

YILDIZ TEKNİK ÜNİVERSİTESİ
ELEKTRİK-ELEKTRONİK FAKÜLTESİ FAKÜLTESİ / BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

Öğrencinin Adı Soyadı:		Öğrenci No:		İmza:	
Dersin Adı: Yapısal Programlamaya Giriş		Tarih/Saat: 09.04.2019 09:00		Sınav süresi: 80	
Sınav Türü:	Vize 1				
Unvan Ad-Soyad: Doç. Dr. M. Fatih AMASYALI, Dr. Öğr. Üyesi. H. İrem TÜRKMEN (Ders Yürütücüsü)					

1. Bir firmaya bağlı satış elemanları farklı ürünleri farklı fiyatlara satmaktadırlar. Satış elemanlarının ID'leri, ürünlerin stok kodları, ürünlerin birim fiyatı ve satış adedi aynı sırayla N*4 boyutunda bir matriste tutulmaktadır. Aşağıda tanımlanmış fonksiyonları yazınız. Global değişken tanımlamayınız. (Ana fonksiyonda kullanılan BilgiAl() fonksiyonu satış fonksiyonunu kullanıcıdan almak için kullanılmaktadır. Bu fonksiyonu yazmanız gerekmez)

İpucu: Fonksiyona kullanıcı seçimlerini aktarırken sütun numaralarından yararlanabilirsiniz.

- Ana fonksiyonu boşlukları doldurarak tamamlayınız. İlave degisken tanımlamayınız. (10p)
- Kullanıcının tercihine göre verilen Urun Stok Kodu veya Satış Elemanı ID için o urun veya satıcıya ait Birim Fiyat veya Satış Adedi ortalamasını ve ortalamaya dahil olan kayıt sayısını bulan ortalamaBul() fonksiyonunu boşlukları doldurarak tamamlayınız. Belirtilenden fazla kontrol veya döngü kullanmayınız. İlave degisken tanımlamayınız. Kodun örnek bir çalışması ile elde edilmiş çıktılar aşağıda verilmiştir. (20p)

Örnek Çalışma:

N? 5
Matris[][]?

Satış Elemanı ID	Ürün Stok Kodu	Birim Fiyat	Satış Adedi
10	1	53	7
10	2	65	6
20	2	45	8
20	1	70	4
10	3	55	8

Urun kodu? 2
Birim Fiyat Ortalaması= 55.000 Urunu satan kisi sayisi= 2
Satış Adedi Ortalaması= 7.000 Urunu satan kisi sayisi= 2

Satış Elemanı ID? 10
Birim Fiyat Ortalaması= 57.666 Satış elemanının sattığı urun sayisi= 3
Satış Adedi Ortalaması= 7.000 Satış elemanının sattığı urun sayisi = 3

```
#include <stdio.h>
...float..... OrtalamaBul( int Tablo[][4], int N,int kodNo, int InputSecim, int OutputSecim, int *sayi)
{
    int i;
    float toplam=0;
    *sayi=0;.....
    for (i=0;i<N;i++)
    {
        if (.....Tablo[i][InputSecim]==kodNo.....)
        {
            .....toplam=toplam+Tablo[i][OutputSecim];.....
            .....(*sayi)++;.....
        }
    }
    return toplam/(*sayi);
}
```

```

int main()
{
    int Tablo[50][4];
    int N,UrunStokKod,SaticiID,adet;
    float Ort;

    printf("N?\n");
    scanf("%d",&N);

    printf("Matris[][]?\n");
    BilgiAl (Tablo, N);

    printf("Urun Stok kodunu?\n");
    scanf("%d",&UrunStokKod);
    Ort= OrtalamaBul (.....Tablo, N, UrunStokKod, 1, 2, &adet.....);
    printf("Birim Fiyat Ortalaması=%f Urunu satan kisi sayisi:%d\n",Ort,adet);

    Ort= OrtalamaBul (.....Tablo, N, UrunStokKod, 1, 3, &adet.....);
    printf("Satis Adedi Ortalaması=%f Urunu satan kisi sayisi:%d\n",Ort,adet);

    printf("Satis Elemanı ID?\n");
    scanf("%d",&SaticiID);
    Ort= OrtalamaBul (.....Tablo, N, SaticiID, 0, 2, &adet.....);
    printf("Birim Fiyat Ortalaması=%f Satis elemanının sattigi urun sayisi%d\n",Ort,adet);

    Ort= OrtalamaBul (.....Tablo, N, SaticiID, 0, 3, &adet.....);
    printf("Satis Adedi Ortalaması=%f Satis elemanının sattigi urun sayisi:%d\n",Ort,adet);
    return 0;
}

```

2. Aşağıdaki sözde kod çalışmasını bitirdiğinde D (5P), T (5P) ve R (10P)'nin değerlerini N'e bağlı olarak bulunuz. N, 3'ün pozitif bir üssü olarak girilmektedir. (Ör: N=243, N=27 vb.) (20p)

```

oku(N)
T=0;
D=1;
while (D<N)
    D=D*3;
    T=T+1;
end
R=1;
for i=1:T
    for j=1:D
        R=R*3;
    end
    R=R*2;
end
end

```

D=N
T=log₃N
R=2^{log₃N}*3^{N*log₃N}

3. Bir uzay gezgini N gezegenin olduğu bir sistemde gezegenden gezegene geçebilmek için solucan delikleri kullanmaktadır. Gezegenlerin birbirine geçişi olup olmadığı 0 ve 1'lerden oluşan $N \times N$ lik matriste tutulmaktadır. (Aralarında geçiş olan gezegenler için ilgili hücre 1, aksi takdirde 0'dır. Matrisin diagoneli 0'dır) Verilen bir x ve y gezegen çifti arasındaki yolculukta en fazla 1 gezegene uğranabildiğine göre kullanıcıdan alınan geçiş matrisi ve x,y gezegen çifti için takip edilebilecek kaç farklı yol olabileceğini hesaplayan programın akış diyagramını çizin. Çözümünüzde if, while ve switch kullanmayınız. (30p)

Örnek: N=4 mat[][]=

	1	2	3	4
1	0	1	1	1
2	1	0	1	0
3	1	1	0	1
4	1	0	1	0

x=2 y=4 arasında 2 farklı yol izlenebilir (2,1,4 ve 2,3,4)

x=1 y=4 arasında 2 farklı yol izlenebilir (1,4 ve 1,3,4)

```
int main()
{
    int mat[10][10];
    int N,x,y,i,p;
    printf("N?\n");          scanf("%d",&N);
    printf("Matris[][]?\n"); BilgiAl (mat, N);
    printf("x ve y?\n");     scanf("%d %d",&x, &y);

    p=mat[x][y];
    for (i=0;i<N;i++)
    {
        p=p+mat[i][y]*mat[x][i];
    }
    printf("yol sayisi:%d",p);
    return 0;
}
```

4. Aşağıdaki sözde kod çalıştırıldığında $(say1*2)/say2$ değeri N'in büyük değerleri için yaklaşık olarak kaçaya yakınsar? (20p)

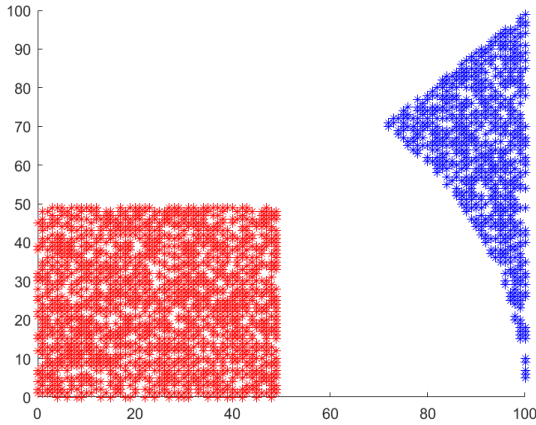
rand(): (0,1) aralığında ondalıklı rasgele bir sayı üretir.

round(x) : x'i aşağı ya da yukarı (hangisine yakınsa) yuvarlar.

Çözümünüzü bir şekil üzerinde gösteriniz.

İpucu: S1 ve S2'yi bir noktanın 2 koordinatı şeklinde düşünebilirsiniz.

```
oku(N)
A=100;
say1=0;
say2=0;
for i=1:N
    s1=round(rand()*A);
    s2=round(rand()*A);
    if (s1*s1+s2*s2>A*A) && (s1>s2)
        say1=say1+1;
    end
    if (s1<A/2) && (s2<A/2)
        say2=say2+1;
    end
end
yaz((say1*2)/say2)
```



Say2=A²/4 (kırmızı bölgenin alanı)

Say1= (A²-(pi*A²)/4)/2 (mavi bölgenin alanı)

say1*2/say2 = (A²-(pi*A²)/4) / (A²/4) = 4-pi