

**T.C**  
**MANİSA CELAL BAYAR ÜNİVERSİTESİ**

**PERSONEL MAAŞ HESAPLAMA YAZILIMI**

**İSMET KIZGIN**  
**KADİR CAN KARADEMİR**

**DÖNEM PROJESİ**  
**ALGORİTMA VE PROGRAMLAMA I**

**PROJE DANIŞMANI**  
**Dr. Öğretim Üyesi Aytuğ ONAN**

**ÖDEV 1**

**MANİSA , 2018**

# ALGORİTMA VE PROGRAMLAMA

## PERSONEL MAAŞ HESAPLAMA YAZILIMI

<b>ANALİZ</b>	<b>3</b>
PROBLEM TANIMI	3
Çalışanlardan İstenilecek Bilgiler	3
Çalışan Bireye Özel Çıktı Ekranında Bulunacak Olan Veriler	4
İstatistiki Çıktı Ekranında Bulunacak Olan Veriler	4
<b>TASARIM</b>	<b>6</b>
ÇÖZÜM ALGORİTMASI	6
Çalışanlardan İstenilecek Bilgiler	6
Çalışan Bireye Özel Çıktı Ekranında Şunlar Bulunmalı	7
<b>PROGRAMCI KATALOĞU</b>	<b>10</b>
PROJE İÇİN ZAMANLAMA PLANI	10
PROJE KAYNAK ÇIKTI KODU	10
KÜTÜPHANELER	10
GLOBAL DEĞİŞKENLER	11
En Yüksek Maaşlı Bireylerin Bilgilerinin Tutulduğu Değişkenler	11
Banknot Değişkeni	11
FONKSİYONLAR	12
Engel Derecesi Fonksiyonu	12
Engelli İndirimi Fonksiyonu	12
Eş Ödeneği Fonksiyonu	13
Çocuk Ödeneği Fonksiyonu	13
Aylık Brüt Ücret Fonksiyonu	14
Gelir Vergisi Yüzdesi Fonksiyonu	14
Gelir Vergisi Fonksiyonu	14
Net Ücret Fonksiyonu	15
En Yüksek Çalışanların Bilgileri Alma Fonksiyonu	15
Banknot Sayısı Bulma Fonksiyonu	16
Banknot Sayısı Yazdırma Fonksiyonu	16
MAIN FONKSİYONU	17
Main Fonksiyonu Değişkenleri	17
Do While Döngüsü İçinde Bulunan Kodlar	17
Değişkenler	17
Çalışan Bilgilerini Alma	17
Girilen Çalışana Özel Çıktı Ekranı	21

# ALGORİTMA VE PROGRAMLAMA

## PERSONEL MAAŞ HESAPLAMA YAZILIMI

İstatistik Bilgilerin Hesaplanması	22
Çalışan Sorgusu ( Devamlılık )	23
İstatistik Bilgilerinin Çıktısı	23
GitHub KAYNAĞI	24
<b>KULLANICI KATALOĞU</b>	<b>25</b>
Çalışan Bilgilerinin Girilmesi	25
Çalışan Bilgilerinin Ekrana Yazılması	25
Farklı Çalışan Sorgusu	26
İstatistik Bilgilerinin Ekrana Yazılması	26

# ALGORİTMA VE PROGRAMLAMA

## PERSONEL MAAŞ HESAPLAMA YAZILIMI

### ANALİZ

#### PROBLEM TANIMI

Bir firmada bulunan çalışanların otomatik maaş hesaplamasını ve bazı istatistiki verilerini elde edebilmek için çalışanların kişisel bilgileri gerekmektedir. Hesaplama gelir vergisi kesintisi dikkate alınarak net maaş hesaplanmalıdır. Maaş ödemesinde ise en az sayıda banknot ve madeni para kullanılmalıdır.

#### Çalışanlardan İstenilecek Bilgiler

1. TC Kimlik numarası
2. Ad, soyad
3. Aylık brüt ücreti (aylık brüt asgari ücret ya da daha büyük)
4. Medeni durumu
  - Evli ise eşinin çalışma durumu
5. Bakmakla yükümlü olduğu çocuk sayısı
  - Bakmakla yükümlü olduğu çocuk varsa, yaşı 6'dan büyük olanların sayısı
6. Engellilik durumu
  - Engelliyse, engellilik oranı yüzlük sistem üzerinden belirtilmelidir.

Bu bilgiler ışığında gerekli ek ödenekler ve vergi hesaplandıktan sonra net ücret tespit edilip, firmanın istediği şekilde bilgileri girilen çalışana özel bir çıktı ekranı verilmelidir.

# ALGORİTMA VE PROGRAMLAMA

## PERSONEL MAAŞ HESAPLAMA YAZILIMI

### Çalışan Bireye Özel Çıktı Ekranında Bulunacak Olan Veriler

1. TC Kimlik numarası
2. Ad, Soyad
3. Aylık brüt ücreti
4. Eş için aile yardım ödeneği
5. Çocuk için aile yardım ödeneği
6. Gelir vergisi kesintisi
7. Aylık net ücret
8. Aylık Net Ücretin En Az banknot ile Ödenmesi İçin Gerekli Banknot Adetleri

Bu işlemler yapıldıktan sonra başka bir çalışan olup olmadığı kontrol edilmelidir. Eğer Başka bir çalışan var ise yukarıdaki işlemler tekrar uygulanmalıdır.

Tüm çalışanların bilgileri girildikten sonra çalışanlar ile ilgili bazı istatistiki veriler hesaplanıp ekranda gösterilmelidir.

### İstatistiki Çıktı Ekranında Bulunacak Olan Veriler

1. Tüm çalışanların net ücretlerinin toplamı
2. Devlete ayrılan toplam gelir vergisi ücretleri
3. Tüm çalışanların brüt ücretlerinin ortalaması
4. Tüm çalışanlarının net ücretlerinin ortalaması
5. Aylık 2000 ₺ altında maaş alan çalışan sayısı
6. Evli olan çalışanların yüzdesi
7. Bekar olan çalışanların yüzdesi
8. Evli olan çalışanların içinde eşlerinin çalışma yüzdesi
9. Sadece bakmakla yükümlü olduğu çocuğu olanlar dikkate alınarak, çalışanların bakmakla yükümlü oldukları çocuk sayısının ortalaması
10. Bakmakla yükümlü olduğu çocuk sayısı 3'ten fazla olan çalışanların sayısı
11. Engelli çalışan sayısı

# ALGORİTMA VE PROGRAMLAMA

## PERSONEL MAAŞ HESAPLAMA YAZILIMI

12. Tüm çalışanlar içinde engelli çalışan yüzdesi
13. Aylık net ücretin en az banknot ile ödenmesi için banknot adetleri
14. Gelir vergisi tablosu (vergi yüzdesi, çalışan sayısı ve tüm çalışanları baz alan vergi yüzdesi )
15. Aylık brüt ücreti en fazla olan çalışan bilgileri
  - TC Kimlik numarası
  - Ad, Soyad
  - Aylık brüt ücret
  - Gelir vergisi
  - Aylık net ücret
16. Aylık net ücreti en fazla olan çalışan bilgileri
  - TC Kimlik numarası
  - Ad, Soyad
  - Aylık brüt ücret
  - Gelir vergisi
  - Aylık net ücret

# ALGORİTMA VE PROGRAMLAMA

## PERSONEL MAAŞ HESAPLAMA YAZILIMI

### TASARIM

#### ÇÖZÜM ALGORİTMASI

**Adım 1** Çalışan bilgilerinin alınması :

##### Çalışanlardan İstenilecek Bilgiler

1. TC Kimlik numarası
2. Ad, soyad
3. Aylık brüt ücreti (aylık brüt asgari ücret ya da daha büyük)
4. Medeni durumu
  - Evli ise eşinin çalışma durumu
5. Bakmakla yükümlü olduğu çocuk sayısı
  - Bakmakla yükümlü olduğu çocuk varsa, yaşı 6'dan büyük olanların sayısı
6. Engellilik durumu
  - Engelliyse, engellilik oranı yüzlük sistem üzerinden belirtilmelidir.

YUKARIDAKİ BİLGİLER BU SIRALAMA İLE İSTENMEKTEDİR.

**Adım 2** Girilen çalışan bilgileri alınırken istenilen kriterlere uygunluğu durumunun kontrol edilmesi:

Girilen bilgiler için şu durumlar kontrol edilmelidir:

1. TC Kimlik numarasının 11 haneli sayı olmasının kontrolü
2. Aylık brüt ücretin asgari ücret ya da asgari ücretten büyük olmasının kontrolü
3. Medeni durumu girildiği zaman e/E/b/B bu karakterlerden birinin girilip girilmediğinin kontrolü
  - Evliyse eşinin çalışma durumunun kontrolü için e/E/h/H bu karakterlerden birinin girilip girilmediğinin kontrolü
4. Bakmakla yükümlü olduğu çocuk sayısının doğal sayı olma durumunun kontrolü (cocuk\_sayisi>=0)
  - 6 yaşından büyük çocuk sayısı girildiği takdirde çocuk sayısından büyük

# ALGORİTMA VE PROGRAMLAMA

## PERSONEL MAAŞ HESAPLAMA YAZILIMI

- olmama ve doğal sayı olma durumunun kontrolü  
(  $\text{buyuk\_cocuk} \leq \text{cocuk\_sayisi} \ \&\& \ \text{buyuk\_cocul} \geq 0$  )
5. Engelli durumu girildiği zaman e/E/h/H bu karakterlerden birinin girilip girilmediğinin kontrolü
    - o Engellilik oranı girildiği zaman 100'lük sistem üzerinden girilip girilmediğinin kontrolü

### Adım 3 Temel fonksiyonların oluşturulması:

1. Eş için aile yardımı ödeneğinin hesaplandığı fonksiyonun oluşturulması
2. Çocuk için aile yardım ödeneğinin hesaplandığı fonksiyonun oluşturulması
3. Aylık toplam brüt ücretin hesaplandığı fonksiyonun oluşturulması
4. Engel derecesinin hesaplandığı fonksiyonun oluşturulması
5. Engelli vergi indiriminin hesaplandığı fonksiyonun oluşturulması
6. Gelir vergisinin hesaplandığı fonksiyonun oluşturulması
7. Aylık net ücretin hesaplandığı fonksiyonun oluşturulması
8. Tedavüldeki banknot sayısı ve türlerini hesaplayan fonksiyon oluşturulması  
(200₺, 100₺, 50₺, 20₺, 10₺, 5₺, 1₺, 50 kr, 25 kr, 10 kr, 5 kr, 1 kr)

### Adım 4 Kişiye özel çıktı ekranının hazırlanması:

Çalışan bilgileri fonksiyonlara gönderilerek gerekli bilgiler elde edildikten sonra aşağıdaki sırayla ekrana çıktı olarak gösterilecektir.

#### Çalışan Bireye Özel Çıktı Ekranında Şunlar Bulunmalı

1. TC Kimlik numarası
2. Ad, Soyad
3. Aylık brüt ücreti
4. Eş için aile yardım ödeneği
5. Çocuk için aile yardım ödeneği
6. Gelir vergisi kesintisi
7. Aylık net ücret
8. Aylık Net Ücretin En Az banknot ile Ödenmesi İçin Gerekli Banknot Adetler



# ALGORİTMA VE PROGRAMLAMA

## PERSONEL MAAŞ HESAPLAMA YAZILIMI

### **Adım 5** İstatistik fonksiyonlarının ve değişkenlerinin oluşturulması:

En yüksek, brüt ücrete ve net ücrete sahip olan kişinin belirlenmesi için gereken fonksiyonun oluşturulması. Bilgilerin tutulması için istatistik çıktısında istenilen bilgilerin doğrultusunda değişken tanımlanması.

### **Adım 6** İstatistiki verilerin oluşturulması ya da güncellenmesi:

Çalışan için özel olarak hazırlanan ekran çıktısından sonra bu bilgiler doğrultusunda istatistik için değişkenler içinde tutulan bilgilerin güncellenmesi.

### **Adım 7** Farklı çalışan durum sorgusu:

Girilmesi gereken bir çalışan var ise ADIM 1, ADIM 2, ADIM 4, ADIM 6 adımlarının tekrarlanması için gerekli döngünün oluşturulması ve bu sorgu sırasında girilmesi istenilen e/E/h/H karakterleri dışında farklı bir karakter girilip girilmediğine dair kontrollerin yapılması.

### **Adım 8** İstatistik verilerinin ekran çıktısının oluşturulması:

Gerekli fonksiyonlarda hesaplamalar yapılarak ve istatistik değişkenlerinden veriler çekilerek aşağıdaki bilgiler sırayla yazdırılır:

1. Tüm çalışanların net ücretlerinin toplamı
2. Devlete ayrılan toplam gelir vergisi ücretleri
3. Tüm çalışanların brüt ücretlerinin ortalaması
4. Tüm çalışanlarının net ücretlerinin ortalaması
5. Aylık 2000 ₺ altında maaş alan çalışan sayısı
6. Evli olan çalışanların yüzdesi
7. Bekar olan çalışanların yüzdesi
8. Evli olan çalışanların içinde eşlerinin çalışma yüzdesi
9. Sadece bakmakla yükümlü olduğu çocuğu olanlar dikkate alınarak, çalışanların bakmakla yükümlü oldukları çocuk sayısının ortalaması
10. Bakmakla yükümlü olduğu çocuk sayısı 3'ten fazla olan çalışanların sayısı
11. Engelli çalışan sayısı

# ALGORİTMA VE PROGRAMLAMA

## PERSONEL MAAŞ HESAPLAMA YAZILIMI

12. Tüm çalışanlar içinde engelli çalışan yüzdesi
13. Aylık net ücretin en az banknot ile ödenmesi için banknot adetleri
14. Gelir vergisi tablosu (vergi yüzdesi, çalışan sayısı ve tüm çalışanları baz alan vergi yüzdesi )
15. Aylık brüt ücreti en fazla olan çalışan vergileri
  - TC Kimlik numarası
  - Ad, Soyad
  - Aylık brüt ücret
  - Gelir vergisi
  - Aylık net ücret
16. Aylık net ücreti en fazla olan çalışan bilgileri
  - TC Kimlik numarası
  - Ad, Soyad
  - Aylık brüt ücret
  - Gelir vergisi
  - Aylık net ücret

# ALGORİTMA VE PROGRAMLAMA

## PERSONEL MAAŞ HESAPLAMA YAZILIMI

### PROGRAMCI KATALOĞU

### PROJE İÇİN ZAMANLAMA PLANI

PROJE İÇİN AYRILAN TOPLAM SÜRE: 25.11.2018 - 23.12.2018

SÜREÇ	SÜRE
ANALİZ	20 DAKİKA
TASARIM	20 DAKİKA
GERÇEKLEŞTİRİM	2 SAAT
TEST	20 DAKİKA
RAPORLAMA	24 SAAT

### PROJE KAYNAK ÇIKTI KODU

### KÜTÜPHANELER

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

Yazılan kodların çalışması için gerekli kütüphanelerden olan Giriş/Çıkış kütüphanesi stdio.h eklendi ve basit matematiksel işlemler için ise stdlib.h kütüphanesi eklendi.

# ALGORİTMA VE PROGRAMLAMA

## PERSONEL MAAŞ HESAPLAMA YAZILIMI

### GLOBAL DEĞİŞKENLER

#### En Yüksek Maaşlı Bireylerin Bilgilerinin Tutulduğu Değişkenler

```
float i_brut_ucret[2]={0,0}, i_gelir_vergisi[2]={0,0}, i_net_ucret[2]={0,0};  
char b_ad[30], n_ad[30];  
long int i_tc[2];
```

Bizden istenen en yüksek net ücretli çalışan ve en yüksek brüt ücretli çalışanın bilgilerini bu değişkenler içinde tutuyoruz. Her ikisi içinde istenilen bilgiler aynı olduğundan kaynaklı diziler ile işlemler yapılmıştır. Dizilerde 0. indiste bulunan bilgiler brüt ücreti en fazla olan çalışan, 1. indiste bulunanlar ise net ücreti en yüksek olan çalışana aittir.

#### Banknot Değişkeni

```
float banknot[12][3]={200,0,0,100,0,0,50,0,0,20,0,0,10,0,0,5,0,0,1,0,0,0.50,0,0,0.25,0,0,0.10,0,0,0.05,0,0,0.01,0,0};
```

Banknot bilgilerini bizden 2 farklı şekilde tutmamız istenmiştir. Bunlardan biri çalışanların bilgileri girilirken net ücretin en az banknot ile hesaplanması ve ekrana basılması ikincisi ise tüm çalışanların net ücretlerinin en az banknot ile ödenmesi. Bu işlemleri yaparken her çalışan için istenen banknot sayıları hafızada tutmak gerektiğinde dolayı kaynaklı bir dizi oluşturuyoruz bu dizinin 0. indisinde banknotun büyüklüğü 1. indisinde ise tüm çalışanların net ücretlerinin en az banknot ile ödenmesi için gerekli veriler, 2. indiste ise en son fonksiyona net ücreti gönderilen çalışanın net ücretinin en az banknot ile ödenmesi için gerekli veriler tutulmaktadır.

# ALGORİTMA VE PROGRAMLAMA

## PERSONEL MAAŞ HESAPLAMA YAZILIMI

### FONKSİYONLAR

#### Engel Derecesi Fonksiyonu

```
int engel_derecesi(int engel_orani)
{
    if(engel_orani>=80)
        return 1;
    else if(engel_orani>=60 && engel_orani<80)
        return 2;
    else if(engel_orani>=40 && engel_orani<60)
        return 3;
    else if(engel_orani<40)
        return 0;
}
```

1 ile 100 arasında girilecek olan engel oranı belirlendikten sonra engel derecesini belirlemek için gerekli fonksiyonumuz. Tam sayılardan oluştuğundan kaynaklı integer değişken kullanıyoruz.

#### Engelli İndirimi Fonksiyonu

```
float engelli_indirimi(int engel_yuzdesi)
{
    if(engel_yuzdesi >= 80 && engel_yuzdesi <=100)
        return 900;
    else if(engel_yuzdesi >= 60 && engel_yuzdesi <80)
        return 470;
    else if(engel_yuzdesi >= 40 && engel_yuzdesi <60)
        return 210;
    else
        return 0;
}
```

# ALGORİTMA VE PROGRAMLAMA

## PERSONEL MAAŞ HESAPLAMA YAZILIMI

Engelli indirim fonksiyonuna gönderilen engel oranı ile çalışanın maaşının ne kadarlık kısmının vergiden muaf tutulduğu hesaplanıyor.

### Eş Ödeneği Fonksiyonu

```
float es_odenegi(char medeni_durumu, char es_durumu)
{
    if(medeni_durumu=='E' || medeni_durumu=='e')
    {
        if(es_durumu=='H' || es_durumu=='h')
            return 220;
        else
            return 0;
    }
    else
    {
        return 0;
    }
}
```

Bu fonksiyonda medeni durum ve eşin çalışma durumu bilgisi girildikten sonra aileye yapılacak olan yardımı sorguluyoruz.

### Çocuk Ödeneği Fonksiyonu

```
float cocuk_odenegi(int cocuk_sayisi, int buyuk_cocuk_sayisi)
{
    return ((cocuk_sayisi - buyuk_cocuk_sayisi) * 25) + (buyuk_cocuk_sayisi * 45);
}
```

Çocuk sayısı ve 6 yaşından büyük çocuk sayısı fonksiyona gönderiliyor ve fonksiyonda 6 yaşından küçük çocuk sayısını bulmak için çocuk sayısından 6 yaşından büyük çocuk sayısını çıkarıyoruz daha sonra 25 her bir çocuk için ödenek veriyoruz. 6 yaşından büyük çocuklar içinde her biri için 45 ödenek verilir toplanıyor.

# ALGORİTMA VE PROGRAMLAMA

## PERSONEL MAAŞ HESAPLAMA YAZILIMI

### Aylık Brüt Ücret Fonksiyonu

```
float aylık_brut_ucret(float cocuk, float es, float brut)
{
    return (cocuk + es + brut);
}
```

Ek ödenek fonksiyonlarından gelen bilgileri hesapladıktan sonra brüt ücret fonksiyonuna gönderip burada aylık brüt ücreti hesaplıyoruz.

### Gelir Vergisi Yüzdesi Fonksiyonu

```
float gelir_vergisi_yuzdesi(float brut_ucret, float engel_yuzdesi)
{
    brut_ucret -= engelli_indirimi(engel_yuzdesi);

    if(brut_ucret < 2000)
        return 0.15;
    else if(brut_ucret >= 2000 & brut_ucret < 5000)
        return 0.20;
    else if(brut_ucret >= 5000 & brut_ucret < 10000)
        return 0.27;
    else if(brut_ucret >= 10000)
        return 0.35;
}
```

Brüt ücret ve engel oranı gönderildikten sonra engelli indirim fonksiyonundan yararlanılarak vergiye tabi tutulan maaş tutarını tespit edip, maaşın vergi oranını tespit ediyoruz.

### Gelir Vergisi Fonksiyonu

```
float gelir_vergisi(float brut_ucret, float engel_yuzdesi)
{
    return (float) ((brut_ucret-engelli_indirimi(engel_yuzdesi)) * gelir_vergisi_yuzdesi(brut_ucret,engel_yuzdesi));
}
```

# ALGORİTMA VE PROGRAMLAMA

## PERSONEL MAAŞ HESAPLAMA YAZILIMI

Brüt ücret ve engel oranını parametre ile istedikten sonra fonksiyon içinde matematiksel işlem yaparak vergi ücretini tespit ediyoruz. Brüt ücretten engelli indirimini çıkarıp vergi oranı ile çarpıyoruz.

### Net Ücret Fonksiyonu

```
float net_ucret(float brut, float vergi)
{
    return (brut - vergi);
}
```

Net ücret fonksiyonuna parametre yardımı ile gönderilen brüt ücret ve vergi ücreti birbirinden çıkarılıp net ücret tespit edilmiştir.

### En Yüksek Çalışanların Bilgileri Alma Fonksiyonu

```
void en_yuksek(float brut_ucret, float net_ucret, float vergi, long int tc, char ad[])
{
    if(brut_ucret >= i_brut_ucret[0])
    {
        i_brut_ucret[0]=brut_ucret;
        i_gelir_vergisi[0]=vergi;
        i_net_ucret[0]=net_ucret;
        i_tc[0]=tc;
        for(int i=0;i<30;i++)
            b_ad[i]=ad[i];
    }

    if(net_ucret >= i_net_ucret[1])
    {
        i_brut_ucret[1]=brut_ucret;
        i_gelir_vergisi[1]=vergi;
        i_net_ucret[1]=net_ucret;
        i_tc[1]=tc;
        for(int i=0;i<30;i++)
            n_ad[i]=ad[i];
    }
}
```

En yüksek fonksiyonunda yapmış olduğumuz iki sorgu var ilki brüt ücretin büyüklüğünü sorguluyor en son girilen çalışan ile elde tutulan çalışanın brüt ücretleri birbiriyle



# ALGORİTMA VE PROGRAMLAMA

## PERSONEL MAAŞ HESAPLAMA YAZILIMI

kıyaslanıyor ve en büyük brüt ücrete sahip olan çalışanın bilgileri dizilerin 0. indislerine kayıt ediliyor. İkinci sorgumuzda yapmış olduğumuz işlem ise net ücret ile ilgilidir. Brüt ücretteki gibi aynı yolları izleyip en büyük net ücreti ele alıp dizilerin 1. indislerine kayıt ediyor bilgileri.

### Banknot Sayısı Bulma Fonksiyonu

```
void banknot_sayisi(float para)
{
    for(int i=0;i<12;i++)
    {
        banknot[i][2]=0;
        while(1)
        {
            if(para<banknot[i][0])
                break;
            banknot[i][1]++;
            banknot[i][2]++;
            para-=banknot[i][0];
        }
    }
}
```

Net ücreti parametre ile alıyoruz daha sonra fonksiyonda daha önceden tanımlamış olduğumuz diziye gerekli bilgileri atıyoruz. Banknot indislerine göre 0. indiste bulunan banknot tutarını sürekli net ücretten çıkarıyoruz 0 olana kadar. İşlem yapılan banknotun 1. ve 2. indislerinde ki değerleri birer artırıyoruz.

# ALGORİTMA VE PROGRAMLAMA

## PERSONEL MAAŞ HESAPLAMA YAZILIMI

### Banknot Sayısı Yazdırma Fonksiyonu

```
void banknot_sayisi_yazdir(int cikti)
{
    printf("\n**** Aylık Net Ücretin En Az Banknot ile Ödenmesi İçin Gerekli Banknot Adetleri ****\n\n");
    for(int i=0; i<12; i++)
    {
        if(banknot[i][cikti]>0)
        {
            if(i<7)
                printf("%.0f lira - %.0f Adet\n",banknot[i][0],banknot[i][cikti]);
            else
                printf("%.0f kurus - %.0f Adet\n",(banknot[i][0]*100),banknot[i][cikti]);
        }
    }
}
```

Banknot sayılarını tuttuğumuz dizi içinde bulunan verilerimizi ekrana basıyoruz. Parametre sayesinde almış olduğumuz indis ile genel banknot sayısı yada en son bilgileri girilen çalışana ait banknot sayıları yazılmaktadır.

## MAIN FONKSİYONU

### Main Fonksiyonu Değişkenleri

```
Float t_net_ucret=0, t_vergi=0, t_brut=0, vergi_yuzde;
int ikibin_alti=0, calisan_sayisi=0,evli=0,es_calisma=0,cocuklu_calisan=0,t_cocuk_sayisi=0,cocuk_sayisi_uc=0,engelli_calisan=0,vergi_yuzdesi[4][2]={15,0, 20,0, 27,0, 35,0};
```

Main fonksiyonu içinde istatistik bilgilerinin hesaplanması için gerekli olan değişkenlerimiz.

### Do While Döngüsü İçinde Bulunan Kodlar

Sonsuz bir do while döngüsü içine eklenecek olan kodlar.

### *Değişkenler*

```
int engel_orani, cocuk_sayisi, buyuk_cocuk_sayisi=0;
float brut_ucret, toplam_brut_ucret, net;
char medeni_durumu, es_durumu, engel_durumu, ad[30], devam;
long int tc;
```

Bilgileri girilen çalışanın bilgilerini almak için oluşturmuş olduğumuz değişkenler.

# ALGORİTMA VE PROGRAMLAMA

## PERSONEL MAAŞ HESAPLAMA YAZILIMI

*Çalışan Bilgilerini Alma*

```
printf("TC Kimlik Numaranızı Giriniz: ");  
scanf("%ld",&tc);
```

Çalışanın TC kimlik numarasını almak için gerekli çıktı ekranını ve girilen bilgiyi alıyoruz.

```
printf("Ad Soyad Bilgisini Giriniz(Max 30 karakter): ");  
getchar();  
fgets(ad, sizeof ad, stdin);
```

Çalışanın ad ve soyad bilgilerini girmesini gerekli çıktı ekranı ile belirtip girilen bilgileri alıyoruz.

```
while(1)  
{  
    printf("Brut Ucreti Giriniz (Aylık askeri ucret ya da daha büyük): ");  
    scanf("%f",&brut_ucret);  
  
    if(brut_ucret >= 1777.50)  
        break;  
}
```

Kullanıcıdan brüt ücret girilmesi istenmektedir. Girilen brüt ücret if sorgu ile kontrol edilip aylık brüt asgari ücretten yüksek olup olmadığını sorguluyor asgari ücretin altında bir ücret girilmiş ise tekrardan brüt ücretin girilmesi istenmektedir.

```
while(1)  
{  
    printf("Medeni Durumunu Giriniz 'Evli/Bekar (e/E/b/B karakterleri)': ");  
    scanf("%s",&medeni_durumu);  
  
    if(medeni_durumu == 'e' || medeni_durumu == 'E' || medeni_durumu == 'b' || medeni_durumu == 'B')  
        break;  
}
```

Kullanıcının medeni durumunun girilmesi istenmektedir. Girilen medeni durumda istenilen şekillerde belirtilmediği durumda tekrardan giriş yapılması istenmektedir.

# ALGORİTMA VE PROGRAMLAMA

## PERSONEL MAAŞ HESAPLAMA YAZILIMI

```
if(medeni_durumu == 'E' || medeni_durumu == 'e')
{
    while(1)
    {
        printf("Evli İse Esinin Çalışıp Çalışmadığı 'Evet/Hayır (e/E/h/H karakterleri)': ");
        scanf("%s",&es_durumu);

        if(es_durumu == 'e' || es_durumu == 'E' || es_durumu == 'h' || es_durumu == 'H')
            break;
    }
}
```

Medeni durum bilgisinin girilmesi istenmektedir. Medeni durum istenilen şekillerde belirtilmediği durumda tekrardan giriş yapılması istenmektedir.

```
while(1)
{
    printf("Bakmakla Yükümlü Olduğu Çocuk sayısı 'tamsayı (0 ya da daha büyük)': ");
    scanf("%d",&cocuk_sayisi);

    if(cocuk_sayisi >= 0)
        break;
}
```

Bakma yükümlü olduğu çocuk sayısının girilmesi istenmektedir. Çocuk sayısı istenilen şekillerde belirtilmediği durumda tekrardan giriş yapılması istenmektedir.

```
if(cocuk_sayisi>0)
{
    while(1)
    {
        printf("6 Yasından Büyük Çocuk Sayısını Giriniz: ");
        scanf("%d",&buyuk_cocuk_sayisi);

        if(buyuk_cocuk_sayisi <= cocuk_sayisi && buyuk_cocuk_sayisi >=0)
            break;
    }
}
```

Bakmakla yükümlü çocuğu olan çalışanların 6 yaşından büyük çocuk sayısı istenmektedir. Negatif bir değer veya toplam çocuk sayısından fazla bir değer girilirse tekrardan giriş yapılması istenmektedir.

# ALGORİTMA VE PROGRAMLAMA

## PERSONEL MAAŞ HESAPLAMA YAZILIMI

```
while(1)
{
    printf("Engelli Misiniz? Evet/Hayir (e/E/h/H karakterleri): ");
    scanf("%s", &engel_durumu);

    if(engel_durumu == 'e' || engel_durumu == 'E' || engel_durumu == 'h' || engel_durumu == 'H')
        break;
}
```

Çalışanın bir engellinin olup olmadığı istenmektedir. Engel durumu istenilen şekillerde belirtilmediği durumda tekrardan giriş yapılması istenmektedir.

```
if(engel_durumu == 'E' || engel_durumu == 'e')
{
    while(1)
    {
        printf("Engel Oranini Giriniz 'tamsayı (1 ya da daha büyük ve 100 ya da daha küçük)': ");
        scanf("%d",&engel_orani);

        if(engel_orani >=0 && engel_orani <=100)
            break;
    }
}
```

Engel durumu olan çalışanlardan engel durumlarının oranını 1 ile 100 arasında girmesi istenmektedir. Engel oranı istenilen şekillerde belirtilmediği durumda tekrardan giriş yapılması istenmektedir.

```
brut_ucret += es_odenegi(medeni_durumu,es_durumu) + cocuk_odenegi(cocuk_sayisi, buyuk_cocuk_sayisi);
```

Aylık toplam brüt ücretin hesaplanmasını gerekli fonksiyonlara verileri gönderip hesaplamasını yapıyoruz.

```
net = net_ucret(brut_ucret,gelir_vergisi(brut_ucret,engel_orani));
```

Aylık net ücret hesaplaması için gerekli fonksiyona veriler gönderilip hesaplama yapıyoruz.

# ALGORİTMA VE PROGRAMLAMA

## PERSONEL MAAŞ HESAPLAMA YAZILIMI

*Girilen Çalışana Özel Çıktı Ekranı*

```
printf("\n##### Kaydi Gerceklesen Calisan #####\n\n");
printf("TC Kimlik Numarasi: %ld\n",tc);
printf("Ad Soyad: %s",ad);
printf("Aylık Brut Ucreti(TL): %.2f\n",brut_ucret);
printf("Es İçin Aile Yardım Odenegi(TL): %.2f\n",es_odenegi(medeni_durumu,es_durumu));
printf("Cocuk İçin Aile Yardım Odenegi(TL): %.2f\n",cocuk_odenegi(cocuk_sayisi, buyuk_cocuk_sayisi));
printf("Gelir Vergisi Kesintisi(TL): %.2f\n",gelir_vergisi(brut_ucret,engel_orani));

if(engel_durumu == 'E' || engel_durumu == 'e')
{
    if(engel_derecesi(engel_orani))
        printf("Engel Derecesi(1/2/3): %d\n",engel_derecesi(engel_orani));
    else
        printf("Engel Derecesi(1/2/3): %d Oranında Engelli\n",engel_orani);
}

printf("Aylık Net Ucret(TL): %.2f\n",net);

banknot_sayisi(net);
banknot_sayisi_yazdir(2);
```

Çalışan bilgileri alındıktan sonra değişkenlerden alınan bilgiler ve fonksiyona gönderilen veriler doğrultusunda gelen bilgiler ile oluşturulan çalışana özel çıktı ekranı.



# ALGORİTMA VE PROGRAMLAMA

## PERSONEL MAAŞ HESAPLAMA YAZILIMI

*İstatistik Bilgilerin Hesaplanması*

```
calisan_sayisi++;
t_net_ucret += net;
t_brut += brut_ucret;
t_vergi += gelir_vergisi(brut_ucret,engel_orani);
if(net < 2000)
    ikibin_alti++;
if(medeni_durumu == 'e' || medeni_durumu == 'E')
{
    evli++;
    if(es_durumu == 'e' || es_durumu == 'E')
        es_calisma++;
}
if(cocuk_sayisi > 0)
{
    cocuklu_calisan++;
    t_cocuk_sayisi += cocuk_sayisi;
    if(cocuk_sayisi > 3)
        cocuk_sayisi_uc++;
}
if(engel_durumu == 'E' || engel_durumu == 'e')
    engelli_calisan++;
vergi_yuzde=gelir_vergisi_yuzdesi(brut_ucret,engel_orani);
if(vergi_yuzde == (float) 0.15)
    vergi_yuzdesi[0][1]++;
else if(vergi_yuzde == (float) 0.20)
    vergi_yuzdesi[1][1]++;
else if(vergi_yuzde == (float) 0.27)
    vergi_yuzdesi[2][1]++;
else if(vergi_yuzde == (float) 0.35)
    vergi_yuzdesi[3][1]++;
en_yuksek(brut_ucret,net,gelir_vergisi(brut_ucret,engel_orani),tc,ad);
```

İstatistik değişkenlerine ve fonksiyonlarına aktarılması gereken bilgilerin aktarılması ve istatistik bilgilerinin oluşturulması. Burada oluşturmuş olduğumuz değişkenleri bazılarını sayaç mantığı ile sorgu neticesinde artırılması ile tespit edilmesini sağlıyoruz. Bazı bilgilerimizi ise fonksiyonlara göndererek hesaplanması ve tespit edilmesini sağlıyoruz.

# ALGORİTMA VE PROGRAMLAMA

## PERSONEL MAAŞ HESAPLAMA YAZILIMI

Çalışan Sorgusu ( Devamlılık )

```
while(1)
{
    printf("\nBaska Calisan Var Mi ? 'Evet/Hayir (e,E,h,H karakteri)': ");
    scanf("%s",&devam);

    if(devam == 'e' || devam == 'E' || devam == 'h' || devam == 'H')
        break;
}

if(devam == 'h' || devam == 'H')
    break;
```

İstatistik bilgileri hesaplandıktan sonra girilen çalışan veya çalışanlar dışında çalışan olup olmadığını istemektedir. Farklı çalışan olması durumunda do while döngüsü içinde olan işlemlerin hepsini tekrarlaması sağlanmaktadır. Çalışanların bitmesi durumunda genel istatistik bilgilerinin ekrana basılması için do while döngüsünü bitirir. Farklı çalışan bilgisinin sorgusu yapılırken istenilen şekillerde belirtilmediği durumda tekrardan giriş yapılması istenmektedir.

### İstatistik Bilgilerinin Çıktısı

```
printf("\n##### İstatistik Tablosu #####\n");
printf("Toplam Odenen Net Ucret(TL): %.2f\n", t_net_ucret);
printf("Devlete Verilen Toplam Gelir Vergisi(TL): %.2f\n", t_vergi);
printf("Tum Calisanlarin Brut Ucretlerinin Ortalamasi(TL): %.2f\n", (t_brut/calisan_sayisi));
printf("Tum Calisanlarin Net Ucretlerinin Ortalamasi(TL): %.2f\n", (t_net_ucret/calisan_sayisi));
printf("Aylık 2000 TL Altında Maaş Alan Calisan Sayisi: %d\n", ikibin_alti);
printf("Evli Olanların Yuzdesi: %.2f\n", (float) ((evli*100)/calisan_sayisi));
printf("Bekar Olanların Yuzdesi: %.2f\n", (float) (((calisan_sayisi-evli)*100)/calisan_sayisi));

if(evli == 0)
    evli=1;
printf("Evli Olan Calisanların İçinde, Eslerinin Calisma Yuzdesi: %.2f\n", (float) ((es_calisma * 100)/evli));

if(cocuklu_calisan == 0)
    cocuklu_calisan=1;
printf("Sadece Bakmakla Yukumlu Cocugu Olanlar Dikkate Alinarak, Calisanların Bakmakla Yukumlu Oldukları Çocuk Sayisinin Ortalamasi: %.2f\n", (float) (t_cocuk_sayisi/cocuklu_calisan));
printf("Bakmakla Yukumlu olduğu Çocuk Sayisi 3'ten Fazla Olan Calisanların Sayisi: %d\n", cocuk_sayisi_uc);
printf("Engelli Calisan Sayisi: %d\n", engelli_calisan);
printf("Engelli Calisan Yuzdesi: %.2f\n", (float) ((engelli_calisan*100)/calisan_sayisi));
banknot_sayisi_yazdir(1);
printf("\n*** Gelir Vergisi Tablosu ***\n");
printf("Vergi Yuzdesi\tCalisan Sayisi\tVuzdesi\n");
printf("-----\t-----\t-----\n");

for(int i=0; i<4; i++)
    printf(" %d \t\t %d \t\t %.2f\n", vergi_yuzdesi[i][0], vergi_yuzdesi[i][1], (float) (vergi_yuzdesi[i][1] * 100 / calisan_sayisi));
```

İstatistik bilgileri için oluşturulan değişkenlerde oluşturulmuş bilgileri çekerek ekrana bastırıyoruz.



# ALGORİTMA VE PROGRAMLAMA

## PERSONEL MAAŞ HESAPLAMA YAZILIMI

```
for(int i =0; i<2; i++)
{
    if(i==0)
        printf("\n**** Aylık Brut Ücreti En Fazla Olan Çalışan Bilgileri ****\n\n");
    else
        printf("**** Aylık Net Ücreti En Fazla Olan Çalışan Bilgileri ****\n\n");

    printf("TC Kimlik Numarası: %ld\n",i_tc[i]);
    printf("Ad Soyad: ");
    if(i==0)
        printf("%s",b_ad);
    else
        printf("%s",n_ad);
    printf("Aylık Brut Ücret: %.2f\n",i_brut_ucret[i]);
    printf("Gelir Vergisi Kesintisi: %.2f\n",i_gelir_vergisi[i]);
    printf("Aylık Net Ücret: %.2f\n\n",i_net_ucret[i]);
}
```

En büyük brüt ücret ve net ücretli çalışanın bilgilerini çekmek için oluşturulmuş dizilerden bilgilerimizi çekiyoruz.

### GitHub KAYNAĞI

Oluşturulan projenin main.c dosyasına github hesabından ulaşabilirsiniz.

[Kaynağa Git](#)

# ALGORİTMA VE PROGRAMLAMA

## PERSONEL MAAŞ HESAPLAMA YAZILIMI

### KULLANICI KATALOĞU

#### Çalışan Bilgilerinin Girilmesi

```
TC Kimlik Nmaranızı Giriniz: 12345678901
Ad Soyad Bilgisini Giriniz(Max 30 karakter): ismet kadir
Brut Ucreti Giriniz (Aylık askeri ucret ya da daha büyük): 1900
Medeni Durumunu Giriniz 'Evli/Bekar (e/E/b/B karakterleri)': e
Evli ise Eşinin Çalışıp Çalışmadığı 'Evet/Hayır (e/E/h/H karakterleri)': e
Bakmakla Yukumlu Olduğu Çocuk sayısı 'tamsayı (0 ya da daha büyük)': 2
6 Yasından Büyük Çocuk Sayısını Giriniz: 1
Engelli Misiniz? Evet/Hayır (e/E/h/H karakterleri): e
Engel Oranını Giriniz 'tamsayı (1 ya da daha büyük ve 100 ya da daha küçük)': 50
```

Çalışanın kişisel bilgilerini baştan girilmesi isteniyor bunların hepsini koşullara uygun bir şekilde girmeniz gerekmektedir. Doğru girilmeyen bilgi karşısında sizden aynı bilgiyi tekrar isteyecektir.

#### Çalışan Bilgilerinin Ekrana Yazılması

```
##### Kaydı Gerçekleşen Çalışan #####
TC Kimlik Numarası: 12345678901
Ad Soyad: ismet kadir
Aylık Brut Ücreti(TL): 1970.00
Eş İçin Aile Yardım Odenegi(TL): 0.00
Çocuk İçin Aile Yardım Odenegi(TL): 70.00
Gelir Vergisi Kesintisi(TL): 264.00
Engel Derecesi(1/2/3): 3
Aylık Net Ücret(TL): 1706.00

**** Aylık Net Ücretin En Az Banknot ile Ödenmesi İçin Gerekli Banknot Adetleri ****

200 lira - 8 Adet
100 lira - 1 Adet
5 lira - 1 Adet
1 lira - 1 Adet
```

Karşınıza yukarıda görmüş olduğunuz gibi çalışana özel bazı istatistik bilgileri gelecektir.

# ALGORİTMA VE PROGRAMLAMA

## PERSONEL MAAŞ HESAPLAMA YAZILIMI

### Farklı Çalışan Sorgusu

```
Baska Calisan Var Mi ? 'Evet/Hayir (e,E,h,H karakteri)': h
```

Çalışan bilgilerini girdikten sonra karşınıza gelen çalışana özel çıktı ekranının en altında sizden farklı bir çalışan olup olmadığı sorgusu yapılır eğer “Evet” dersanız bir başka çalışanın bilgilerini aynı şekilde girmenizi sağlar. “Hayır” demeniz durumunda ise girmiş olduğunuz çalışanların hepsinin istatistik bilgisini size çıktı olarak verir.

### İstatistik Bilgilerinin Ekrana Yazılması

```
##### İstatistik Tablosu #####
Toplam Odenen Net Ucret(TL): 82063.55
Devlete Verilen Toplam Gelir Vergisi(TL): 43533.45
Tum Calisanlarin Brut Ucretlerinin Ortalamasi(TL): 62798.50
Tum Calisanlarin Net Ucretlerinin Ortalamasi(TL): 41031.77
Aylık 2000 TL Altında Maas Alan Calisan Sayisi: 1
Evli Olanların Yuzdesi: 100.00
Bekar Olanların Yuzdesi: 0.00
Evli Olan Calisanların İçinde, Eslerinin Çalışma Yuzdesi: 50.00
Sadece Bakmakla Yukunlu Cocugu Olanlar Dikkate Alinarak, Calisanların Bakmakla Yukunlu Oldukları Çocuk Sayisinin Ortalamasi: 2.00
Bakmakla Yukunlu olduğu Çocuk Sayisi 3'ten Fazla Olan Calisanların Sayisi: 0
Engelli Calisan Sayisi: 1
Engelli Calisan Yuzdesi: 50.00

**** Aylık Net Ucretin En Az Banknot ile Odenmesi İcin Gerekli Banknot Adetleri ****

200 lira - 409 Adet
100 lira - 2 Adet
50 lira - 1 Adet
5 lira - 2 Adet
1 lira - 3 Adet
50 kuruş - 1 Adet
1 kuruş - 4 Adet

**** Gelir Vergisi Tablosu ****

Vergi Yuzdesi   Calisan Sayisi   Yuzdesi
-----
15              1                50.00
20              0                0.00
27              0                0.00
35              1                50.00

**** Aylık Brut Ucreti En Fazla Olan Calisan Bilgileri ****

TC Kimlik Numarasi: 123123123
Ad Soyad: asdasd
Aylık Brut Ucret: 123627.00
Gelir Vergisi Kesintisi: 43269.45
Aylık Net Ucret: 80357.55

**** Aylık Net Ucreti En Fazla Olan Calisan Bilgileri ****

TC Kimlik Numarasi: 123123123
Ad Soyad: asdasd
Aylık Brut Ucret: 123627.00
Gelir Vergisi Kesintisi: 43269.45
Aylık Net Ucret: 80357.55
```

# ALGORİTMA VE PROGRAMLAMA

## PERSONEL MAAŞ HESAPLAMA YAZILIMI

Tüm çalışanların bilgileri girdikten sonra son işlem olarak size bütün çalışanlar ile ilgili birkaç istatistiki bilgi verecektir. Bu bilgilerin hepsini üst kısımda yer alan çıktıdan inceleyebilirsiniz.