi alfabetik bilgiler içerisinde tutulur. Karakter dizileri char türü ve değişken isimlendirme kurallarına uyarak yapılır.Karakter dizilerinde araya boşluk koyarak değer göndermek istersek gets(degiskenadi); //olarak boşluk olmadan karater dizisini alır her boşluk karakter dizisinde 1 yer kaplar.

Kullanımı:

```
char s[6]={‘D’,’o’,’g’,’a’,’n’,’\0’}; //entera basıldığı için sonuna boşluk değeride eklenir
char calisanadlari[4][10]={“Dogan”,“Dorukan”,“Duran”,“Durmuş”};
```

4_4_ornek.c)Bir fabrikada çalışan işçilerin adını ve soyadını kullanıcıdan alan ve ekrana yazdıran program yazınız.

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

```
int main()
{
    char adsoyad[4][10];
    int i;
    for(i=0;i<4;i++)
    {
        printf("Isminizi ve soyisminizi giriniz=");
        //scanf("%s",&adsoyad[i]);
        gets(adsoyad[i]); //bosluktan sonrasınıda almasını istediğimiz için gets ile alıyoruz
    }

    for(i=0;i<4;i++)
    {
        printf("\n%d.Iscinin ismi ve soyismi=",i+1);
        puts(adsoyad[i]); //puts ile aldığımız degerleri ekrana yazdırıyoruz
    }
    return 0;
}
```

Belgenin Başı

5. Fonksiyonlar (Functions)

Her fonksiyon faydalı birtakım işlemleri yerine getirmek için tasarlanır ve çağrılır. Böylece tanımlanan fonksiyon gerekli oldukça çağrılır ve çağıran bloktaki kod tekrarı önlenmiş olur. Tanımlanan fonksiyon sadece bir kez çağrılacak da olsa bütünden ayrıldığı için program üzerinde kontrol artar. Ayrıca büyük bir programın tamamını tek bir main bloğu içine yerleştirmek iyi değildir. C dili bir çok hazır fonksiyon içerir. Bunlardan daha önce kullandığımız printf scanf de standart giriş çıkış kütüphanesi olan <stdio.h> 'ın fonksiyonlarıdır.

5.1) Fonksiyon Tanımı:

```
tip fonksiyon_ismi(parametre bildirimleri) //fonksiyon başlığı
{
    program komutları; //fonksiyon bloğu
}
```

Tip: Fonksiyonun döndüreceği değerin tiğini belirtir. Eğer hiç bir tip belirleyici bulunmuyorsa fonksiyonun int tipinde değerler döndüreceği kabul edilir. Hiç bir değer döndürmeyen fonksiyonlar ise void tipi ile tanımlanır.

```
void isim(void) //hiçbir değer döndürmeyen fonksiyon tanımlama
{
    ...
}
```

Fonksiyon Çağırma işlevi:

```
funkismi(argüman listesi); //fonksiyona gönderilen veriler argüman listesidir
```

Çağırılan fonksiyona değer döndürmek için return deyimi kullanılır.

5.2) Fonksiyon Tipleri:

- İteratif fonksiyonlar (yinelemeli fonksiyonlar)
- Rekürsif fonksiyonlar (özyinelemeli fonksiyonlar)

Parametreler call-by-value (değer gönderimli) veya call-by-reference (adres[pointer] gönderimli) olabilirler.

[Belgenin Başı](#)

Call-by-value ve call-by-reference aynı fonksiyonun parametreleri içinde olabilirler. Örn:
void sum(int* ptr, int a) şeklindeki fonksiyon tanımlaması doğrudur.

5.3)Fonksiyon Tanımlama Kuralları:

- Fonksiyonun iki parametresinin adı aynı olamaz.
- Dönüş tipi belirtilmişse mutlaka return ifadesi ile dönüş yapılmalıdır:
- If-else-else if gibi seçimli yapılar içinde dönüş yapılıyorsa her bir blokta dönüş belirtilmelidir.
- Fonksiyon adı ve parametrelerinde Türkçe karakter kullanılamaz.

5.4)İtaratif Fonksiyonlar (Yinelemeli fonksiyonlar):

Döngü, switch-case ve if-else-else if gibi standart yapıların dışında herhangi bir şekilde kendisini çağırmayan fonksiyonlardır.

C dilindeki fonksiyonların çoğu iteratiftir.

5.5)Rekürsif Fonksiyonlar(Özyinelemeli Fonksiyonlar):

- Kendi kendini çağırır.
- Fonksiyonda en az bir tane fonksiyonun kendisini çağırdığı satır olmalıdır.
- Fonksiyon her kendini çağırdığında problem belli oranda küçülmelidir.

COVID-19 / NEZLE / GRİP / ALERJİ SEMPTOMLARI								
	COVID-19		SOĞUK ALGINLIĞI NEZLE		GRİP		ALERJİ	
Ateş	✓	Sık	✓	Nadiren	✓	Sık	-	-
Yorgunluk	✓	Sık	✓	Bazen	✓	Sık	✓	Bazen
Kuru Öksürük	✓	Sık	-	-	✓	Sık	✓	Bazen
Solunum Zorluğu	✓	Şiddetli	-	-	-	-	✓	Bazen
Öksürük	✓	Sık	-	-	✓	Bazen	✓	Bazen
Ağrı	✓	Bazen	✓	Nadiren	✓	Sık	-	-
Hapşırma	-	-	✓	Sık	✓	Bazen	✓	Sık
Burun Akıntısı	✓	Bazen	-	-	✓	Bazen	✓	Sık
Burun Tıkanıklığı	-	-	✓	Sık	✓	Bazen	✓	Sık
Gözlerde Sulanma	-	-	-	-	-	-	✓	Sık
Boğaz Ağrısı	✓	Bazen	✓	Sık	✓	Bazen	-	-
İshal	✓	Bazen	-	-	✓	Nadiren	-	-

5_1_ornek.c) Kullanıcıdan hastalık belirtileri secimi yapması istenecek ve corona virüsü olup olmadığına karar veren programı yazınız hastalık belirtileri yukarıdaki tablodadır.

```
#include <conio.h>
#include <stdio.h>
int derecelendirme(int secenek,int derecem); //
int main()
{
    printf("Corona tespiti programımıza hosgeldiniz\nGerekli kriterleri doldurun test icin\n");
    int secenek,toplam=0,toplamim=0,derecem=0;
    do{
        printf("\nAtes=1\t,Yorgunluk=2\t,Kuru oksuruk=3\t,Solunum
zorlugu=4\t,Oksuruk=5\t,Agri=6\t,Hapsirma=7\t,Burun akintisi=8\n,Burun tikanikligi=9\t,Gozlerde
sulanma=10\t,Bogaz agrisi=11\t,İshal=12\t,Cikmak icin=0 basiniz\n");
```

```

scanf("%u",&secenek); //unsigned sayi girisidir negatif sayi giremez
switch(secenek) //secilen belirtiye göre işlem yapan parametre
{
case 1: //eger bir se ates derecelendirmesi yapicak
    toplamim+=derecelendirme(secenek,derecem); //derecelendirme fonksiyonundan döndürülen değeri
toplamim degiskene ekliyoruz
    break;
case 2: //eger 2 ise yorgunluk gibi
    toplamim+=derecelendirme(secenek,derecem);
    break;
case 3:
    toplamim+=derecelendirme(secenek,derecem);
    break;
case 4:
    toplamim+=derecelendirme(secenek,derecem);
    break;
case 5:
    toplamim+=derecelendirme(secenek,derecem);
    break;
case 6:
    toplamim+=derecelendirme(secenek,derecem);
    break;
case 7:
    toplamim+=derecelendirme(secenek,derecem);
    break;
case 8:
    toplamim+=derecelendirme(secenek,derecem);
    break;
case 9:
    toplamim+=derecelendirme(secenek,derecem);
    break;
case 10:
    toplamim+=derecelendirme(secenek,derecem);
    break;
case 11:
    toplamim+=derecelendirme(secenek,derecem);
    break;
case 12:
    toplamim+=derecelendirme(secenek,derecem);
    break;
default:
    printf("Hatali bir tuslama eger cikmayi istiyorsaniz 0 a basiniz=");
    break;
}
}while(secenek!=0); //secenek 0 a eşit olmadığı zaman bu döngü dönsün seçenek 0 ise döngü biter
if(toplamim>10 && toplamim<=20){ //toplamim degiskeni >10 büyük ve <=20 se
    printf("\nSoguk alginligi ve nezle olabilirsiniz=%d",toplamim); //soğuk alginligi belirtileri vardır
}
else if(toplamim>21 && toplamim<=30){ //toplamim değışkeni >21 den büyük ve <=30 se
    printf("\nAlerji olabilirsiniz=%d",toplamim); //alerji belirtileri olabilir
}
else if(toplamim>31 && toplamim<=37){ //toplamim değışkeni>31 den büyük ve <=37 den küçükse
    printf("\nGrip olabilirsiniz=%d",toplamim); //grip belirtileri
}
else if(toplamim>=38 && toplamim<50){ //toplamim degiskeni >= 38 eşit ve toplamim <50 küçükse
    printf("\nCorona olma ihtimaliniz vardır devlet hastanesine gidiniz=%d",toplamim); //corona olma
belirtisidir

```

```

    }
    else{
        printf("Hicbirseyiniz yoktur yalan soylemeyiniz=%d",toplamim); //eger toplamim deęişkeni yukarıdaki
        şartlardan hiç biri işe hiçbirşeyiniz yoktur.
    }
    system("pause"); //sistemin durup beklemesi sağlanır sonuçları ekrana yazdırdıktan sonra
}
int derecelendirme(int secenek,int derecem)
{
    printf("Derecelendirmenizi yapiniz=Nadiren=2\tBazen=3\tDaima=4\t,Siddetli olarak=5\n"); //secilen
    hastalığın üzerinizdeki belirtisi kontrolü
    scanf("%u",&derecem);
    return derecem; //klavyeden alınan derecelendirmeyi döndürdüğümüz deęişken
}

```

[Belgenin Başı](#)

5_2_ornek.c Klavyeden alınan 3 sayıyı büyükten küçüğe sıralayan programı yazınız
fonksiyon kullanarak.

```

#include <stdio.h>
int sayibulma(int sayi1,int sayi2,int sayi3); //programın başında tanımladık çünkü daha esnek duruyor ve
kodlardan önce nasıl bir işlem yapılacağı belli oluyor
int main()
{
    int sayi1,sayi2,sayi3;
    printf("3 tane sayi giriniz=");
    scanf("%d %d %d",&sayi1,&sayi2,&sayi3);
    sayibulma(sayi1,sayi2,sayi3); //sayibulma fonksiyonunu içine gönderdiğimiz sayılarla çağırıyoruz
    return 0;
}
int sayibulma(int sayi1,int sayi2,int sayi3) //sayılar
{
    if(sayi1>sayi2 && sayi1>sayi3) //sayi1 in diğer sayılardan büyük olup olmadığını karşılaştırdık
    {
        if(sayi2>sayi3) //eger sayi1 büyükse sayi2 mi yoksa sayi3 mü büyük onu kontrol ettik
        {
            printf("%d>%d>%d",sayi1,sayi2,sayi3);
        }
        else
        {
            printf("%d>%d>%d",sayi1,sayi3,sayi2);
        }
    }
    else if(sayi2>sayi1 && sayi2>sayi3) //sayi2 diğer sayılardan büyük mü kontrol ettik
    {
        if(sayi1>sayi3) //eger büyükse sayi1 mi sayi3 mü daha büyük bunu kontrol ettik
        {
            printf("%d>%d>%d",sayi2,sayi1,sayi3);
        }
        else
        {
            printf("%d>%d>%d",sayi2,sayi3,sayi1);
        }
    }
    else //hiçbiri değilse sayi3 büyüktür
    {
        if(sayi2>sayi1) //sayi3 büyükse,sayi2 mi sayi1 mi büyük onu kontrol ettirdik
        {

```



```

printf("%d>%d>%d",sayi3,sayi1,sayi2);
}
else
{
printf("%d>%d>%d",sayi3,sayi1,sayi2);
}
}

```

[}Belgenin Başı](#)

5.4)Dizilerin Fonksiyonlarda Kullanımı:

Fonksiyon içerisindeki işlemlerden gönderilen parametrenin kendisi etkilenmez.Ama diziler böyle değildir.Normal kullanımda dizi fonksiyona gönderildiğinde elemanlar değiştirilebilir (referans).Dizinin sadece herhangi bir elemanı gönderildiğinde ise değeri değiştirilemez.Eğer dizi elemanlarını teker teker gönderiyorsak onların adreslerini pointer aracılığıyla almalıyız.

Kullanımı:

tip fonksiyon_ismi(int sayılar[],int n)

```

{
Komut blogu;
}

```

tip fonksiyon_ismi2(char [10],int n)

```

{
Komut blogu;
}

```

Main altında dizili fonksiyon çağırma işlevi:

fonksiyon_ismi(degiskenadi[],boyut);

fonsiyon_ismi2(karakterdizisi[10],boyut); //burada çift boyutlu karakter dizi gönderdik
içindeki 10,karakter uzunluklu boyutu kadar bir char dizisi string ifade döndürecek

5_3_ornek.c)Bir iş yerinde çalışan 4 işçinin maaşını ve adını kullanıcıdan alan fonksiyonu ve programı yazınız.

```

#include <stdio.h>
void iscilistesi(char isciler[10],int maas[],int n);
int main()
{
char isciler[4][10];
int maas[4];
int i;
iscilistesi(isciler[10],maas,4); //iscilerin karakter uzunluğunu ve kaç tane işçi girileceğini kaç tane maas
girileceği gönderdik 4 ile
}
void iscilistesi(char isciler[10],int maas[],int n)
{
int i=0;
for(i=0;i<n;i++)
{

```

```
printf("%d.iscinin adini ve maasini giriniz=",i+1);
scanf("%s %d",&isciler[i],&maas[i]);
```

```
}
```

[\)Belgenin Başı](#)

5_4_ornek.c) Bir değerin birden fazla tekrar edilmesi olayına frekans denir. Bir zar atma deneyinde kaç kez zar atılacağını kullanıcıdan alan ve zarları atan (rastgele sayı atama) ve her bir yüzün kaç gez geldiğini bulan programı yazınız.

```
#include <stdio.h>
#include <stdlib.h> //rand fonksiyonunu çağırmak için kullandık rand fonksiyonu rastgele sayı atmaya yarar
#include <time.h> //zamana göre yani programın çalışmasına göre yeni sayılar gönderir
int zaratma(int deney); //zar atma deneyini yapmasını sağlayan deney
int main()
{
    int yuz;
    int deney;
    int i;
    printf("Kac kere zar atma deneyi yapmak istersiniz\n"); //zar atma deneyinin kaç kere olacağını
    scanf("%d",&deney);
    zaratma(deney); //zaratma fonksiyonuna deneyi gönderdik çünkü deney kadar zar atma işlemi yapılmalı
}
int zaratma(int deney)
{
    {
        srand(time(NULL)); //her program çalıştığında rastgele sayı değerleri değişsin
        int zarim;
        int i=0;
        int toplam1=0,toplam2=0,toplam3=0,toplam4=0,toplam5=0,toplam6=0;
        for(i=0;i<deney;i++){
            zarim=rand()%6+1; //1 ile 6 arasında bir sayı atar zarim değişkenine
            printf("\nZar=%d",zarim);
            switch(zarim)
            {
                case 1: //rastgele atanan sayı 1 se
                    toplam1++; //toplam1 artar
                    break;
                case 2: //rastgele atanan sayı 2 ise
                    toplam2++; //toplam2 artar
                    break;
                case 3: //rastgele atanan sayı 3 ise
                    toplam3++; //toplam3 artar
                    break;
                case 4: //rastgele atanan sayı 4 ise
                    toplam4++; //toplam4 artar
                    break;
                case 5: //rastgele atanan sayı 5 ise
                    toplam5++; //toplam5 artar
                    break;
                case 6: //rastgele atanan sayı 6 ise
                    toplam6++; //toplam6 artar
                    break;
            }
        }
        printf("\n1 gelme frekansi=%d,\n2 gelme frekansi=%d,\n3 gelme frekansi=%d,\n4 gelme frekansi=%d,\n5 gelme frekansi %d,\n6 gelme frekansi=%d",toplam1,toplam2,toplam3,toplam4,toplam5,toplam6);
    } }
}
```

5_5_ornek.c) Tanimli kullanıcı adı,şifre,bakiye ile para çekme,para yatırma,bakiye gösterme gibi işlemleri yapabilen ardından belirli bir tuşa bastığında bankamatığı sonlandırıp kartı iade eden ve şifre 3 kere hatalı girdiğinde kartı bloke eden programı yazınız.

```
#include <stdio.h>
#include <string.h> //bu kütüphaneyi strcmp fonksiyonu için kullandım çünkü string ifadeleri karşılaştırma
yapmam gerekiyordu kullanıcıadı ve klavyeden girilen kullanıcı adı
int bankaislemleri(int bakiye);
int main()
{
    int hak=0;
    char kullaniciid[15]="dogan"; //tanimli kullanıcı
    int password=12345;
    char mykullaniciid[15];
    int mypassword;
    int bakiye=2000;
    while(1){
        printf("Kullanıcı adini giriniz=");
        scanf("%s",&mykullaniciid); //klavyeden girilen değerler
        printf("Kullanıcı şifreyi giriniz=");
        scanf("%d",&mypassword);
        if(strcmp(kullaniciid,mykullaniciid)==0 && mypassword==password) //klavyeden girilen kullanıcı adı ve
        tanımlı kullanıcı adı eşitse ve şifre klavyeden girilen şifreye eşitse
        {
            printf("Bankamiza hosgeldiniz\n"); //giriş işlemi başarılı olacal
            bankaislemleri(bakiye); //tanimli bakiye bilgisi gönderilecek ve bankaislemleri fonksiyonu
            çalıştırılacak
            break; //döngüyü bitirdik fonksiyona gittikten sonra;
        }
        else if(strcmp(kullaniciid,mykullaniciid)==0 || mypassword==password)
        {
            printf("Kullanıcı adı ya da password hatalı\n");
        }
        else
        {
            printf("Hatalı giriş");
            hak++;
            if(hak>=3)
            {
                printf("\nKartınız bloke olmuştur");
                break;
            }
            else
            {
                printf("\nLutfen tekrar deneyiniz kalan hak=%d\n",3-hak);
            }
        }
    }
}
}
int bankaislemleri(int bakiye)
{
    char secenek;
    do{
        printf("Para yatırma icin='+' \nPara cekme icin='-'\n,Bakiye gormek icin='*' \nCikmak icin='q' basiniz\n");
        //hangi işlemin yapılması gerektiği sorulacak
        scanf(" %c",&secenek);
```

```

int paramiktari;
switch (secenek){ //secilen işleme göre belirli parametreye gidilecek
    case '+':
        printf("Ne kadar para girmek istersiniz=");
        scanf("%d",&paramiktari);
        bakiye=paramiktari+bakiye;
        printf("\nYeni bakiyeniz=%d\nBaska bir islem yapmak istermisiniz herhangi bir tusa
basiniz ",bakiye);

        scanf(" %c",&secenek);
        break;
    case '-':
        printf("\nNe kadar para cekmek istersiniz="); //para çekme işleminde paramiktari
        scanf("%d",&paramiktari);
        if(paramiktari>bakiye){ //eğer para miktarı > bakiye den büyükse
            printf("Paranız yetmiyor"); //para yetmicek işlem gerçekleştirilemeyecek
            printf("Bakiyeniz=%d\nBaska bir islem yapmak istermisiniz herhangi bir tusa
basiniz",bakiye);

            scanf(" %c",&secenek); //başka bir işleme gitmek için
        }
        else{
            bakiye=bakiye-paramiktari;
            printf("Yeni bakiyeniz=%d\nBaska bir islem yapmak istermisiniz herhangi bir tusa
basiniz",bakiye);

            scanf(" %c",&secenek);
        }
        break;
    case '*':
        printf("Bakiyeniz=%d",bakiye);
        printf("\nBaska bir islem yapmak istermisiniz herhangi bir tusa basiniz ",bakiye);
        scanf(" %c",&secenek);
        break;
    case 'q':
        printf("Cikis islemi secilmistir");
        break;
    default:
        printf("Hatali secim tekrar deneyiniz");
        printf("Baska bir islem yapmak istermisiniz herhangi bir tusa basiniz");
        scanf(" %c",&secenek);
        break;
}
}while(secenek!='q'); //kullanıcı q ya başana kadar döngü dönecek
printf("\nKartınız iade ediliyor"); //q ye baştığında döngü bitecek ve kart iade edilecek

```

[}Belgenin Başı](#)

5_6_ornek.c) Diziler kullanılarak kullanıcının girdiği 3 sayının yine kullanıcının girdiği 3 diğer sayıyla 4 işlemi yapan programı yazınız.

```

#include <stdio.h>
#include <stdlib.h>
#define n 3
int toplama(int a[],int b[],int boyut);
int cikarma(int a[],int b[],int boyut);
float bolme(int a[],int b[],int boyut);
int carpma(int a[],int b[],int boyut);
int main()
{
    char secenek;
    int a[n];
    int b[n];

```

```

int i;
for(i=0;i<3;i++){
    printf("a[%d] degerini giriniz=",i+1); //klavyeden dizinin her bir elemanini aldık
    scanf("%d",&a[i]);
}
for(i=0;i<3;i++){
    printf("b[%d] degerini giriniz=",i+1); //klavyeden diğer dizisinin her bir elemanini aldık
    scanf("%d",&b[i]);
}
do{
    printf("Hangi Ė • slemleri yapmak istersiniz=\nToplama=+\nCikarma=-\nBolme=/\nCarpma=*\nCikmak
icin=q\n"); //dizi üzerinde hangi işlemi yapmak istediğimize karar veriyoruz
    scanf(" %c",&secenek);
    switch(secenek)
    {
        case '+':
            printf("Toplama islemi secildi");
            toplama(a,b,n); //toplama fonksiyonumuzu a,b dizilerimizi ve boyutunu gönderdik n=boyut
            break;
        case '-':
            printf("Cikarma islemi secildi\n");
            cikarma(a,b,n); //cikarma fonksiyonumuzu a,b dizilerimizi ve boyutunu gönderdik n=boyut
            break;
        case '*':
            printf("Carpma islemi secildi\n");
            carpma(a,b,n); //carpma fonksiyonumuzu a,b dizilerimizi ve boyutunu gönderdik n=boyut
            break;
        case '/':
            printf("Bolme islemi secildi=\n");
            bolme(a,b,n); //bolme fonksiyonumuzu a,b dizilerimizi ve boyutunu gönderdik n=boyut
            break;
        case 'q':
            printf("Cikma islemi secilmistir\n");
            break;
        default:
            printf("Hatali islem secildi tekrar deneyiniz\n"); //hatali bir tusa başıldığında q dışında döngü dönmeye
            devam edecek
            break;
    }
}while(secenek!='q'); //kullanici klavyeden q ye başana kadar döngü dönsün
printf("Hesap makinesi kapat\½lacak");
}
int toplama(int a[],int b[],int boyut) //dizilerimizi aldık ve dizimiz üzerinde toplama işlemi yaptırıyoruz
{
    int i;
    int c[boyut]; //yeni bir dizi oluşturduk ve toplama sonucunu onun üzerine ekledik
    for(i=0;i<3;i++)
    {
        c[i]=a[i]+b[i];
        printf("c[%d]=%d\n",i,c[i]);
    }
}
int cikarma(int a[],int b[],int boyut) //dizilerimizi aldık ve dizimiz üzerinde cikarma işlemi yaptırıyoruz
{
    int i;
    int c[boyut]; //yeni bir dizi oluşturduk ve cikarma sonucunu onun üzerine ekledik
    for(i=0;i<3;i++)

```

```

    {
        c[i]=a[i]-b[i];
        printf("c[%d]=%d\n",i,c[i]);
    }
}
int carpma(int a[],int b[],int boyut) //dizilerimizi aldık ve dizimiz üzerinde carpma işlemi yaptırıyoruz
{
    int i;
    int c[boyut]; //yeni bir dizi oluşturduk ve carpma sonucunu onun üzerine ekledik
    for(i=0;i<3;i++)
    {
        c[i]=a[i]*b[i];
        printf("c[%d]=%d\n",i,c[i]);
    }
}
float bolme(int a[],int b[],int boyut)
{
    int i;
    float c[boyut];
    for(i=0;i<3;i++)
    {
        c[i]=(float)a[i]/(float)b[i]; //float'a çevirdim çünkü bölme işlemleri ondalıklı sonuçlar üretebilir
        printf("c[%d]=%0.2f\n",i+1,c[i]);
    }
}

```

[Belgenin Başı](#)

5_7_ornek.c)Kullanıcıdan alınan sayıyı basamak sayısını bulup sonra sayıyı tersten yazıp ekranda gösteren fonksiyonun programını yazınız.

```

#include <stdio.h>
int basamagibul(int sayi);
int tersinecevir(int sayi);
int main() //main blogu altında sayi girişimizi aldık
{
    int sayi;
    printf("Sayıyı giriniz=");
    scanf("%d",&sayi);
    basamagibul(sayi); //sayinin kaç basamaklı olduğunu bulmak için basamagibul fonksiyonunu çağırdık
    tersinecevir(sayi);
}
int basamagibul(int sayi)
{
    int islem=0,i=0; //i=basamaksayisi,islem boş değişken
    if(sayi>0) //sayi 0 dan büyükse
    {
        do{
            islem=sayi%10; //sayiyinin 10 la modunu alıp islem değişkenine (boş bir değişkene)atadık
            sayi=sayi/10; //tekrar sayıyı 10 ile bölüp bölme işleminden çıkan sayıyı sayinin yeni değeri yaptık
            i++; //i yi artırdık yani basamak sayimizi
        }while(sayi>0); //sayi>0 dan büyük olduğu her durumda döngümüzü çalıştırdık
    }
    return i; //i değişkenimizi döndürdük
}
int tersinecevir(int sayi)
{
    int basamaksayisi=basamagibul(sayi); //rekürsif bir fonksiyon, fonksiyon kendi kendini çağırdı bu sayede
    basamak sayimiz bulunmuş oldu bunu basamaksayisinin içine koyduk
    printf("%d Basamaklı\n",basamaksayisi); //basamaksayimizi yazdırdık
    int i,islem; //i değişkeni tanımladık döngüde kullanmak için ve işlemler için işlem
    for(i=0;i<basamaksayisi;i++)

```

```
{
islem=sayi%10; //sayiyinin 10 ile modunu alıp islem değişkenine kalanı koyduk
    sayi=sayi/10; //sayiyi /10 bölüp yeni sayi değişkenimizi oluşturduk
    printf("%d",islem); //kalanı gösterdik
    if(sayi<10) //eger sayi 10 dan küçükse tekrardan kalanı bulunamayacağı için
        {printf("%d",sayi); //sayiyi direk olarak ekrana yazdırdık
break;    }
}
}
```

Belgenin Başı

6. Göstericiler (pointer) Belgenin Başı

Bir pointer ana bellekte bir adres belirtir. Bir pointer, başka bir değişkenin adresini gösteren bir değişkendir. Dolayısıyla, aynen öteki değişkenler için olduğu gibi, programda kullanılmadan önce bildirilmelidir. Bildirim için, pointer değişkenine istenilen bir ad verilebilir; ancak bu adın önüne * simgesi konulur. Bir değişken önüne * simgesi konulunca, derleyici onu bir pointer olarak görmeye başlar.

6.1) Pointer bildirimi:

```
int *ucret; //integer türünden yani sayısal bir veri yaratır ve bellekte yer kaplar
```

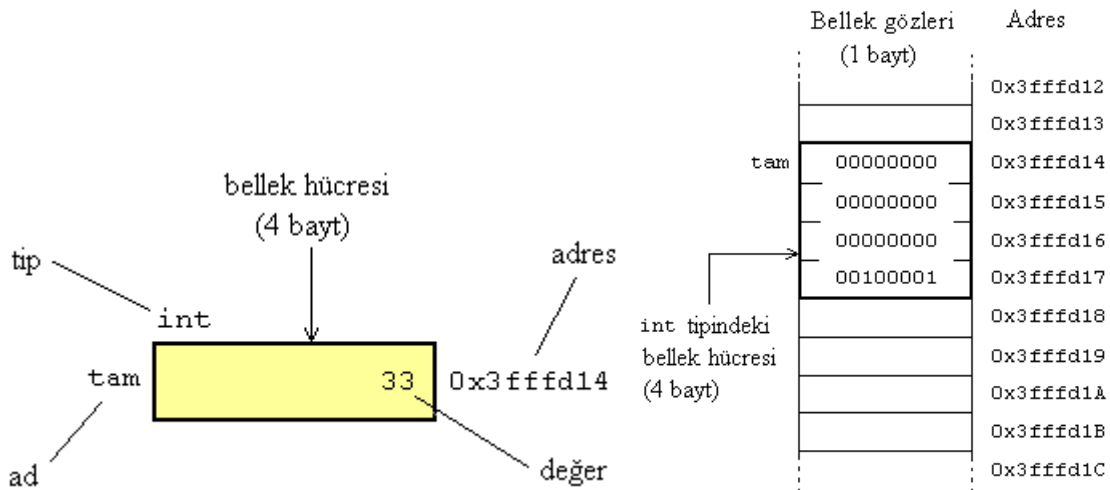
```
int x,y;
```

```
int *ptr;
```

Değişken adı	Adresi	Kapsamı
x	24634	0
y	24634	0
*ptr	24638	0

x=8; //diye değer ataması olduğunda x değişkeninin kapsamı 8 olacaktır.

Pointer değişkenler için iki türlü değer kullanılacaktır. Birincisi pointerın işaret ettiği adres, ikincisi ise işaret edilen adresteki değer. Program, bu iki değerden hangisini istiyorsa onu dikkate alır.



6.2) Adres operatörü: &

Adres operatörü pointerın adresini belli eder.

Kullanımı.


```
int y,*ptry;
```

```
y=8;
```

ptry=&y; //adres operatörüyle atanan ptry değişkeni y değişkenin bulunduğu adresi işaret etmesini sağlar; yani bu atamadan sonra ana bellekteki durum ptry nin kapsamı y değişkenin adresi olur.Yani ptry pointerı y değişkeninin bulunduğu adresi işaret eder işaret edilen adres 8 değerini içerir.

Ad	Adres	Kapsam
y	24634	8
*ptry	24636	24634

6.3) İçerik (indirection) Operatörü:*

Bu operatör, pointerin işaret ettiği adresteki değeri verir. Bir bakımdan, adres operatörü & nın karıştıdır; işaret edilen adresi değil, o adresteki değeri gösterir.

Kullanımı:

```
int a=10,ptr,a,b;
```

```
ptr=&a; //adresini verdik
```

```
b=*ptr //işaret ettiği değeri gönderdik sonuç olarak b değişkeni 10 değerini alır
```

Pointerler ile işlemler:

```
int *ptr=65; //65 değişkenin ptr değişkenimize atadık
```

```
int a=*ptr; //a değişkenine ptr değişkeninin işaret ettiği sayıyı atadık
```

```
int y=*ptr +5; // y değişkenine ptr nin işaret ettiği sayıyı atayıp 5 artırdık
```

```
*ptr++; //ptr nin adresteki yerinin artmasını sağlar bu bir integer değişkense 24634 olacağına 24636 olur
```

```
*(ptr++); //ptr nin içinde işaret ettiği ayıyı +1 artırır.
```

Not:Hafızada kaplanılan yeri onaltılık sayı sisteminde göstermek için %x kullanılır.Bu adresi onaltılık olarak ekranda gösterir.Tam adresin gösterimi için ise %p kullanılır.

6_1_ornek.c) Tanımlı olan 232,13.32 sayısını ve D harfini ekranda yazdırıp adresini gösteren sonra adres değerini +1 artıran programı yazınız.

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

```
int main()
```

```
{
```

```
    int sayi=232;
```

```
    char harf='D';
```

```
    float ondalikli=13.32;
```

```
    int *sayiptr;
```

```
    char *harfptr;
```

```
    float *ondalikptr;
```

```
    sayiptr=&sayi;
```

```

harfptr=&harf;
ondalikliptr=&ondalikli;
printf("Sayimiz =%d,\t adresi=%x\t%p\n",*sayiptr,sayiptr,sayiptr); //tanimli sayiyi ve onaltilik sayi
sisteminde adresini,tam adresinide gösterdik
printf("Harfimiz=%c,\t adresi=%x\t%p\n",*harfptr,harfptr,harfptr); //tanimli harf ,onaltılık sayi
sisteminde adresi ,tam adresini gösterdik
printf("Ondalikli sayimiz=%0.2f,\t adresi=%x\t%p\n",*ondalikliptr,ondalikliptr,ondalikliptr); // tanimli
ondalik sayi,onaltılık sayi sisteminde adres, tam adresini gösterdik
*sayiptr++; //pointerlarımızın adresini +1 artırdık
*harfptr++;
*ondalikliptr++;
printf("int adresi +1 artırdık=%x\t%p\n",sayiptr,sayiptr); // onaltılık sayi sisteminde artmış galini
gösterdik ve tam adresini bu sayede integerın 4 bytlık bir yer kapladığını gördük
printf("char adresini +1 artırdık=%x\t%p\n",harfptr,harfptr); //onaltılık sayi sisteminde artmış galini
gösterdik ve tam adresini bu sayede integerın 1 bytlık bir yer kapladığını gördük
printf("Float adresini +1 artırdık=%x\t%p\n",ondalikliptr,ondalikliptr); //onaltılık sayi sisteminde
artmış galini gösterdik ve tam adresini bu sayede integerın 8 bytlık bir yer kapladığını gördük
return 0;

```

[\)Belgenin Başı](#)

6_2_ornek.c)Girilen 2 sayiya göre 5 işlemi yaptıran hesap makinesini her işlem yapıldıktan sonra tekrar dönüş yapan ve sayıların bolme işleminde 0 dan büyük olma şartı olduktan sonra tekrar sayı girişini sağlayan programı yazınız.

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

```
int tekrarsayigirisi(int *sayi1,int *sayi2);
```

```
int main()
```

```
{
```

```
    char secenek;
```

```
    int sayi1,sayi2;
```

```
    printf("1.Sayiyi giriniz="); //sayi girişlerini aldık
```

```
    scanf("%d",&sayi1);
```

```
    printf("2.Sayiyi giriniz=");
```

```
    scanf("%d",&sayi2);
```

```
    int toplama,cikarma,carpma,mod;
```

```
    float bolme;
```

```
    do
```

```
    {
```

```
        printf("Hangi islemi yapmak istersiniz\nToplama:+,Cikarma:-,Carpma:*,Bolme:/,Mod
        almak=%,Cikmak icin:q\n"); //işlem seçimi yapılır
```

```
        scanf(" %c",&secenek);
```

```
        switch(secenek)
```

```
        {
```

```
            case '+': //toplama işlemi ise buraya girer
```

```
            toplama=sayi1+sayi2; //toplama işlemi yapılır
```

```
            printf("Toplama islemi sonucu=%d\n",toplama); //sonucu ekrana yazılır
```

```
            break;
```

```
            case '-': //cikarma işlemi için buraya girer
```

```
            cikarma=sayi1-sayi2; //cikarma işlemi yapılır
```

```
            printf("Cikarma islemi sonucu=%d\n ",cikarma); //sonucu yazdırılır
```

```
            break;
```

```
            case '*':
```

```
            carpma=sayi1*sayi2; //carpma işlemi yapılır
```

```
            printf("Carpma islemi sonucu=%d\n",carpma); //carpma işleminin sonucu
```

```
            break;
```

```
            case '/':
```

```
            if(sayi1>0 && sayi2>0) //bolme işlemi yapılması için sayıların 2 sininde 0 dan büyük olması gerekir
```

```
            {
```

```

        bolme=(float)sayi1/(float)sayi2; //doğruysa bolme işlemi yapılır
        printf("Bolme islemi sonucu=%.2f\n",bolme);
        break;
    }
    else
    {
printf("Sayi girisi paneline yonlendiriliyorsunuz"); //sayilarin 2 sinden biri 0 ya da 0 dan küçükse
tekrarsayigirisi(&sayi1,&sayi2); //tekrar sayi girisi fonksiyonuna sayilar gonderilir
bolme=(float)sayi1/(float)sayi2; //gonderilen sayilar ondalik cinsinden tekrar isleme tabi tutulur
printf("Bolme islemi sonucu=%.2f",bolme);
        break;
    }
    case '%':
if(sayi1>0 && sayi2>0) //mod alma isleminde bolme isleminde olduğu gibi sayilarin 0 dan büyük olma şarti
sorgulanir
    {
        mod=sayi1%sayi2; //sayilar 0 dan büyükse mod alma işlemi yapılır
        printf("Kalan sonucu=%d\n",mod);
        break;
    }
    else
    {
printf("Sayi girisi paneline yonlendiriliyorsunuz\n"); //sayilar 0 dan büyük değilse
tekrarsayigirisi(&sayi1,&sayi2); //sayilar tekrarsayigirisi fonksiyonuna gönderilir adresleriyle birlikte
        mod=sayi1%sayi2; //mod alma işlemi yapılır
        printf("Kalan sonucu=%d\n",mod); //kalan sonucu ekrana yazar
        break;
    }
    case 'q':
printf("Hesap makinesinden cikis islemi secildi\n"); //hesap makinesinden çıkış işlemi
secilmistir
        break;
    default:
printf("Yanlis bir tuslama dogru islemi seciniz:\n"); //tekrar tuslama işlemi döndürülür
    }
}while(secenek!='q'); //çıkış işlemi q tuşlanana kadar yani işlem devam eder
printf("Hesap makinesi kapatildi"); //q tuşlandığında hesap makinesi kapanır
}
int tekrarsayigirisi(int *sayi1,int *sayi2) //tekrar sayi girisi fonksiyonumuza gelen sayilari işaret ettik
{
    int sayilar1,sayilar2; //sayi girisi için değişkenler atayıp sayi girisini aldık
    printf("1.Sayi giriniz:");
    scanf("%d",&sayilar1);
    printf("2.Sayi giriniz:");
    scanf("%d",&sayilar2);
    *sayi1=sayilar1; //girilen sayilari işaretli değişkenlere atarız
    *sayi2=sayilar2;
}

```

6.4)Pointerlarda Diziler: [Belgenin Başı](#)

Pointerler üzerinde geçerli aritmetik yardımıyla dizilere pointer değişkenler ile erişmek mümkündür.

Kullanımı:

```
int mydizi[5];
```

```
int *ptrdizi;
```

ptrdizi=mydizi; //dizinin elemanına erişmek için ilk elemanın adresi kullanılabilir

Sonra for döngüsü ile pointer değişkeni artırma işlemi yapılabilir

ptrdizi++; şeklinde

6_3_ornek.c) 10 tane asal sayiyi sırasıyla ekrana yazdırıp ve bunların adreslerini sırasıyla gösteren programı yazınız.

```
#include <stdio.h>
int main()
{
    int mydizi[10]; //10 tane asal sayinin bulunacağı diziyi tanımladık
    int *ptrdizi; //pointer dizimizi tanımladık
    int i,j,a=0;
    int sayac=0; //sayac 0 dan başladı asal sayı kontrolü için oluşturduk
    ptrdizi=mydizi; //başlangıç değerini atadık dizimizin
    for(i=2;i<100;a<10;i++) //asal sayılar 2 den başladığı için sayimizi 2 den başlattık 10 tane dizi
    elemanımız olunca döngü bitecek
    {
        sayac=0; //her döngüde sayac 0'lanacak asal sayi için
        for(j=2;j<i;j++) // j=2 yi 2 den başlattık çünkü asal sayıların kuralı 1 e ve kendisine
        bölünemeyen sayılar i den küçük yaptık ve kendisine bölünemeyeceği anlamını verdik
        {
            if(i%j==0) //i%j== bölüp kalanı 0 a esitse
            {
                sayac++; //sayacı ++artırdık
                break; //döngüyü kırdık
            }
        }
        if(sayac==0) //sayac 0 a esitse
        {
            mydizi[a]=i; //i.sayiyi dizimize atadık
            a++; //dizimizin eleman sayisini artırdık
        }
    }
    for(i=0;i<10;i++)
    {
        //printf("%d.dizi\n",mydizi[i]);
        printf("\n%d.Asal sayi=%d,ve adresi=%p",i+1,*(ptrdizi+i),(ptrdizi+i)); //kacinci asal sayi
        olduğunu yazdık ve adresteki asal sayiyi işaret ettik ve adresini gösterdik sırasıyla
    }
}
```

6.5)Pointerların Fonksiyonlarla Kullanımı:[Belgenin Başı](#)

Bu çağırma gönderilecek olan verilerin değerleri değil, değerlerin bulunduğu bellek gözünün adresi gönderilir. Bu şekilde kullanımda, fonksiyon kendisine gelen gerçek değeri alıp kullanır ve istenirse sonucu doğrudan o adrese yazabilir.

Kullanımı:

```
int *ptr; //tanımlı pointerı
```

Çağırırken

```
void fonksiyon_ismi(int *t);
```

Fonksiyon ismi tanımlama aynıdır

fonksiyon_ismi(t);

fonksiyon_ismi(&t); //adres yollaması yapılır bellekte kapladığı yer

Pointerlarda işlemler bellek üzerinde gerçekleştirildiğinden değer değişimi tüm programa yansımaktadır.

Ama fonksiyon pointer olarak tanımlanırsa

int (*islem)(int, int) ifadesi ile yazılır.

6_4_ornek.c) Pointer ve fonksiyonlar kullanarak fibonacci serisinin klavyeden girilen adımını bulan programı yazınız.

```
#include <stdio.h>
#include <stdlib.h> //bu kütüphaneyi çağırmanın sebebi malloc fonksiyonunu getirip hafızada yer açmak
int fibonacci(int *terim,int n);
int main()
{
    int n;
    printf("Kac terimli olmasını istersiniz="); //kaç terimli olmasını istersek o terimi alıyoruz
    scanf("%d",&n);
    int terim[n]; //terim dizisini oluşturuyoruz
    terim[n]=(int*)malloc(n*sizeof(int)); //hafızada n*4 yani integer değer kadar yer açıyoruz
    terim[0]=1; //ilk terimimiz 1 dir
    terim[1]=1; //2. terimimiz yine 1 dir
    printf("%d.Fibonacci serisinin=%d adresi:\t %p",n,terim[n],terim); //fibonacci fonksiyonuna terim
    pointerının ve seçilen terimi gönderdik ve çağırdık sonra terimin adresini aldık
    free(terim); //hafızada kaplanan alanı boşaltırız.
}
int fibonacci(int *terim,int n)
{
    int i;
    for(i=2;i<=n;i++){ //2 den başlatmamızdaki sebep zaten 0 ve 1 terimi biz tanımladık
        terim[i]=terim[i-1]+terim[i-2]; //fibonacci serisi seçilen terimin -1 küçüğü ile -2 küçüğü ile toplanır ve ona
        atanır
    }
    return terim[n];
}
```

[Belgenin Başı](#)

7.Dinamik Bellek Kullanımı Belgenin

Başı

Bir C programı içerisinde, dizilerin boyutu ve kaç elemanlı olduğu program başında belirtilirse, derleyici o dizi için gereken bellek alanını (bölgesini) program sonlanıncaya kadar saklı tutar ve bu alan başka bir amaç için kullanılamaz [1]. Bu türden diziler statik dizi olarak isimlendirilir. Statik dizinin boyutu programın çalışması esnasında (run time) değiştirilemez.

Fakat, programın çalışırken bir dizinin boyutu ve eleman sayısı bazı yöntemler kullanılarak değiştirilebilir. Bu tür dizilere dinamik dizi denir. Dinamik diziler için gereken bellek bölgesi, derleyici tarafından işletim sisteminden istenir, kullanılır ve daha sonra istenirse bu bölge boşaltılır. Bu konuda dinamik dizi oluşturmayı göreceğiz ve bunları bir hazır kütüphane fonksiyonları ile yapacağız.

7.1 Dinamik Dizi Fonksiyonları:

Dinamik diziler işaretçi kullanılarak ve standart kütüphane <stdlib.h> fonksiyonlarıyla malloc(), calloc(), realloc() ve free() yardımıyla ile oluşturulur veya boşaltılır. Pointer bir değişkenle kullanılır.

Dinamik Bellek Fonksiyonu	Açıklaması	Kullanımı
malloc();	Bellekte herbiri size_t tipinde olan <i>eleman_sayısı</i> kadar yer (bellek bloğu) ayırır. Bu yer verilmezse geriye NULL gönderir.	<code>degiskenimiz = (int *) malloc(sizeof(int)*n);</code> //integer cinsinden hafızada gerekli işaretçiye n kadarlık bir byte ayırır.
calloc();	Bellekte herbiri nbayt kadar yer işgal edecek <i>eleman_sayısı</i> kadar boş yer ayırır ve bütün bitleri sıfırlar. Bu yer ayrılamazsa geriye NULL gönderir.	<code>degiskenimiz = (int *) calloc (sizeof(int)*n);</code>

realloc();	<i>ptr</i> işaretçisi ile gösterilen bellek bloğunu, <i>nbayt</i> kadar büyütür veya küçülterek değiştirir. Bu iş gerçekleşmezse geriye NULL gönderir.	degiskenimiz=realloc(degiskenimiz,2) //realloc ile hafızada 2*nbayt kadarlık yer daha açtı.
free();	Daha önce ayrılan adresi <i>ptr</i> 'de saklanan bellek alanının boşaltır.	free(degiskenimiz); //değişkenimiz için ayırdığımız bellek alanını boşalttık bunu her bellek ayrılan dinamik dizimiz için yapmalıyız yoksa bellekte yer kaplar.

7_1_ornek.c) Hafızada yer açılarak dinamik dizi ile istenen sayıdan başlayarak 0 a kadar olan sayıların faktoriyelini alan ve adreslerini gösteren kodu yazınız.

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    int n;
    printf("Kaca kadar sayinin faktoriyelini bulmak istersiniz\n");
    scanf("%d",&n);
    int islem=1;
    int islemdizisi[n];
    int i,j,a=0;
    int *faktoriyeldizisi;
    faktoriyeldizisi=(int *) malloc( sizeof(int)*n );
    faktoriyeldizisi=islemdizisi;
    for(i=n;i>0;i--,a++) //sayıların kaçtan başlayacağı ve kaç a kadar olacağı a ise pointerımızın değeri
    {
        islem=1; //her yeni bir sayıda islem ifadesi 1 e döndürülür
        for(j=i;j>0;j--) //j=i den başlar ve 0 a kadar olan sayılara azalır
        {
            islem*=j; //her j değeri islem ile carpılarak döngü tamamlanır
            islemdizisi[a]=islem;
        }
        printf("%d.faktoriyelim=%d\t%p\n",i,*(faktoriyeldizisi+a),faktoriyeldizisi-a); //hangi sayının faktoriyeli ve sayının faktoriyel hali ve sayının büyükten başlayarak kucuge kadar hafızadaki adresleri
    }
    free(faktoriyeldizisi);
}
```

[Belgenin Başı](#)

7_2_ornek.c) 5 tane asal sayıyı bulan ve relloc fonksiyonu ile bu asal sayıları artıran programı yazınız.

```
#include <stdio.h>
#include <stdlib.h> //realloc ve malloc fonksiyonu var
```

```

int main()
{
    int n;
    printf("Kac tane asal sayi yazdirmak istersiniz=");
    scanf("%d",&n);
    int mydizi[n]; //10 tane asal sayinin bulunacağı diziyi tanımladık
    int *ptrdizi; //pointer dizimizi tanımladık
    int i,j,a=0;
    ptrdizi=(int*)malloc(sizeof(int)*5);
    int sayac=0; //sayac 0 dan başladı asal sayı kontrolü için oluşturduk
    ptrdizi=(int*)realloc(ptrdizi,2); //realloc fonksiyonu ile asal sayimizi nbayt*2 ile yükselttik.
    ptrdizi=mydizi; //başlangıç değerini atadık dizimizin
    for(i=2;i<100,a<n+2;i++) //asal sayilar 2 den başladığı için sayimizi 2 den başlattık 10 tane dizi
    elemanımız olunca döngü bitecek
    {
        sayac=0; //her döngüde sayac 0'lanacak asal sayi için
        for(j=2;j<i;j++) // j=2 yi 2 den başlattık çünkü asal sayıların kuralı 1 e ve kendisine
        bölünemeyen sayılar i den küçük yaptık ve kendisine bölünemeyeceği anlamını verdik
        {
            if(i%j==0) //i%j== bölüp kalanı 0 a esitse
            {
                sayac++; //sayacı ++artırdık
                break; //döngüyü kırdık
            }
        }
        if(sayac==0) //sayac 0 a esitse
        {
            mydizi[a]=i; //i.sayıyı dizimize atadık
            a++; //dizimizin eleman sayisini artırdık
        }
    }
    for(i=0;i<n+2;i++)
    {
        //printf("%d.dizi\n",mydizi[i]);
        printf("\n%d.Asal sayi=%d,ve adresi=%p\n",i+1,*(ptrdizi+i),(ptrdizi+i)); //kacinci asal sayi
        olduğunu yazdık ve adresteki asal sayiyi işaret ettik ve adresini gösterdik sırasıyla
    }
    free(ptrdizi);
}

```

Belgenin Başı

8.Karakter Dizileri Strings [Belgenin Başı](#)

Karakter dizileri metin,isim,soyisim tutan tek boyutlu çift boyutlu dizilerdir.

Sadece karakter dizilerine özel olarak, karakter dizilerinin sonuna sonlandırıcı karakter olarak adlandırılan bir simge eklenir.Sonlandırıcı karakter,dizinin bittiği yeri gösterir,ASCII tablosunun sıfır numaralı ('\0') karakteridir.

Karakter dizisi tanımlama:

```
char name [6]={'d','o','g','a','n','\0'}; //bu 0 değerini biz vermeyiz ama bilgisayar kendiliğinden sonlandırıcı karakter olarak ekler
```

```
char calisanlar[4][10]={"Dogan","Dorukan","Duran","Hüseyin"} //burada 4 işçinin işimlerini yazdırdık char dizisi yardımı ile 4 işçi sayısıdır 10 işe işçilerin isim uzunluklarının boyutudur
```

puts();

Printf gibi ekrana yazı yazdırma fonksiyonudur,printf den tek farkı yazıyı yazdıktan sonra ir satır imleci aşağı kaydırmasıdır.<stdio.h> kütüphanesinde yer alır.

gets();

Bu fonksiyon c dilinde <stdio.h> kütüphanesinde yer alır.Scanf gibi işlemi vardır ama scanf'den farkı scanf boşluk görene kadar değeri kullanıcıdan alır,gets(); fonksiyonu ise satır bitene kadar devam eder.

```
scanf("%s",&name);
```

```
gets(name);
```

8.1.Bazı Katar Fonksiyonları <string.h>:

Bu fonksiyonlar standart C'de iki katarı karşılaştırmak, bir katarın içeriğini diğerine kopyalamak ve katarın uzunluğunu bulmak vb işlemler için tanımlı fonksiyonlardır. Bu ve benzeri fonksiyonlar kullanılırken string.h kütüphanesi programın başına ilave edilmelidir.

String.h kütüphanesi fonksiyonları

Fonksiyon	Açıklama	Kullanımı
int strcmp(char str1,char str2);	str1 ve str2 yi karşılaştırır. Eşitse 0, str1 büyükse 0'dan büyük bir değer aksi halde	if(strcmp(str1,str2)=0) //aynısıdır if(strcmp(str1,str2)>0) //str1

	O'dan küçük bir değer gösterir.	karakterinin uzunluğu str2 den büyüktür if(strcmp(str1,str2)<0)//str2 karakterinin uzunluğu str1 den büyüktür.
strcpy	Boş bir string dizisine Diğer diziyi kopyalar.	strcpy(name,truename);
strcat	2 değişkeni 1 değişkene ekler	strcat(name," "); //bosluk karakteri strcat(name,surname);
strrev();	İstenen stringi ters çevirir.	strrev(name);
strlen();	İstenen stringin uzunluğunu belli eder.	strlen(name);
strchr	Birinci değişkenin ikinci değişken içindeki konumunu belirler. Yoksa null döndürülür.	strchr(name,surname);
strstr	1.değişken katarının 2.değişken içerisindeki ilk karşılaştığı yeri verir.	strstr(name,surname);
strlwr	İşaret edilen katarın bütün karakterlerini küçük harfe çevirir.	strlwr(name);
strupr	İşaret edilen katarın bütün karakterlerini büyük harfe çevirir.	strupr(name);

8_1_ornek.c) Kullanıcı id,sifre tanımlı klavyeden girilen değerle karşılaştıran ve eğer doğru ise anne kızlık soyadini soran eğer oda doğru ise şifre değişikliği yapmasını sağlayan şifre değiştirme panelini yapınız ve yenisifreyi tanımlı sifreye kopyalayınız.

```
#include <stdio.h>
#include <string.h>
int main()
{
    printf("Sifremi degistir paneline hosgeldiniz=\n");
    char tanimliid[20]="dogan06"; //tanimli id
    char tanimlisifre[20]="dogan0658"; //tanimli sifre
    int hak; //3 giriste sistem kapamacak onun için hak
    char kizliksoyadi[10]="CAN"; //sifre ve id dogru ise kullanılacak
    char isim[10]="DOGAN"; //tanimli kullanıcı ismi
```

```

char merhaba[10]="MERHABA "; //merhaba mesajı
char yenisifre[10]; //degistirecegimiz sifrenin tutulacagi yer
while(hak<3){
char kullanicid[20]; //giris için kullanıcı id
char sifre[20]; //giris için sifre
printf("Kullanıcı adını giriniz=");
scanf("%s",&kullanicid);
printf("Sifrenizi giriniz=");
scanf("%s",&sifre);
strlwr(kullanicid); //girilen kullanıcı id küçük harfe cevirdik
strlwr(sifre); //girilen sifreyi küçük harfe cevirdik
if(strcmp(tanimliid,kullanicid)==0 && strcmp(tanimlisifre,sifre)==0){ //tanimli olan id girilen id ve
girilen sifre tanimli sifreyi karşılaştırdık
printf("Basarili giris=\n"); //dogru ise basarili bir giristir
printf("%s hosgeldiniz=\n",strcat(merhaba, isim)); //kullanıcının ismini mesajımızla birlikte yazdırdık
do{
char oncekisoyadi[10]; //soyad girişi için
printf("Anne kizlik soyadinin uzunlugu=%d bunu dogru sekilde giriniz=\n",strlen(kizliksoyadi));
//tanimli olan kizlik soyadinin uzunlugunu gösterdik
scanf("%s",&oncekisoyadi);
strupr(oncekisoyadi); //kullanıcıdan aldığımız kizlik soyadı degiskenini büyük harfle yazdırdık
if(strcmp(kizliksoyadi,oncekisoyadi)==0) //eger tanimli kizlik soyadi ve girilen soyad aynı ise
{
printf("Yeni sifrenizi giriniz"); //yeni sifre paneline gectik
scanf("%s",&yenisifre);
printf("\nYeni sifreniz tanimlanmistir=%s\n",strcpy(tanimlisifre,yenisifre)); //yeni
sifreyi tanimli sifremize kopyaladik
}
else
{
hak++;
printf("Hatali soyadi girisi,kalan hak=%d\n",3-hak);
}
}while(hak==3);
printf("Hatali kizlik soyadi girisi\n");
}
else
{
hak++;
printf("Hatali giris kalan hakkiniz=%d\n",3-hak);
}
}
printf("Cikis yapilmistir");
}

```

8_2_ornek.c) Girilen bir metinde secime göre kelime araması yapan, harf araması yapan baştan ve sondan ,metni ters ceviren ve belirlenen tuşa başıldığında programı kapatan programı yazınız.

```

#include <stdio.h>
#include <string.h>
int main()
{
char metin[53],cumle[23];
char *sonuclar; //girilen kelimenin bulunduğu yeri işaret etmek için kullandık
int secim; //secim yapmak için
char *bas,*son; //bas son karakterlerinin tutulduğu yeri işaret etmek için tanımladık

```

```

char basi,sonu; //bas ve sondaki aratmak istedigimiz harf girisi için
printf("Aratmak istediginiz metni giriniz=");
gets(metin); //bosluk karakterini de dahil ederek metni almak için
do{
printf("\nKelime aramasi icin=1\nHarf aramak icin=2\nTers cevirmek icin=3\nCikmak icin:4 basiniz=");
scanf("%d",&secim);
switch(secim)
{
    case 1: //secim 1 se kelime araması
        printf("Aramak istediginiz kelimeyi giriniz=");
        scanf("%s",&cumle);
        sonuclar=strstr(metin,cumle); //aranan metin ve arayan kelime degiskenleri
        printf("Aranan kelime=%s\n",sonuclar);
        break;
    case 2:
        printf("\nAratmak istediginiz harf bas icin");
        scanf(" %c",&basi);
        bas=strchr(metin,basi); //aranan metin ve kelimenin basindan itibaren aranan
        karakter
        printf("\nAranan bastan basliyerek pozisyon=%d\n",bas-metin+1);
        printf("\nAratmak istediginiz harf sondan bakmak icin");
        scanf(" %c",&sonu);
        son=strrchr(metin,sonu); //right sagdan basliyerek ara anlamına gelir bunun için çift r
        kullandik kelimenin sonundan itibaren
        printf("Aranan sondan basliyerek pozisyon=%d\n",son-metin+1);
        break;
    case 3:
        printf("Metin ters cevirecek\n");
        printf("\n%s",strrev(metin)); //metni ters cevirme için gerekli fonksiyon
        break;
    case 4:
        printf("Cikis secildi\n"); //cikis islemi secildiğini söyler
        break;
    default:
        printf("Tekrar secim yapiniz\n"); //4 secimden biri degilse dondu döner
}
}while(secim!=4); //4 e basılmadığı sürece döner
printf("Program kapatildi"); //basilrsa döngü durur program kapanir
return 0;
}

```

[Belgenin Başı](#)

Belgenin Başı

9. Matematik İşlemler

Matematiksel fonksiyonlar matematik işlemlerini daha kolay bir şekilde yapmamıza olanak sağlar. Bu fonksiyonları program içinde kullanmak için `#include <math.h>` kütüphanesini eklemeliyiz. `Math.h` kütüphanesi fonksiyonları bkz.

Fonksiyon Bildirimi	Açıklama	Kullanımı	Sonucu
<code>int abs(int x);</code>	X tam sayısı eğer negatif bir tam sayı ise bu sayıyı pozitif çevirir yani mutlak değer işlemidir	<code>abs(-5);</code>	5
<code>float fabs(float x);</code>	X gerçel sayısının mutlak değerini hesaplar.	<code>fabs(-5.1);</code>	5.10000
<code>float sqrt(x);</code>	Karekökünü hesaplar	<code>sqrt(6);</code>	36
<code>int floor(float x);</code>	X'i x den küçük olan en yakın tamsayıya gönderir.	<code>abs(-0.5);</code>	-1
<code>int ceil(float x);</code>	X'i x den büyük olan en yakın tamsayıya gönderir.	<code>ceil(-0.5);</code>	0
<code>float pow(x,y);</code>	X^y değerini hesaplar	<code>pow(5,3);</code>	125
<code>float log(float x);</code>	Pozitif x sayısının doğal logaritmasını hesaplar $\ln(x)$;	<code>log(5.0);</code>	1.6094379
<code>float log10(float x);</code>	Pozitif x sayısının 10 tabanındaki logaritmasını hesaplar.	<code>log10(5.0);</code>	0.69897

float sin(float x);	Radyan cinsinden girilen x sayısının sinüs değerini hesaplar.	sin(6.0);	0.104528
float cos(float x);	Radyan cinsinden girilen x sayısının kosinüs değerini hesaplar.	cos(8.0);	0.990268
float tan(float x);	Radyan cinsinden girilen x sayısının tanjant değerini hesaplar.	tan(5.0);	0.0874886
float asin(float x);	Sinüs değeri x olan açıyı gönderir. Açı – $\pi/2$ ile $\pi/2$ arasındadır.	asin(0.5);	0.523599
float acos(float x);	Cosinüs değeri x olan açıyı gönderir. Açı – $\pi/2$ ile $\pi/2$ arasındadır.	acos(0.5);	1.047198
float atan(float x);	Tanjant değeri x olan açıyı gönderir. Açı – $\pi/2$ ile $\pi/2$ arasındadır	atan(0.5);	0.463648

9_1_ornek.c) Matematik kütüphanesi kullanılarak istenen işlemleri secim doğrultusunda yaptıran programı yazınız. Secimler; Mutlak deger, karekok hesabi icin, üssünü alma, ln alma, 10 tabanında log, sinus hesaplama, cosinus hesaplama, tanjant hesaplama, cotanjant hesaplama gibi işlemler yapan.

```
#include <stdio.h>
#include <math.h>
int main()
{
    int sayi, sayi2;
    printf("Matematik kütüphanesi islemlerine hoşgeldiniz\n");
    int secim;
    do //döngü kurduk çünkü birden fazla işlem yaptırmak istenebilir
    {
        printf("İslem yaptırmak istediginiz sayiyi giriniz=");
        scanf("%d", &sayi);
        printf("\nSecim yapiniz\nMutlak deger=1\nKarekok hesabi icin=2\nUstunu alma=3\nLn alma=4\n10 tabanında log=5\nSinus hesaplama=6\nCosinus hesaplama=7\nTanjant hesaplama=8\nCotanjant hesaplama=9\nCikmak icin=0 basiniz=");
        scanf("%d", &secim);
        switch(secim)
        {
```

```

        case 1:
            printf("Sayinin mutlak degeri=%d\n",abs(sayi));
            break;
        case 2:
            printf("Sayinin karekok degeri=%0.2f\n",sqrt(sayi)); //değerleri ekrana yazdırırken float kullandık
            //cünkü bu fonksiyonlar float deger döndürür
            break;
        case 3:
            printf("Sayiyi giriniz=");
            scanf("%d",&sayi2);
            printf("Ussunu alma =%0.2f\n",pow(sayi,sayi2));
            break;
        case 4:
            printf("Ln sonucu=%0.2f\n",log(sayi));
            break;
        case 5:
            printf("Logaritma sonucu 10 tabanında=%0.2f\n",log10(sayi));
            break;
        case 6:
            printf("Sinus hesaplama=%0.2f\n",sin(sayi));
            break;
        case 7:
            printf("Cosinus hesaplama=%0.2f\n",cos(sayi));
            break;
        case 8:
            printf("Tanjant hesaplama=%0.2f\n",tan(sayi));
            break;
        case 9:
            printf("Cotanjant hesaplama =%0.2f\n",1/tan(sayi));
            break;
        case 0:
            printf("Cikis islemi secildi\n");
            break;
        default:
            printf("Hatali islem denemesi tekrar deneyiniz");
            break;
    }
}while(secim!=0); //0 a başılmadığı sürece döngü devam etsin.
    printf("Islemler tamamlanmistir kapatildi");
}

```

9_2_ornek.c)Matematik fonksiyonları ve normal bir fonksiyon kullanarak sayı yuvarlama işlemini yapan programı yazınız.

```

#include <stdio.h>
#include <math.h>
float sayiyuvarlama(float sayi1);
int main()
{
    float sayi1;
    printf("Ondalikli sayiyi giriniz=");
    scanf("%f",&sayi1);
    sayiyuvarlama(sayi1); //fonksiyona girilen sayiyi girip döndürdük işlemi
}
float sayiyuvarlama(float sayi) //float bir fonksiyon oluşturduk çünkü sayimiz float ve geri döndürülecek sonuç
değeride floattır.
{

```

```
int secim;
printf("Sayiyi kucuk degere yuvarlamak icin=1\nBuyuk degere yuvarlamak icin=2\n");
scanf("%d",&secim);
if(secim==1)
{
    printf("Sayinin kucuge yuvarlanmasi=%0.2f",floor(sayi));
}
else if(secim==2)
{
    printf("Sayinin buyuge yuvarlanmasi=%0.2f",ceil(sayi));
}
else
printf("Hatali giris tekrar seciminizi yapiniz");
}
```

Belgenin Başı

Belgenin Başı

10.Yapılar,Structures Deyimi

Aralarında mantıksal bir ilişki bulunan farklı türden bilgiler "yapılar (structures)" içerisinde mantıksal bir bütün olarak ifade edilir. Yapılar diziler gibi bellekte sürekli kalır. Bir yapı içerisindeki elemanlara üye (member) denir. Üyelerin herbiri farklı veri tipine sahip olabilir. C, kullanıcının kendi veri tipini tanımlamasına müsaade eder bu olay aracılığıyla.

Kullanımı :

```
typedef struct etiket_adi{  
degisken türü degisken_ismi; //char baslik[50];  
degisken türü degisken_ismi; //int kitap_no;  
}etiket_adi;
```

Typedef deyimi kullanmadaki amaç:Çünkü kullandığımız bu yapıda değişken tanımlamak için tekrar struct deyimine ihtiyacımız olmicak.

Structırların üyelerini ana main blogunda tanımlarken etiketten sonra kaçınıcı üye olduğunu belli ederek bir tanımlama yapılır.

```
etiket_adi member1;
```

Pointer tanımlamada ise;

```
etiket_adi *member2;
```

Pointer tanımladığımız bir structırdaki malloc fonksiyonunu kullanıp bellekte yer açmalıyız.

```
member2 =(etiket_adi*)malloc(sizeof(etiket_adi));
```

Her üye için bunu yapmalıyız malloc ile hafızada yer açmalıyız.

Normal tanımlı structıra klavyeden değer girmek ve ekrana yazdırmak;

```
scanf("%d",&member1.kitap_no); //klavyeden kullanıcı girişi.
```

```
printf("%d",member1.kitap_no); // ve bu kullanıcının girdiği değeri ekrana yazdırmak.
```

Pointerlarda üyeye değer girmek ve ekrana yazdırmak;

```
scanf("%d",&member2->kitap_no);
```

```
printf("%d",member2->kitap_no)
```

shallow copy: Sadece adreslerin kopyalandığını ifade eder.

deep copy: Sadece adresleri değil aynı zamanda içindeki bilgilerin de kopyalandıklarını ifade eder.

Arasındaki fark: Temel olarak yeni veya malloc tarafından yapılan ve bir üye işaretçisi ile gösterilen her şeyin deep copy kopya için ayrı olarak tahsis edilmesi gerekir.

Bit Alanı: (Bit Field):

Bit –alanı (bit field) bellekte bir tamsayı için ayrılan alanın belli bir kısmındaki bir ya da daha fazla yan yaba bit’ten oluşur. Bir bit alanı yalnızca int, signed int ya da unsigned int olarak bildirilebilir. Int olarak bildirildiğinde derleyiciye bağlı olarak signed ya da unsigned olabilir. Dolayısıyla programların taşınabilir olması için signed ya da unsigned olduğu belirtilmelidir.

Kullanımı:

```
struct etiket_adi{  
    unsigned alan_ismi:genislik;  
}etiket_adi;
```

Structırların Fonksiyonlarla Kullanımı:

Main blogu içinde fonksiyona gönderilmek istenen üye member belirtilir.

Gonderme işlemi

```
Fonksiyon_ismi(member1);
```

Fonksiyon tanımlama yapılırken;

```
void fonksiyon_ismi(tag member1); //void tipinde structırın degeri döndürölür
```

```
{
```

```
    kod blokları;
```

```
}
```

```
void fonksiyon_ismi(tag * member 2); //burada pointer tipli bir üye aldık
```

```
{
```

```
    kod blokları;
```

```
}
```

[Belgenin Başı](#)

10_1_ornek.c) Bir alışveriş merkezinde tanımlı firmanın geliş fiyatı 5 ve satis fiyatı 10 TL’dir,kalemin ise geliş fiyatı 2 ve satis fiyatı 5 TL’dir.Bu alışveriş merkezindeki ürünleri tanımlayıp ekrana yazdıran programın fonksiyonunu yazınız ikinci üründe pointer kullanınız.

```
#include <stdio.h>
#include <string.h> //memcpy fonksiyonu bu kütüphanede bulunur
#include <stdlib.h> //malloc fonksiyonu bu kütüphanede bulunur

typedef struct urun
{
    char urun_adi[50];
    int gelis_fiyati;
    int satis_fiyati;
}urun;
void goster_urun( urun urun1);
void goster_urun2(urun *urun2);
int main()
{
    urun urun1;
    memcpy(urun1.urun_adi,"Firca",50); //bellege tanimli ürünü kopyaladık
    urun1.gelis_fiyati=5; //normal bir tanımlama olduğu için . kullandık
    urun1.satis_fiyati=10;
    goster_urun(urun1); //urunumuzu göstermek için fonksiyona gönderdik.
    urun *urun2; //2.ürünümüzü tanımladık
    urun2=(urun*)malloc(sizeof(urun)); //hafızada ikinci ürüne yer açtık
    memcpy(urun2->urun_adi,"Kalem",50);
    urun2->gelis_fiyati=2;
    urun2->satis_fiyati=5;
    goster_urun2(urun2); //ikinci ürünümüzü fonksiyona döndürdük
    return 0;
}
void goster_urun(urun urun1) //ürün1 in gosterildiği fonksiyon
{
    printf("Tanimli Urun bilgisi\n");
    printf("Urun isim=%s\n",urun1.urun_adi);
    printf("Urun gelis fiyatı=%d\n",urun1.gelis_fiyati);
    printf("Urun satis fiyatı=%d\n",urun1.satis_fiyati);
}
void goster_urun2(urun *urun2) //ürün2 nin gösterildiği fonksiyon member ismini pointer olarak tanımladığımız için işaretçi koyduk
{
    printf("Tanimli Urun bilgisi\n");
    printf("Urun isim=%s\n",urun2->urun_adi); //ürünleri ekranda yazdırmak için -> bunu kullandık çünkü üyemiz pointer.
    printf("Urun gelis fiyatı=%d\n",urun2->gelis_fiyati);
    printf("Urun satis fiyatı=%d\n",urun2->satis_fiyati);
}
```

10_2_ornek.c) Şirkette çalışan 2 kişinin adı ve soyadını,mevkisini,maaşını kullanıcıdan alan ve ekrana yazdıran programı yapınız.

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

typedef struct isci{ //structimizi tanımladık
```

```

char namesurname[20];
char mevk[20];
int maaas;
}isci;
int main()
{
    isci isci1; //sınıf açtık normal
    isci *isci2;
    isci2=(isci*)malloc(sizeof(isci)); //hafızada isci2 ye yer ayırdık

    printf("Iscinin ad soyadi giriniz=\n");
    gets(isci1.namesurname); //sınıfa isim soyisim mevk maas ekledik
    printf("Iscinin mevkisi=\n");
    gets(isci1.mevki);
    printf("Iscinin maasi=\n");
    scanf("%d",&isci1.maaas);
    printf("2.Iscinin ad soyadi giriniz=\n");
    fflush(stdin); //tamponu temizlemek için kullandım çünkü gets ile aldığım namesurname değerini
    görmeden mevk bölümüne geciordu.
    gets(&isci2->namesurname); //2.iscimize namesurname değerini arada boşluk olmadan almak için
    kullandık.Pointer olduğu için başına &koyduk
    printf("2.Iscinin mevkisi=\n");
    scanf("%s",&isci2->mevk);
    printf("2.Iscinin maasi=\n");
    scanf("%d",&isci2->maaas);
    printf("\nTanimli isci adi ve soyadi=%s",isci1.namesurname); //bu verileri ekrana yazdırdık
    printf("\nTanimli isci mevkisi=%s",isci1.mevki);
    printf("\nTanimli isci maasi=%d",isci1.maaas);
    printf("\nTanimli isci adi ve soyadi=%s",&isci2->namesurname); //bu verileri ekrana yazdırdık pointer
    olduğu için ekrana yazdırmada -> kullandık
    printf("\nTanimli isci mevkisi=%s",isci2->mevk);
    printf("\nTanimli isci maasi=%d",isci2->maaas);
    return 0;
}

```

10_3_ornek.c)Fenerbahce futbol takiminin oyuncularini ve oynadığı mevkileri kullanicidan alan programi yaziniz.

```

#include <stdio.h>
#include <string.h>
#define takim 11 //futbol takimi 11 kisi olduğu için girişte tanımladık
typedef struct futbolcu{
    char name[50];
    char mevk[50];
}futbolcu;
typedef struct fenerbahce{
    futbolcu *futbolcular;
    int sayi;
}fenerbahce;
void futbolcu_ekle(futbolcu *player);
void futbolcu_goster(fenerbahce *player);
int main()
{
    fenerbahce *fenerbahcem;
    fenerbahcem=(fenerbahce*)malloc(sizeof(fenerbahce)*takim); //takimin boyutunu futbolcu sayısı ile
    carpilmasi kadar da bir yer ayırdık
    fenerbahcem->sayi=takim; //fenerbahce takimina takimin mevcudunu işaretledik
    fenerbahcem->futbolcular=(futbolcu*)malloc(sizeof(futbolcu)*takim); //takimin futbolcu sayisi kadar
    futbolculuk yer ayırdık
}

```

```

        int i;
        for(i=0;i<takim;i++)
        {
            futbolcu_ekle(&(fenerbahcem->futbolcular[i])); //takimdaki futbolcular structürünü
            futbolcu_ekleye gönderdik gerekli özelliklerin girilmesi için
        }
        futbolcu_goster(fenerbahcem);
        return 0;
    }
    void futbolcum (futbolcu *players) //gelen her bir futbolcu structürünü ekrana yazdırdık
    {
        printf("Futbolcu bilgileri=\n");
        printf("Futbolcu adi ve soyadi=%s\n",players->name);
        printf("Futbolcu oynadigi mevki=%s\n",players->mevki);
    }
    void futbolcu_goster(fenerbahce *player) //takimin işaretli futbolcularini gosterdik
    {
        int i;
        for(i=0;i<player->sayi;i++)
        {
            futbolcum(&(player->futbolcular[i])); //playerda bulunan futbolcuları futbolcum
            fonksiyonuna gönderdik
        }
    }
    void futbolcu_ekle(futbolcu *player) //işaretli futbolcuları playere işaretleyip gerekli futbolcuları aldık
    {
        printf("\nFenerbahce Futbol Takiminin futbolcularini ve özelliklerini giriniz:\n");
        printf("Futbolcu ismini giriniz=\n");
        fflush(stdin);
        gets(player->name);
        printf("Futbolcu hangi bolgede oynuyor\n");
        scanf("%s",&player->mevki);
    }
}

```

10_4_ornek.c) Kullanıcı tarafından ismi girilen şirketin ne kadar ürün bilgisi gireceğini belirleyen ve belirlenen ürünlere göre geliş fiyatını ve satış fiyatını belirleyen ve fiyata göre her ürünün kazançını hesaplayan programı yazınız.

```

#include <stdio.h>
#include <string.h>
#include <stdlib.h>
typedef struct urun //ürünün özelliklerinin tutulduğu structür
{
    char urunname[50];
    int gelis_fiyati;
    int satis_fiyati;
}urun;
typedef struct sirket{ //sirkete girilen ürünler
    char sirketname[50]; //sirketin ismi tek seferlik giriş
    urun *urunler; //sirkete gelen ürünleri işaretlettik
    int sayi; //sirkette kaç tane ürün olacağı bilgisi
}sirket;
void urun_ekle(urun *urunum); //urun etiketli structürümüzü urunum de işaretledik
void urunleri_goster(sirket *urunum); //sirketimize urunum adlı işletciyi gönderdik
int main()
{
    int adet,i=0;
    printf("Kac adet urun girmek istersiniz=");
}

```

```

scanf("%d",&adet);
sirket *sirketim; //sirketimiz structürüne sirketimi tanımladık
sirketim=(sirket*)malloc(sizeof(sirket)*adet); //sirketime girilen ürün adeti kadar yer ayırdık
sirketim->sayi=adet; //sayi degiskenimize yolladık girilen ürün sayısını
sirketim->urunler=(urun*)malloc(sizeof(urun)*adet); //ürünler structürümüzdeki ürün özellikleri kadar
yer ayırdık ürünler e
printf("Sirket ismini giriniz=");
fflush(stdin); //peş peşe string ifadelerde kullanıcıdan almak için
gets(sirketim->sirketname); //sirketime sirketisini kullanıcıdan aldık
for(i=0;i<adet;i++)
{
    urun_ekle(&(sirketim->urunler[i])); //sirketimin içinde bulunan urunler işaretcisini döngüye
soktuk her ürün girişi ve özellikleri için
}
urunleri_goster(sirketim);
return 0;
}
void urun_goster(urun *malzemeler) //girilen her urun structürümüzün ürün özelliğini malzemeler işaretcisiyle
ekrana yazdırdık
{
    printf("\nUrun bilgisi=");
    printf("\nUrun ismi=%s",malzemeler->urunname);
    printf("\nUrun gelis fiyati=%d TL",malzemeler->gelis_fiyati);
    printf("\nUrun satis fiyati=%d TL",malzemeler->satis_fiyati);
    printf("\nUrun kazanc=%d TL",((malzemeler->satis_fiyati)-(malzemeler->gelis_fiyati)));
}
void urunleri_goster(sirket *urunum) //girilen ürünleri göstermek için
{
    int i;
    for(i=0;i<urunum->sayi;i++) //urunumun içinde kaç tane ürün varsa döngü o kadar dönecek
    {
        urun_goster(&(urunum->urunler[i])); //herbir ürün özelliği için urun_goster fonksiyonuna
gittik
    }
}
void urun_ekle(urun *urunum) //urun etiketli structürümüzü urunum de işaretledik
{
    printf("Urun ismini giriniz\n");
    fflush(stdin);
    gets(urunum->urunname);
    printf("Urun gelis fiyati\n");
    scanf("%d",&urunum->gelis_fiyati);
    printf("Urun satis fiyati\n");
    scanf("%d",&urunum->satis_fiyati);
}

```

10_5_ornek.c) Bit alanını gösteren programı yazınız.

```

#include <stdio.h>
struct Bitalani
{
    unsigned alan1:1,alan2:1; //işaretsiz tam sayı olarak alanlarımızı tanımladık
}bitdeger;
int main()
{
    bitdeger.alan1=1; //alanımıza 0 ya da 1 koyduk
    bitdeger.alan2=0;
}

```

```

        printf("Bit degeri alan1=%c\n",bitdeger.alan1 ? '1':'0'); //bit degerinin alanındaki deger 1 veya 0 mı
        diye sorguladık
        printf("Bit degeri alan2=%c\n",bitdeger.alan2 ? '1':'0');
        return 0;}

```

10_6_ornek.c)Kullanıcının girdiği işçilere yine kullanıcının istediği kadar zam yapan programı yazınız.Ne kadar işçi girilmesinede kullanıcı karar versin.

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

```

```

typedef struct isci{
    char namesurname[50];
    char isi[50];
    int maasi;
    int zam;
}isci;

```

```

typedef struct sirket{
    isci *isciler;
    int sayi;
    int zam;
}sirket;

```

```

void iscileri_goster(sirket *workers);
void isci_ekle(isci *workers);

```

```

int main()
{
    int i;
    int n,zam;
    printf("Ne kadar calisaniniz var=");
    scanf("%d",&n);
    sirket *sirketim;
    sirketim=(sirket*)malloc(sizeof(sirket)*n); //sirkete sirketteki isci sayisi kadar yer aç
    sirketim->sayi=n;
    sirketim->isciler=(isci*)malloc(sizeof(isci)*n); //şirketimde bulunan işçiler işaretcisine isci sayisi ve
    isci özelliklerinin bulunduğu structtır kadar yer aç
    for(i=0;i<n;i++)
    {
        isci_ekle(&(sirketim->isciler[i]));
    }
    iscileri_goster(sirketim);
}

```

```

void isci_goster(isci *iscim) //her iscimizin özelliklerini gösterdik
{

```

```

    printf("\nIsçi bilgileri=\n");
    printf("Isçi adi ve soyadi=%s\n",iscim->namesurname);
    printf("Isçinin isi=%s\n",iscim->isi);
    printf("Isçinin eski maasi=%d\n",iscim->maasi);
    printf("Isçinin guncel maasi=%d\n",(iscim->maasi)+(iscim->zam)); //işçinin eski maasi ve eklenen

```

```

    zamı koyarak güncel maasi uyguladık
}

```

```

void iscileri_goster(sirket *workers) //şirkette bulunan işçileri ekrana yazdırdık
{

```

```

    int i;
    for(i=0;i<workers->sayi;i++) //isci sayisi kadar döngüyü döndürdük ve
    {

```

```

        isci_goster(&(workers->isciler[i])); //sirkette bulunan iscilerimizin bilgilerini isci göster
        fonksiyonunda gösterdik
    }
}

```

```

    }
}
void isci_ekle(isci *workers) //isci structürümüzü işaretleyip her işçi için özelliklerini girdik
{
    printf("\nIsci ad ve soyadi giriniz=");
    fflush(stdin); //isci ad ve soyadi giriniz den sonra yaptığı iş bölümüne geçmesin diye fflush(stdin)
    kullandık
    gets(workers->namesurname);
    printf("\nYaptigi is\n");
    scanf("%s",&workers->isi);
    printf("\nIscinin maasi=");
    scanf("%d",&workers->maasi);
    printf("\n Isciye yapılacak zam=");
    scanf("%d",&workers->zam);
}

```

[Belgenin Başı](#)

Belgenin Başı

11.Dosya Yönetimi

Verileri uzun süreli saklamak için dosyalama işlemleri yapılır.Dosyalama yapıları 2 ye ayrılır.

1. Metin(text)
2. Binary(ikili)

Bu dosya türlerinden herhangi biri açılırken belirtilebilir.Metin(text) dosyaları için,fopen(“t”); İkili binary dosyaları için fopen(“b”); kullanılır.

C’dilinde standart bir dosya tipi tanımlanmamıştır ancak “stdio.h” başlık dosyası içerisinde FILE yapısal veri tipi tanımlanmıştır.

Bir dosyalama işlemli programı çalıştırdığımız zaman standart metin akışlarını otomatik olarak açar. Bu metin akışları yoluyla, program bilgisayara bağlı aygıtlara (klavye, ekran, yazıcı gibi) tıpkı bir dosya gibi erişebilir:

Standart Akışlar	Görevi
Stdin:	Klavyeden veri okur.
Stdout:	Verileri ekrana yollar.
Stderr:	Verileri ekrana yollar.
StdauX:	Seri porttan veri alışverişi yapar.
Stdprn:	Verileri yazıcıya yollar.

Kullanımı:

FILE *dosya_adi; //istenilen dosya_adi dir.

Dosya Açma modları:

Açılış Modu	İşlem Türü
r	Salt okunur (read only). Dosyanın açılabilmesi için önceden oluşturulmuş olması gerekir. Bu modda açılmış olan bir dosyaya yazma yapılamaz.
w	Yalnızca yazma (write only). Dosya diskte kayıtlı olsun veya olamsın dosya yeniden oluşturulur. Bu modda açılmış olan bir dosyadan okuma yapılamaz.

a	Ekleme (append). Kayıtlı bir dosyanın sonuna veri eklemek için açılır. Bu modda açılmış olan bir dosyadan okuma yapılamaz.
r+	Okuma ve yazma. Bu modda açılmış olan bir dosyanın daha önce varolması gerekir
w+	Okuma ve yazma. Bu modda açılmış olan bir dosya var olsun veya olmasın dosya yeniden oluşturulur.
a+	Okuma ve yazma. Kayıtlı bir dosyanın sonuna veri eklemek için açılır.
rb	Binary ikili dosyanın okuma için açılması.
wb	Binary (ikili) dosyanın veri ekleme için açılması.
ab	Binary (ikili) dosyanın veri ekleme için açılması.
w+b	Binary (ikili) dosyayı okuma/yazma için.
a+b	Binary ikili dosyaya okuma/yazma için ekleme yapılması.

Bu modlar fopen(); komutu ile tırnak içinde kullanılır.

yaz=fopen("deneme.txt","w"); //açılmış bir dosyaya yazma işlemi için yalnızca.

Dosyalama Fonksiyonları:[Belgenin Başı](#)

Fonksiyon	Görevi
fopen()	Dosya oluşturur, açar
fclose()	Dosyayı kapatır
putc()	Dosyaya bir karakter yazar
getc()	Dosyadan bir karakter okur
feof()	Dosya sonuna gelindiğini sorgular
fprintf()	Dosyaya formatlı veri yazar
fscanf()	Dosyadan formatlı veri okur
fputs()	Dosyaya katar yazar
fgets()	Dosyadan katar okur

fwrite()	Dosyaya dizi yazar
fread()	Dosyadan dizi okur
Fseek()	Bir dosyadaki belirli bir byte'ı bulur.
ferror()	Bir hata durumunda doğru bir değer verir.
rewind()	Dosya aktif konumunu başa alır
remove()	Dosyayı siler.
fflush()	Tampon belleği siler.

Binary Dosya Fonksiyonları:[Belgenin Başı](#)

Fonksiyon	Görevi	Kullanımı
filelength();	Dosyanın uzunluğunu byte olarak verir.#include <io.h> kütüphanesi eklenmelidir.	long filelength(int dosyano)
fileno();	Belirtilen ve açık olan dosyanın numarasını verir.	int fileno(FILE *dosya_adi);
fseek();	Dosyada istenen kayda konumlanmak için kullanılır.	int fseek(FILE *dosya,başlangıç,başlama pozisyonu);
ftell();	O anda üzerinde bulunan kayıt pozisyonunun değerini byte olarak verir.	long ftell(file *dosya);
rewind();	Dosya başlangıcına konumlanmak için kullanılır.	void rewind(FILE *dosya_adi);
fwrite();	Dosyalara bilgileri yazdırmak için	fwrite(kayit_değişkeni,kayıt_uzunluğu,sayi,dosya);
	Dosyada okuma	fread(&kayit_degiskeni,kayıt_uzunluğu,sayi,dosya);

fread();	işlemini yapar.Dosyanın sonuna gelinmesi durumunda feof() fonksiyonu sıfırdan farklı bir değer dönderir.
----------	--

Metin Ve Binary Dosya Farkları:

- İkili dosyalarda veriler bellekteki gösterim biçimleriyle saklanırlar. Yani, örneğin `int` veri türündeki bir tamsayı ASCII koduna çevirilmeden bellekte saklandığı gibi iki byte'lık bir veri olarak dosyaya yazılır.
- İkili dosyaların içeriği ekranda okunamaz ve editör yardımıyla yaratılamaz. İkili dosyalar ancak programlar tarafından yaratılırlar ve yine ancak programlar tarafından okunurlar.
- Metin dosyalarında veriler ASCII koduna çevirilip saklandığı için, her okuma işleminde her verinin gösteriminin ASCII kodundan veri türünün bellekteki gösterim biçimine çevirilmesi için zaman harcamak gerekir. Aynı şekilde bir veriyi metin dosyasına yazarken de bellekteki gösterim biçiminden ASCII koduna çevrilmesi için zaman harcanır. (Bu çevirimler `scanf` ve `printf` gibi işlevler tarafından yapılır).
- İkili dosyalarla girdi çıktı yaparken bu çevirim için ayrıca bir zaman harcamaya gerek yoktur.

[Belgenin Başı](#)

11_1_ornek.c)Kullanıcının girdiği karakteri dosyaya yazdıran c kodunu yazınız.

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

```
#include <string.h>
```

```
int main()
```

```
{
```

```
    FILE *dosya1 = fopen("karakter.txt","a");
```

```
    char ch;
```

```
    printf("Hangi karakteri yazmak istersiniz\n");
```

```
    scanf("%c",&ch);
```

```
    if (dosya1== NULL)
```

```

    {
        printf("Boyle bir dosya bulunamadi");
    }
else
{
    printf("Dosya olusturuldu icine yazilacak karakter\t%c\n",ch);
    fputc(ch,dosya1); //dosyaya karakteri koy
}

printf("\nDosya yazimi basarili");
return 0;
}

```

[Belgenin Başı](#)

11_2_ornek.c)Txt dosyasına kullanıcının girdiği kelimeyi yazdıran kodu yazınız.

```

#include <stdio.h>
#include <string.h>
int main()
{
    FILE *dosya1 = fopen("kelime.txt","a");
    char kelime[23];
    int i;
    printf("Hangi keliemeyi ekrana yazdirmek istersiniz=");
    gets(kelime);
    if (dosya1== NULL)
    {
        printf("Boyle bir dosya bulunamadi");
    }
else
{
    for(i=0;i<strlen(kelime);i++)
    {
        printf("Dosya olusturuldu icine yazilacak karakter\t%c\n",kelime[i]);
        fputc(kelime[i],dosya1);
    }
}
}

```

```

        printf("\nDosya yazimi basarili");
        return 0;
    }

```

[Belgenin Başı](#)

11_3_ornek.c) Bir önceki örnekte yaptığımız txt dosyasından kullanıcının istediği harfi arayan ve kaç tane olduğunu yazdıran kodu yazınız .

```

#include <stdio.h>
#include <conio.h>
int main()
{
    FILE *dosya1;
    char ara[10];
    int sayac=0;
    dosya1=fopen("yaz.txt","r"); //dosya açma işlemi istenen konuma
    if (dosya1 == NULL) // dosya bossa buraya gidicek
    {
        puts("Dosya açılmıyor");
        getch();
        exit(0);
    }
    else{
        printf("Aramak istediginiz karakteri giriniz\n");
        scanf("%s",&ara);
        while(!feof(dosya1)) //dosya sonu gelene kadar devam et
        {
            if(strcmp(ara,(fscanf(dosya1)))=0)
                sayac++;
        }
        printf("\n%c tane harf bulundu metinde=%d",ara,sayac);
        fclose(dosya1);
        return 0;
    }
}

```

[Belgenin Başı](#)

11_4_ornek.c)Kullanıcıdan alınan isim soyisim bilgisini dosyaya yazdırmak ve her dosya açıldığında üstüne ekleyen programı yazınız.

```
#include <stdio.h>
#include <conio.h>
int main()
{
    char namesurname[50],ch[50];
    fopen("isimsoyisims.txt","r");
    printf("İsim ve soyisim bilgisi giriniz\n");
    gets(namesurname);
    FILE *dosya1,*dosya2;
    if (dosya1 == NULL) // dosya bossa buraya gidicek
    {
        puts("Dosya açılmıyor");
        getch();
        exit(0);
    }
    else
    {
        fgets(ch,50,dosya1);
        printf("Giris kaydi=%s\n",ch);
    }
    fopen("isimsoyisim.txt","a+");
    if (dosya2 == NULL) // dosya bossa buraya gidicek
    {
        puts("Dosya açılmıyor");
        getch();
        exit(0);
    }
    else
    {
        fputs(namesurname,dosya2);
```

```

        printf("Kayitli isim soyisim=%s",namesurname);

    }
    return 0;
}

```

Belgenin Başı

11_5_ornek.c)Kullanıcının girdiği sayının faktoriyelini alıp dosyaya yazdırıp sonucunu ekrana yazdıran programı yazınız.

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

```

int main()
{
    FILE *dosya2=fopen("faktoriyelbulma.txt", "a");
    int i,j=1;
    int random;
    int toplam=1;
    printf("Sayiyi giriniz=");
    scanf("%d",&i);
    random=i;
    for(i;i>j;i--)
    {
        toplam*=i;
    }
    fprintf(dosya2,"\n%d Sayinin faktoriyeli %d",random,toplam); //yazılacak dosyayı
belirttik
    fclose(dosya2);
    return 0;
}

```

11_6_ornek.c)Carpim tablosunu dosyaya yazdıran programı yazınız.

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

```

int main()
{
    FILE *dosya1=fopen("carpimtablosu.txt", "w");
    int i,j;

```



```

        for(i=1;i<=10;i++)
        {
            printf("\n");
            for(j=1;j<=10;j++)
            {
                fprintf(dosya1,"%d x %d =%d\n",i,j,i*j);
            }
        }
        fclose(dosya1);
    }

```

11_7_ornek.c) Bir market otomasyonunda kullanıcının girdiği ürünleri ;barkod no (%1000 ve 100 arasında rastgele bir değer),ürün adı ,ürün kategorisi,ürün fiyatını özellikleri ile dosyaya ekleyip dosyadan ürünleri listeleleyen ve kullanıcının istediği kadar ürünü fiyatlandırarak kodu yazınız kodu yazarken herbir işlem için fonksiyon kullanınız.

```

#include <stdio.h>
#include <string.h>
#include <time.h>
#include <stdlib.h>

```

```

typedef struct urun
{
    int barkod;
    char urun_name[50];
    char kategori[20];
    int fiyat;
}urun;

```

```

void urun_ekle(urun *urunler);
void urun_dosyala(urun urunler);
void urun_listele(void);
void urun_satinal(void);
int main()
{
    int birim;

```

```

int secenek;
int barkodum;
struct urun *marketim;
int i=1;
do{
    printf("Secim yapiniz\nUrun ekleme=1\nUrun listeleme=2\nUrun satin
alma=3\nCikmak icin=0\n");
    scanf("%d",&secenek);
    switch(secenek)
    {
        case 1:
            marketim=(urun*)malloc(sizeof(urun));
            urun_ekle(&marketim[0]);
            urun_dosyala(marketim[0]);
            break;
        case 2:
            urun_listele();
            break;
        case 3:
            urun_satinal();
            break;
    }
}while(secenek!=0);
printf("Otomasyon kapatildi iyi gunler");
return 0;
}

void urun_ekle(urun *urunler)
{
    printf("Urun bilgisi giriniz\n");
    srand(time(NULL)); //program çalıştırıldığı zaman rastgele atamak için
    urunler->barkod=rand()%1000+100; //zaman rastgele olduğu için 1000 ve 100
arasında rastgele bir atama yapar
    printf("Urun ismini giriniz=\n");
    fflush(stdin);

```

```

        gets(urunler->urun_name);
        printf("Urun kategori bilgisi giriniz\n");
        scanf("%s",&urunler->kategori);
        printf("Urun fiyatini giriniz=\n");
        scanf("%d",&urunler->fiyat);
    }

void urun_dosyala(urun ekle) //burada dosyaya ürün ekleme işlemi yapıyoruz
{
    FILE *ekleme;
    if((ekleme=fopen("marketim.dat","ab"))==NULL)
    {
        printf("Boyle bir dosya olusturulamadi");
    }
    else
    {
        fwrite(&ekle,sizeof(struct urun),1,ekleme);
        printf("Urun ekleme islemi basarili");
        fclose(ekleme);
    }
}

void urun_listele(void)
{
    struct urun okunacak;
    FILE *oku;
    if((oku=fopen("marketim.dat","rb"))==NULL)
    {
        printf("Boyle bir dosya yok okunamadi");
    }
    else
    {
        fseek(oku,0,SEEK_SET);
        while(fread(&okunacak,sizeof(struct urun),1,oku))
            printf("Urun barkod no=%d\nUrun ismi=%s\nUrun kategorisi=%s\nUrun
fiyati=%d\n",okunacak.barkod,okunacak.urun_name,okunacak.kategori,okunacak.fiyat);
    }
}

```

```

    }
    fclose(oku);
}
void urun_satinal(void)
{
    struct urun okunacak;
    int birim;
    int barkod_no;
    printf("Almak istediginiz urunun barkod no giriniz=");
    scanf("%d",&barkod_no);
    FILE *oku;
    if((oku=fopen("marketim.dat","rb"))==NULL)
    {
        printf("Boyle bir dosya yok okunamadi");
    }
    else
    {
        fseek(oku,0,SEEK_SET);
        while(fread(&okunacak,sizeof(struct urun),1,oku))
        if(barkod_no==okunacak.barkod)
        {
            printf("Urun barkod no=%d\nUrun ismi=%s\nUrun kategorisi=%s\nUrun
fiyati=%d\n",okunacak.barkod,okunacak.urun_name,okunacak.kategori,okunacak.fiyat);
            printf("Ne kadar almak istersiniz=");
            scanf("%d",&birim);
            printf("Odeyeceginiz tutar=%d\n",(birim)*(okunacak.fiyat)); //satin alma
işlemi için birim sayısı ile fiyatı çarpıyoruz.
        }
    }
    fclose(oku);
}

```

[Belgenin Başı](#)

11_8_ornek.c)Kullanıcının girdiği öğrencilerin ad soyadını,numarasını notunu dosyaya kayit edip sonra öğrencileri isteyen eğer öğrenci bulmak isteniyorsa öğrenci numarasına göre arama yapıp gerekli öğrenciyi dosyadan okuyup ekrana yazdıran programı yazınız.

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

```
#include <stdlib.h>
```

```
#include <string.h>
```

```
typedef struct ogrenci
```

```
{
```

```
    char namesurname[20];
```

```
    int notu;
```

```
    int no;
```

```
}ogrenci;
```

```
void ogrenci_ekle(ogrenci *ekle);
```

```
void ogrenci_yazdir(ogrenci ekle);
```

```
void ogrenci_listele(void);
```

```
void ogrenci_ara(int ogrenci_no);
```

```
int main()
```

```
{
```

```
    struct ogrenci *ogrenciler;
```

```
    int ogrenci_no;
```

```
    int secim;
```

```
    do
```

```
    {
```

```
        printf("Secim yapiniz\n Ogrenci eklemek icin:1\nOgrenci listelemek icin:2\n  
Ogrenci secmek icin:3\nCikmak icin:4 basiniz\n");
```

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

```
        switch(secim)
```

```
        {
```

```
            case 1:
```

```
                ogrenciler=(ogrenci*)malloc(sizeof(ogrenci));
```

```
                if(ogrenciler!=NULL)
```

```

        {
            ogrenci_ekle(&ogrenciler[0]);
            ogrenci_yazdir(ogrenciler[0]);
        }
    else
    {
        printf("Hafızada yer yok");
    }
    break;
case 2:
    ogrenci_listele();
    break;
case 3:
    printf("Ogrenci numarasını giriniz\n");
    scanf("%d",&ogrenci_no);
    ogrenci_ara(ogrenci_no);
    break;
}
}while(secim!=4);
printf("Ogrenci kaydi durduruldu");
return 0;
}

```

void ogrenci_ekle(ogrenci *student) //structürümüze ogrenci değerlerini ekleriz

```

{
    printf("Ogrenci bilgilerini giriniz=\n");
    printf("Ogrenci ismini giriniz=\n");
    fflush(stdin);
    gets(student->namesurname);
    printf("Ogrenci no giriniz=\n");
    scanf("%d",&student->no);
    printf("Ogrenci notu giriniz=\n");
    scanf("%d",&student->notu);
}

```

```

void ogrenci_yazdir(ogrenci ekle)
{
    FILE *ekleme;
    if((ekleme=fopen("ogrenciler.dat","ab"))==NULL)
    {
        printf("Dosya acilamadi");
    }
    else
    {
        fwrite(&ekle,sizeof(struct ogrenci),1,ekleme);
        printf("Ekleme gerceklestirildi\n");
        fclose(ekleme);
    }
}

void ogrenci_listele(void)
{
    struct ogrenci okunacak;
    FILE *oku;
    if((oku=fopen("ogrenciler.dat","rb"))==NULL)
    {
        printf("Ogrenci kayitlari yok");
    }
    else
    {
        fseek(oku,0,SEEK_SET); //ilk konumdan başla
        while(fread(&okunacak,sizeof(struct ogrenci),1,oku)) //dosya bitene kadar oku
            printf("Ogrenci ad ve soyad=%s\n,Ogrenci no:%d\nOgrenci
notu=%d\n",okunacak.namesurname,okunacak.no,okunacak.notu);
    }
    fclose(oku);
}

void ogrenci_ara(int ogrenci_no)
{

```

```

struct ogrenci okunacak;
FILE *oku;
if((oku=fopen("ogrenciler.dat","rb"))==NULL)
{
    printf("Ogrenci kayitlari yok");
}
else
{
    fseek(oku,0,SEEK_SET); //ilk konumdan başla
    while(fread(&okunacak,sizeof(struct ogrenci),1,oku)) //dosya bitene kadar oku
    if(ogrenci_no==okunacak.no)
    {
        printf("Ogrenci ad ve soyad=%s\n,Ogrenci no:%d\nOgrenci
notu=%d\n",okunacak.namesurname,okunacak.no,okunacak.notu);
    }
    else
    {
        printf("Boyle bir ogrenci bulunamadi"); //sirayla aradığı için eğer
öğrenci ilk seferde yoksa öğrenci bulunamadi der ve 2.sefer kontrol eder varsa yazar yoksa
yok der ve bitirir
    }
    fclose(oku);
}
}

```

[Belgenin Başı](#)

C Programlama Değerlendirme Soruları

Soru 1:

Dosyalama ile ilgili aşağıdaki işlemleri yapabilen kod bloğunu C programlama dilini kullanarak yazınız.

Numara, ad soyad, çalıştığı bölüm, adres, maaş bilgilerini içeren personel adında bir struct yapısı oluşturularak binary dosya üzerinde aşağıdaki işlemlerin yapılması istenmektedir. Yapılması istenilen işlemlerin her biri için ayrı bir fonksiyon tanımlanmalıdır.

1.Yeni Kayıt Ekle

2.Kayıt Ara

3.Kayıt Değiştir

4.Çıkış

Yeni Kayıt Ekle; Personele ait verilerin (numara, ad soyad, çalıştığı bölüm, adres, maaş) dışarıdan alınarak binary dosyaya yazılması

Kayıt Ara; Personelin numarasını kullanıcıdan alarak binary dosyada arama yapılarak personele ait bilgilerin ekranda gösterilmesi

Kayıt Değiştir; Personelin numarasını kullanıcıdan alarak personele ait hangi bilgi değiştirilmek isteniyorsa o bilgiyi değiştiren kod bloğu

Çıkış; İşlemlerin sonlandırılması.

Soru 2:

Bir telefon hattı ile yapılan son 10 aramanın süreleri bir dizide tutulmaktadır. Arama sürelerini en kısa görüşmeden en uzuna doğru birleştirmeli sıralama(merge sort) algoritmasını kullanarak sıralayan programı C programlama dili ile yazınız.

Not: Algoritma için recursif fonksiyon kullanıp her bir işlem adımındaki elde edilen çıktıları tablo halinde yazınız.

Soru 3:

Harf ve rakamlardan oluşan bir random şifre üretme sistemi oluşturulacaktır. Şifre uzunluğu kullanıcıdan alınarak dinamik dizi yapısını kullanacak şekilde bir şifre üretme sistemi tasarlanacaktır. Ardından üretilen şifreye istediğiniz herhangi bir şifreleme algoritmasını (Sezar, Hill vb.) uygulayıp son halini ekranda gösteriniz. Seçtiğiniz şifreleme algoritmasının ismini ve çalışma mantığını kısaca anlatınız.

NOT:Soruların çözümünü kendiniz denemeden cevaplar bölümüne geçmeyiniz.

[Belgenin Başı](#)

cevap1.c)

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

```
#include <stdlib.h>
```

```
#include <string.h>
```

```
typedef struct personel //personel structırim
```

```
{
```

```
    int numara;
```

```
    char ad_soyad[25];
```

```
    char bolum[15];
```

```
    char adres[25];
```

```
    int maas;
```

```
}personel;
```

```
void kayit_ekle(personel *person); //kayitlarımızın structıra eklendiği fonksiyon
```

```
void kayit_ekle_dosyaya(personel ekle); //kayitlarımızın structırdan dosyaya eklendiği  
fonksiyon
```

```
void kayit_ara(int personel_numara); //kayitların dosyadan okunup eşlesen kayıtlar ekrana  
yazdırıldığı fonksiyon
```

```
void kayit_degistir(int personel_numara); //kayitların önce okunup sonra eşlesen kayıtlar  
değiştirilme işleminin yapıldığı fonksiyon
```

```
int main()
```

```
{
```

```
    printf("Calisanlar veri tabanına hosgeldiniz\n");
```

```
    struct personel *personeller;
```

```
    int personel_numara;
```

```
    int secim;
```

```
        do //döngü tanımladım çünkü secim işlemi istenen değere kadar devam  
etmesini istiyorum
```

```
    {
```

```
        printf("Hangi islemi yapmak istersiniz\nYeni kayit ekle=1\nKayit
```

```
Ara=2\nKayit degistir=3\nCikis=4 icin basiniz\n");
```

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

```

switch(secim)
{
    case 1: //kayit ekleme seceneğidir
        personeller=(personel*)malloc(sizeof(personel)); //her personel için
hafızada yer açıcak
        if(personeller==NULL)
        {
            printf("Hafızada yer acilamadi");
        }
        else{
            kayit_ekle(&personeller[0]); //personellerimizin structura eklendiği
bölüm
            kayit_ekle_dosyaya(personeller[0]); //personellerimiz dosyaya
eklendiği fonksiyon
            break;
        }
        case 2:
            printf("Aramak istediginiz personel numarasini giriniz=\n"); //personel
aratmak için gerekli numara
            scanf("%u",&personel_numara); //unsigned yaptım çünkü - li bir değer
giremez
            kayit_ara(personel_numara); //fonksiyonuma personel_numara
parametresini gönderdim çünkü karşılaştırma yapıcım
            break;
        case 3:
            printf("Degistirmek istediginiz kisinin personel numarasini
giriniz=\n");
            scanf("%u",&personel_numara);
            kayit_degistir(personel_numara); //kullanicidan alınan personel
numara bilgisini kayit değiştir fonksiyonuna gönderdim karşılaştırma yapmak için
            break;
        case 4:
            printf("Cikis islemi secildi=\n");
            break;

```

```

        default:
            printf("Hatali secim tekrar deneyiniz\n");
            break;
    }
} while(secim!=4); //4 e basılmadığı taktirde döngü devam etsin
printf("Calisanlar veri tabani kapatildi\n");
free(personeller); //hafızadaki yer boşaltıyoruz
return 0;
}

```

void kayit_ekle(personel *person) //structürümüze personel değerlerimiz yazılacak pointer olarak yazılacak hafızada tutulması için ve dinamik dizidir

```

{
    printf("Bilgi ekleme bolumu\n");
    printf("Personel numarayi giriniz\n");
    scanf("%d",&person->numara);
    printf("Personel ad ve soyadi giriniz\n");
    fflush(stdin); //bir sonraki değer alma işlemine geçmesin diye tamponu temizledim
    gets(person->ad_soyad);
    printf("Personel calistigi bolumu giriniz\n");
    gets(person->bolum);
    printf("Personel adresi giriniz\n");
    gets(person->adres);
    printf("Personel maasi giriniz\n");
    scanf("%d",&person->maas);
}

```

void kayit_ekle_dosyaya(personel ekle) //personel structürümüze yazılan veriler eklenecek dosyaya

```

{
    FILE *ekleme;
    if ((ekleme=fopen("personeller.dat","ab"))==NULL) //dosya oluşturup kontrol ettirdim ab
dosya açma metodu ile açtım çünkü açıp ekleme yapmasınıda istiyorum
    {

```

```

    printf("Dosya acilmadi hafizada yer yok");
}
else
{
    fwrite(&ekle,sizeof(struct personel),1,ekleme); //dosyaya structürümün ekledim baştan
başlayarak
    printf("Ekleme islemi basari ile gercekleştirildi\n");
    fclose(ekleme);
}
}
void kayit_ara(int personel_numara)
{
    struct personel okunacak; //okunacak structümüzü oluşturduk
    FILE *oku;
    if ((oku=fopen("personeller.dat","rb"))==NULL) //oluşturduğum dosyayı rb dosya
açma metodu ile varsa okuttum yoksa buraya girdi
    {
        printf("Dosya acilmadi hafizada yer yok");
    }
    else
    {
        fseek(oku,0,SEEK_SET); //dosyanın 0 konumundan ve başından başla konum her
zaman arama işlemi için 0 dan başlıcağı için 2 kayıt 1 kere veri bulunamadi döndürür.
        while(fread(&okunacak,sizeof(struct personel),1,oku)) //oku dosyanın boyutuna kadar
        if(personel_numara==okunacak.numara) //eger okunan değerlerdeki numara bilgisi
kullanıcının girdiği numara ile aynı ise
        {
            printf("Numara=%d\nAd
soyad=%s\nBolum=%s\nAdres=%s\nMaas=%d\n",okunacak.numara,okunacak.ad_soyad,oku
nacak.bolum,okunacak.adres,okunacak.maas);
            //bütün bilgileri ekrana yazdir
        }
        else{
            printf("Veri bulunamadi\n"); //oyle birşey olmadigini söyle

```

```

    }
    fclose(oku); //kapat dosyayı
}
}

void kayit_degistir(int personel_numara) //kullanıcının girdiği personel_numarayı
kayit_degistir fonksiyonunda kullandım
{
    struct personel guncelle; //güncelleme yapmak için personel bilgilerini tutacak structır tanımladım
    FILE *oku; //dosyayı oluşturduk okumak için
    if ((oku=fopen("personeller.dat","rb+"))==NULL) //dosyayı okuttum ve sonradan işlem yapılabilmesi için içinde rb dosya açma metodunu kullandım
    {
        printf("Dosya acilmedi"); //dosya yok ya da boşsa bu hatayı verdirdim
    }
    else
    {
        fseek(oku,0,SEEK_SET); //dosyayı 0. konumdan başlayıp
        while(fread(&guncelle,sizeof(struct personel),1,oku)) //okuttum
        if(personel_numara==guncelle.numara) //dosyamdaki numara ile klavyeden girilen numara eşleşiyormu kontrol ettim
        {
            printf("Numara=%d\nAd
soyad=%s\nBolum=%s\nAdres=%s\nMaas=%d\n",guncelle.numara,guncelle.ad_soyad,guncelle.bolum,guncelle.adres,guncelle.maas);
            //numaralar eşleştiyse ekrana yazdırdım.
            int konum;
            konum=ftell(oku); //aktif dosyanın bulunduğu konumu aldım
            fseek(oku,konum-sizeof(personel),SEEK_SET); //konumdan personel structırının boyutunu çıkardım ve bu üyenin başına konumlandım SEEK_SET ile
            int secim;
            printf("Personelin hangi bilgisini guncellemek istersiniz\nNumara=1\nAd
soyad=2\nBolum=3\nAdres=4\nMaas=5\n"); //degistirilmek istenen personel bilgisini seçtirdim

```

```

scanf("%d",&secim);
//gerekli secilen seceneğe göre yeni bilgi kayıtlarını aldım
if(secim==1)
{
    printf("Yeni numarayı giriniz=\n");
    scanf("%d",&guncelle.numara);
}
else if(secim==2)
{
    printf("Yeni ad ve soyadı giriniz=\n");
    fflush(stdin);
    gets(guncelle.ad_soyad);
}
else if(secim==3)
{
    printf("Yeni bölümü giriniz=\n");
    fflush(stdin);
    gets(guncelle.bolum);
}
else if(secim==4)
{
    printf("Yeni adresi giriniz=\n");
    fflush(stdin);
    gets(guncelle.adres);
}
else if(secim==5)
{
    printf("Yeni maaş giriniz=\n");
    fflush(stdin);
    scanf("%d",&guncelle.maas);
}

fwrite(&guncelle,sizeof(struct personel),1,oku); //bu aldığım yeni bilgi
kayıtını okuduğum dosyanın fseek ile işaret edilen yerine yazdırdım
printf("Degistirme islemi basaralı\n");

```

```
        break;
    }
    else{
        printf("Hatali personel numarasi tusladiniz\n");
    }
}

fclose(oku); //dosyayı kapattım
}
```

[Belgenin Başı](#)


```

cevap2.c)
#include <stdio.h>
#include <stdlib.h>

void mergem(int sureler[],int l,int m,int r);
void merge_sort(int sureler[],int l,int r);
void dizi_yazdir(int sureler[],int i);
int main()
{
    int sureler[10]={10,2,3,1,4,5,6,12,9,15}; //çağrı parametrelerinin tutulduğu süreler dizisi
    int i=0;
    for(i=0;i<10;i++)
    {
        printf("%d ",sureler[i]);
    }
    merge_sort(sureler,0,10-1); //sıralama algoritmamıza gönderdik elemanlarımızı
    dizi_yazdir(sureler,10); //dizilerimizin içindeki elemanları yazdırdık 10 tane olarak

    return 0;
}

void mergem(int sureler[],int l,int m,int r) //ayırma ve birleştirme işlemleri yapılır rekursif bir fonksiyondur.
{
    int i;
    int j;
    int k;
    int n1 = m - l + 1; //ana dizimizi 2 ye bölecek şekilde n1 ve n2 yi aldım
    int n2 = r - m;
    int gecici1[n1], gecici2[n2]; //gecici dizi oluşturdum sol ve sağ için
    for (i = 0; i < n1; i++)
        gecici1[i] = sureler[l + i]; //çağrıların sürelerinin tutulduğu diziden diğer diziye eleman attım

```

```

for (j = 0; j < n2; j++)
    gecici2[j] = sureler[m + 1 + j];
i = 0; //yaptığım işlemlerden dolayı değişkenlerin sayıları değişti bunları eski haline
cevirdim
j = 0;
k = 1;
while (i < n1 && j < n2) //birleştirme işlemlerimizin olduğu yer
{

    if (gecici1[i] <= gecici2[j]) //eger sol dizideki ilk eleman sağ dizideki ilk elemanından
küçükse
    {
        sureler[k] = gecici1[i]; //sol dizideki elemanları çağrıların tutulduğu diziye atıyoruz
        i++;
    }

    else //degilse
    {
        sureler[k] = gecici2[j]; //sağ dizinin elemanlarının ilk indisinden başlayarak atıyoruz
        j++;
    }

    k++; //çağrı sürelerinin tutulduğu surelerin içindeki indis yani k değerini artırıyoruz
}
while (i < n1) //atama islemlerinde sol dizide boşta eleman kaldımı kontrol ediyoruz.
{
    sureler[k] = gecici1[i]; //varsa onları da atıyoruz sureler dizisinin devamına
    i++;
    k++;
}
while (j < n2) //atama işlemlerinde sağ dizide boş eleman kaldıysa kontrol edip
{
    sureler[k] = gecici2[j]; //sağ dizideki elemanları sureler dizisinin devamına atıyoruz
    j++;
}

```

```

        k++;
    }
}

void merge_sort(int sureler[],int l,int r)
{

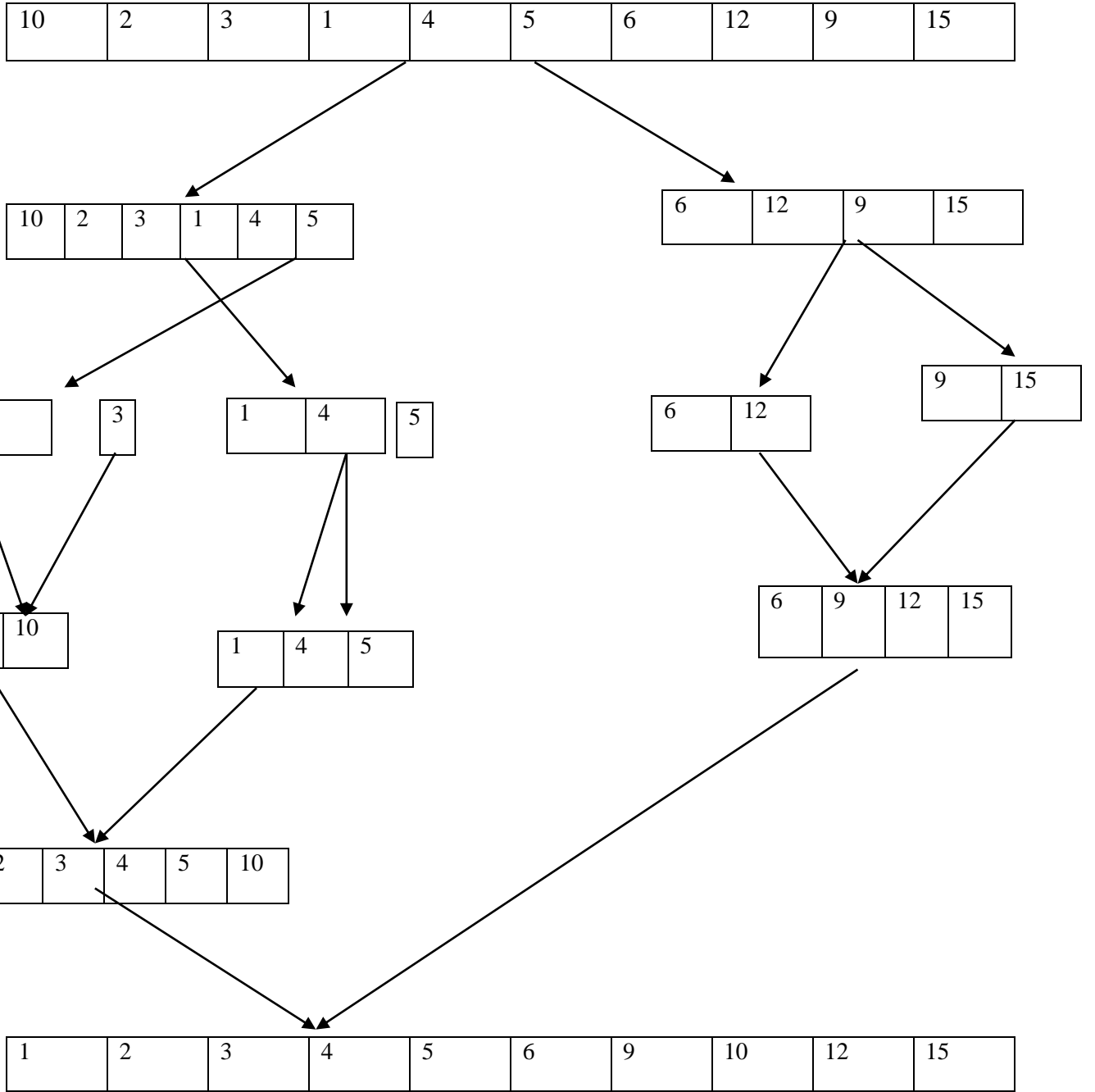
    if (l < r) //sol sağdan küçükse başa gider 0 indise
    {

        int m = l+(r-l)/2; //dizinin ortasını hesaplar 5.eleman 4.indisten ayırır m= 0(10-1)/2
        merge_sort(sureler, l, m);
        merge_sort(sureler, m+1, r);
        mergem(sureler, l, m, r); //merge fonksiyonuna sol orta sağ indisini yolluyorum
    }
}

void dizi_yazdir(int sureler[],int a) //son olarak dizimin yeni halini ekrana yazan fonksiyon
{
    int i=0;
    printf("\n");
    for(i=0;i<a;i++)
    {
        printf("%d ",sureler[i]); //dizimin yeni halini ekrana yazar.
    }
}

```

2 .soru için tablo yapımı:[Belgenin Başı](#)



Cevap3.c)

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

```
#include <stdlib.h>
```

```
#include <time.h>
```

```
#include <string.h>
```

```
void sifrelenmis (char *sifredizisi,int key,int n); //dinamik dizimizi ve anahtarımızı ve  
karakter uzunluğumuzu fonksiyonumuza aldık
```

```
int main(){
```

```
    int a;
```

```
    int n;
```

```
    int i;
```

```
    char *sifredizisi; //sifremizin tutulacağı dinamik dizi
```

```
    srand(time(NULL)); //zamana göre sifre oluşturma blogu değişsin diye yaptım
```

```
    printf("Kac karakter uzunlugunda sifre belirlenip sifrelensin=");
```

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

```
    char
```

```
harfler_sayilar[]={ 'a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r','s','t','u','v','w','x','y','z','A','B',  
, 'C','D','E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T','U','V','W','X','Y','Z','1','2','3','4','5','6','  
7','8','9','0'};
```

```
    //harfler ve sayilar dizisini random büyük harf küçük harf ve sayilar için oluşturdum
```

```
    sifredizisi=(char*)malloc(sizeof(char)*n); //dinamik dizi kaplayacağı alan n kadar  
karakter uzunluğunda bir alan kaplar
```

```
    char sifrelemek[n]; //sifrelemek istediğim dizim n tane elemandan oluşacak yani sifrenin  
karakter uzunluğu
```

```
    if(sifredizisi==NULL)
```

```
{
```

```
    printf("Hafizada alan yok");
```

```
}
```

```
else{
```

```

for(i=0;i<n;i++)
{
    a=rand()%60+1; //1 den başlayarak 60 arasında random bir sayıyı attım

    sifrelemek[i]=harfler_sayilar[a]; //bu random sayım harfler_sayilar dizimin
içindeki elemanlardan herhangi birini buldu ve onu msg sifrelemek istediğim dizimin elemanı
yaptım
}
}

sifredizisi=sifrelemek; //sifrelemek dizimi dinamik bellek yönetimi ile tanımladım
int key=3; //satir degeridir bunu istersek değiştirebiliriz

printf("Kullanilacak sifre: %s",sifredizisi);

sifrelenmis(sifredizisi,key,n);

free(sifredizisi); //hafızada tutulan yeri serbest bırakıyorum
return 0;
}

void sifrelenmis (char *sifredizisi,int key,int n)
{
    int uzunluk =n; //sifremizin boyunu uzunluk parametresinde tuttuk

    int i, j, k = -1, satir = 0, sutun = 0;

    char matrix[key][uzunluk]; //2 boyutlu bir dizi tanımladık key satırı sifrenin uzunluğu ise
sutunu belirtir

    for(i = 0; i < key; ++i) //satir ve sutunumuzu oluşturduk 3 satir mesajımız kaç boyutlu ise
o kadarda sutun var

        for(j = 0; j < uzunluk; ++j)

            matrix[i][j] = '\n';

```

```

for(i = 0; i < uzunluk; ++i){ //mesaj uzunluguna göre

    matrix[satir][sutun++] =*(sifredizisi+i); //her seferinde sutunu artırdık çünkü 0. indisin
1.koyduk ilk,sonra 1.indisin 2.koyduk ve dinamik dizimizin içerisinde bulunan karakterleri
atadık diğer dizimize

    if(satir == 0 || satir == key-1)

        k= k * (-1);

    satir = satir + k;

}

printf("\nSifrelenmis sifre: ");

for(i = 0; i < key; ++i)

    for(j = 0; j < uzunluk; ++j)

        if(matrix[i][j] != '\n')

            printf("%c", matrix[i][j]); //dizinin her elamanını ekrana yazdırdık

}

```

Şifreleme Algoritmasının Açıklaması:

Şifreleme algoritması olarak zigzag (Rail Fence Cipher) şifrelemeyi seçtim. Zigzag şifreleme anahtar yani satır sayısına göre bir satır oluşturup tablomuzun satırına rastgele belirlenmiş harflerden ve sayılardan oluşan karakterlerimizi yazar daha sonra belirlenen şifremizin karakter uzunluğu kadar sütun oluşturup bu sütuna harflerimizi koyar. Şifreleme işlemi için ilk satırdan şifremizin karakterlerini koymaya başlarız sonra diğer satırlara geçeriz. Her döngü dönmesinde satır ve sütun sayıları 1 artar ilk önce 1. satır 1. sütun sonra 2. satır 2. sütun ta ki anahtar uzunluğumuza geldiği zaman satirimiz -1 ile çarpılarak satır sayımız azalarak kaldığımız sütündan karakterlerimizi koymaya devam eder yani anahtarımız 3 dü 3 den 4 e artar ama satır sayımız 2. satır olur geldiğinde tekrar artma işlemi uygulanır anahtar sayımız kadar satır sayımız artar sütun sayımız işe kaldığı sütündan satırı farklı olarak devam eder. Karakter uzunluğumuz kadar sütuna şifremizi koyduktan sonra 1. satırdan okumaya başlanır satırda harf kalmayana kadar sütunun sonuna gelinir ve satır +1 artar tekrar sütunun sonuna gelinir tekrar satır +1 artar anahtar uzunluğumuza gelene kadar ve bu satırlardan elde edilen bilgileri sırasıyla ekrana yazar. Programda bu satır ve sütun sayımızı 0 dan başlatmamdaki sebep indis sayıları 0'dan başlamasıdır.

Örnek olarak dogan06 yı bu şifreleme algoritması ile şifreleyelim. Programda bu şifrenin şifrenleme işlemi olmayacaktır çünkü rastgele rakamlar ve harflerden oluşuyor program. dogan06 karakter uzunluğu 7 ve anahtarımız ise 3 dür 7x3 lük bir tablo oluşturucuz ,7 sütünlü 3 satırlı olacak.

d				n		
	o		a		0	
		g				6

Burada şifrelenmiş hali; dnoa0g6'dır.

[Belgenin Başı](#)

Oyun Yapımı

Oyunumda allegro kütüphanesini kullanarak bir klasik yem yiyen pacman oyunu yapıp sizlere açıklamak istedim öncelikle allegro kütüphanesini kullanmak için microsoft visual studio 2019'u tercih ettiğimi belirtirim oradan paketleri yükleyip bu oyunu yaptım. Oyun dev c++ ya da codeblock gibi uygulamalarda çalışmıyor paket yüklenmediği için oyunu microsoft visual studio ya da başka bir derleyicide denemenizi tavsiye ederim. Allegroyu yükleyemeyen arkadaşlar olursa doganay58068@gmail.com adresinden iletişime geçebilirler benimle.

```
#include <stdio.h>
#include <allegro5/allegro.h>
#include <allegro5/allegro_primitives.h>
int main() {
    srand(time(NULL)); //rastgele bir zamanda belirsin yem
    al_init_primitives_addon();
    int r = 10;
    float start = 0.5;
    float bitis = 5.25;
    ALLEGRO_DISPLAY* display = NULL;
    ALLEGRO_EVENT_QUEUE* queue = NULL;

    int xpos = 300, ypos = 300;

    int dotx = rand() % 550; //pencerinin 550 lik x konumuna kadar rastgele oluştur

    int doty = rand() % 350; //pencerinin 350 lik y konumuna kadar rastgele oluştur

    float rotation = 0.0;

    al_init();
    al_install_keyboard();
    al_install_mouse();

    display = al_create_display(600, 400); //ekran yarat 600 e 400 lük
    queue = al_create_event_queue(); //yaptığım her olayı havuza kaydet

    al_register_event_source(queue, al_get_display_event_source(display));
    al_register_event_source(queue, al_get_keyboard_event_source());
    al_register_event_source(queue, al_get_mouse_event_source());
    while (1)
    {
        ALLEGRO_EVENT ev;
        al_wait_for_event(queue, &ev);
```

```

al_clear_to_color(al_map_rgb(0, 0, 0)); //her yaratılan daireyi sil
al_flip_display();
if (ev.keyboard.keycode == ALLEGRO_KEY_RIGHT)
{
    xpos += 20; //x pozisyonunda 20 kadarlık hareket et 320 yap mesela
    rotation = 0.0;
}
else if (ev.keyboard.keycode == ALLEGRO_KEY_UP)
{
    ypos -= 10; //x pozisyonunda 20 kadarlık hareket et 320 yap mesela 300
    rotation = 4.71; //4.71 lik radyan dönüşü yapıyor
}
else if (ev.keyboard.keycode == ALLEGRO_KEY_DOWN)
{
    ypos += 10; //x pozisyonunda 20 kadarlık hareket et 320 yap mesela
    rotation = 1.57; // radyanlık bir dönüş yapar
}
else if (ev.keyboard.keycode == ALLEGRO_KEY_LEFT)
{
    xpos -= 20; //x pozisyonunda 20 kadarlık hareket et 320 yap mesela 300
    rotation = 3.14; //pi kadarlık bir dönüş yap
}
//tusa basmadan da daire yaratması için yaptık
//al_draw_filled_circle(xpos, ypos, r, al_map_rgb(0, 0, 255)); //daire ciz
al_draw_filled_pieslice(xpos, ypos, r, start+rotation, bitis, al_map_rgb(0, 0,
255)); //pasta dilimi fonksiyonu için gerekli şeyler
al_draw_filled_circle(dotx, doty, 4, al_map_rgb(255, 0,0 )); //daire ciz 0,0,255
al_flip_display(); //ekrana tamamını yansıt

if (abs(dotx - xpos) <= 10 && abs(doty - ypos) <= 10) //arasındaki farkı aldım
çünkü büyük daire ve küçük dairenin üzerinde ise bu yarıcap büyük dairenin yarıcapından
küçük olucak
{
    //abs fonksiyonunu kullanmamdaki sebep mutlak değer ifade olucak
    {
        dotx = rand() % 550;
        doty = rand() % 350;
        r += 3; //yemi yediğinde yarıcapı 5 artar
    }
}

}system("pause");
return 0;
}

```

Yazar Hakkında Kısacık Bilgi

DOĞAN AY (1998-)

Yazar Doğan AY aslen Sivaslı olup Ankara’da ikamet etmektedir .Türküzü Oğuzhan Anadolu ve Meslek lisesi Web Programcılığı ve Yazılım bölümü mezunudur .Şuanda Karamanoğlu Mehmetbey Üniversitesi bilgisayar mühendisliği okumaktadır.Bilgisayara ilgisi ilkokul 5.sınıfta başlamıştır.Daha sonraları lisede bu bölümü seçmesine sebep olmuştur.Bu ilgisi bilgisayar dillerini görünce daha çok artmaya başlamış ve üniversitede de bu bölümü seçmesine olanak sağlamıştır.

Sosyal Medya hesaplarım;

Linkedin: <https://www.linkedin.com/in/do%C4%9Fan-ay-bb1524192/>

FACEBOOK: <https://www.facebook.com/dogan.ay.98>

INSTAGRAM: <https://www.instagram.com/dogan.ay.98/?hl=tr>

GİTHUB: <https://github.com/DOGANAY06>

GMAIL: doganay58068@gmail.com

[Belgenin Başı](#)

Kaynakça

- <https://ckaynak.com/c-programlama-dilinde-operatorler-ve-ifadeler-1048>
 - <http://www.kaptankopekbaligi.com/2013/12/on-islemci-komutlar.html>
 - <https://web.cs.hacettepe.edu.tr/~maydos/Docs/c/fonksyon.pdf>
 - <http://alikeskin.org/?p=1034>
 - <http://web.firat.edu.tr/iserhatlioglu/BOLUM8.htm>
 - https://www.academia.edu/8561877/Ders_1
- GİTHUB KOD KAYNAKÇA: [https://github.com/beyzade06/C-code-Belgenin Başı](https://github.com/beyzade06/C-code-Belgenin-Bası)