

BURSA TEKNİK ÜNİVERSİTESİ
MÜHENDİSLİK VE DOĞA BİLİMLERİ FAKÜLTESİ

BİLGİSAYAR MÜHENDİSLİĞİ

BLM0111 – Algoritmalar ve Programlama
Yılıçi Sınavı

Ad&Soyad	:
Öğrenci Numarası	:

Akademik yıl : 2022-2023
Dönem : Güz
Tarih : 22 Kasım 2022 – 17:00
Sınav süresi : 100 dakika
Öğr. görevlisi : Dr. Öğr. Üyesi Ergün GÜMÜŞ

Soru	1	2	3	4	5	Toplam
Puan	15	30	25	15	15	100
Not						

KURALLAR

- Sınava başlamadan önce Ad&Soyad ve Öğrenci numarası alanlarını doldurunuz.
- Sınav öncesinde ve süresince sınav gözetmenlerinin tüm uyarılarına uymanız gerekmektedir.
- Sınav öncesinde cep telefonlarınızı KAPATINIZ!
- Soruları yanıtlamak için sadece sınav kâğıdınızla beraber verilen kâğıtları kullanmanız gerekmektedir. Yanıtlarınız açık ve okunaklı olmalıdır.
- Sınav boyunca masanızın üzerinde bulunabilecek malzemeler sadece sınav kâğıdınız, kalem ve silgidir.
- Sınav süresince herhangi bir nedenle birbirinizle konuşmak ve malzeme (silgi, kalem, kâğıt vb.) alışverişi yasaktır.
- Bu kuralların herhangi birine uymamak kopya çekmeye yönelik bir hareket olarak değerlendirilir ve ilgili makamlara bildirilir.

Sorular

- 1) [15p] Aşağıda verilen koşullu atama yapısını üçlü (ternary) operatör kullanarak yeniden yazınız. Sorudan puan alabilmeniz için cevabınızın tamamen doğru olması gerekmektedir!

```
int a = -7, b = 5, c;

switch(b % 2){
    case 0:
        c = 10;
        break;
    default:
        switch(a > 0){
            case 0:
                switch(a == 0){
                    case 0:
                        c = 20;
                        break;
                    default:
                        c = 30;
                }
            break;
        default:
            c = 40;
        }
}
```

`c = (b%2 == 0)? 10 : (a > 0)? 40 : (a == 0)? 30 : 20;`
//Tek çözüm değildir. a değişkenine bağlı olan koşullar farklı sırada da yazılabilir.

- 2) [30p] Türkçe'de bir kelimeyi hecelerine ayırmak için 4 temel kural vardır:

Kural1: İki sesli harf arasındaki sessiz harf, kendinden sonraki sesli harf ile birlikte hecelenir.

KABİLE → **KA – Bİ – LE**

Kural2: İki sessiz harf yan yana geldiğinde ilk sessiz harf kendisinden önceki sesli harf ile, ikinci sessiz harf ise kendisinden sonraki sesli harf ile hece oluşturur.

BARDAK → **BAR – DAK**

Kural3: Üç sessiz harf yan yana geldiğinde birinci ve ikinci sessiz harfler kendilerinden önceki sesli harf ile, üçüncü sessiz harf ise kendisinden sonraki sesli harf ile hece oluşturur.

DOSTLUK → **DOST – LUK**

Kural4: İki sesli harf yan yana geldiğinde her birisi ayrı bir hece oluşturur.

SAAT → **SA – AT**

Yukarıda sayılan kuralları dikkate alarak, Türkçe karakter kullanmadan girilen bir kelimeyi bir karakter dizisine okuyan ve bu diziyi heceleyerek aşağıdaki çıktıyı üreten C programını yazınız. Kodunuza mümkün olduğunca açıklama satırı ekleyiniz.

```
Kelimeyi giriniz: muvaffakiyet
Toplam 5 hece var.
mu-vaf-fa-ki-yet
```

```

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

int main()
{
    char kelime[71];
    char sesliler[] = "aeoui"; //Türkçe karakter içermeyen sesli harfler
    char sesli_poz[71] = { }; //Sesli harflerin indislerini tutacak olan bir dizi (israf olmasın diye char)
    int N, sayac = 0;

    printf("Kelimeyi giriniz: ");
    gets(kelime);
    N = strlen(kelime);

    for(int i = 0; i < N; i++) //N harfin her biri
        for(int j = 0; j < 5; j++) // 5 tür sesli harfin her biri ile karşılaştırılıyor
            if(kelime[i] == sesliler[j]){ //kelimedeki i. harf sesli ise i indeksini sesli_poz dizisine kaydet
                sesli_poz[sayac++] = i;
                break; //kelime[i], sesli harf ise bunu diğer sesliler ile kıyaslamana gerek yok
            }

    printf("Toplam %d hece var.\n", sayac); //sesli harf sayısı kadar hece var

    int x, kir_sayac = 0;
    int kirilim[71] = { }; //hecelerin başladığı indisleri tutan bir dizi (israf olmasın diye char)

    for(int i = 0; i < sayac-1; i++){ //kırılım noktalarını tespit eden döngü
        x = sesli_poz[i+1] - sesli_poz[i]; //ardışık iki sesli harf arasında kaç sessiz harf var?
        if(x==1 || x==2) //KURAL1 veya KURAL4 ise kırılım noktası sesli_poz[i]'yi takip eder
            kirilim[kir_sayac++] = sesli_poz[i];
        else if(x==3) //KURAL2 ise kırılım noktası sesli_poz[i]+1'i takip eder
            kirilim[kir_sayac++] = sesli_poz[i] + 1;
        else if(x==4) //KURAL3 ise kırılım noktası sesli_poz[i]+2'yi takip eder
            kirilim[kir_sayac++] = sesli_poz[i] + 2;
    }

    int g = 0, h = 0;

    while(g < N){ //bütün karakterleri ve her kırılımda bir – sembolünü yazdıran döngü
        printf("%c", kelime[g]);
        if(kirilim[h] == g){
            printf("-");
            h++;
        }
        g++;
    }

    return 0;
}

```

3) [25p] Aşağıdaki konsol çıktısını inceleyiniz.

```
1      2      3      4      5      6      7
8      9      10     11     12     13     14
15     16     17     18     19     20     21
22     23     24     25     26     27     28
29     30     31     32     33     34     35
36     37     38     39     40     41     42
43     44     45     46     47     48     49

Istediginiz alt matrisin satir sayisi: 5
Istediginiz alt matrisin sutun sayisi: 2
Istediginiz alt matrisin satir baslangici (1'den baslar): 3
Istediginiz alt matrisin sutun baslangici (1'den baslar): 6
20     21
27     28
34     35
41     42
48     49
```

Konsol çıktısını gördüğünüz kodun amacı 7x7 boyutundaki bir karesel matrisin içinden kullanıcının istediği boyutta bir alt matrisi çıkartıp yine konsola yazdırmaktır.

Kod, öncelikle 7x7 boyutunda bir kare matris oluşturmakta ve bu matrisin içindeki elemanları 1-49 aralığındaki ardışık sayılarla doldurmaktadır. Ardından, bu matrisi konsola yazdırarak (sayılar TAB karakteri ile ayrılmıştır) kullanıcıdan 4 farklı parametre istemektedir. Bu parametreler sırasıyla,

- i) İstenen alt matrisin satır sayısı,
- ii) İstenen alt matrisin sütun sayısı,
- iii) İstenen alt matrisin ilk elemanının (alt matrisin sol üst köşe elemanı) 7x7'lik matristeki satır konumu,
- iv) İstenen alt matrisin ilk elemanının (alt matrisin sol üst köşe elemanı) 7x7'lik matristeki sütun konumu,

şeklindeir. Buna göre, şekildeki gibi 7x7'lik matrisin 3. satır ve 6. sütunundaki 20 sayısını sol üst köşe eleman olarak kabul eden 5x2'lik matris yazdırılmıştır.

Programınız, hatalı parametre girilmesi durumunda (örneğin, alt matrisin 7x7'lik matrisin içinde olmaması gibi) aşağıdaki gibi bir çıktı vermelidir.

```
1      2      3      4      5      6      7
8      9      10     11     12     13     14
15     16     17     18     19     20     21
22     23     24     25     26     27     28
29     30     31     32     33     34     35
36     37     38     39     40     41     42
43     44     45     46     47     48     49

Istediginiz alt matrisin satir sayisi: 4
Istediginiz alt matrisin sutun sayisi: 4
Istediginiz alt matrisin satir baslangici (1'den baslar): 5
Istediginiz alt matrisin sutun baslangici (1'den baslar): 1
Istediginiz matris olusturulamaz!
```

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

```
int main()
```

```
{
    const int N = 7;
    int matris[N][N];
```

```
    for(int i = 0, sayac = 1; i < N; i++)
        for(int j = 0; j < N; j++)
            matris[i][j] = sayac++;
```

```
    for(int i = 0; i < N; i++){ // 7x7'lik matrisi konsola yazdıran döngü
        for(int j = 0; j < N; j++)
            printf("%d\t", matris[i][j]);
        printf("\n");
    }
```

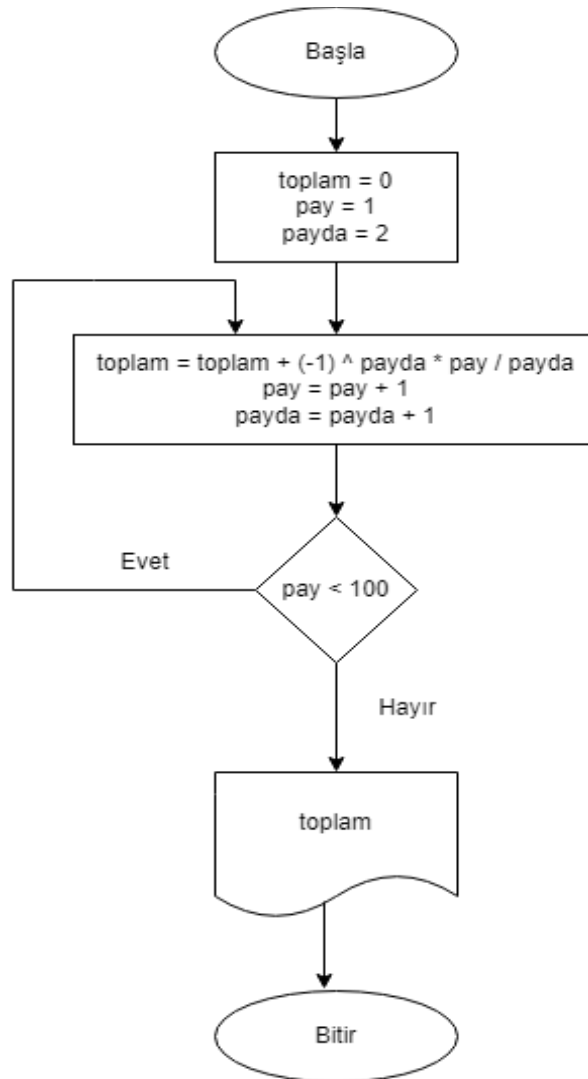
```
    int satir, sutun, i, j;
    printf("\nIstediginiz alt matrisin satir sayisi: ");
    scanf("%d", &satir);
    printf("Istediginiz alt matrisin sutun sayisi: ");
    scanf("%d", &sutun);
    printf("Istediginiz alt matrisin satir baslangici (1'den baslar): ");
    scanf("%d", &i);
    printf("Istediginiz alt matrisin sutun baslangici (1'den baslar): ");
    scanf("%d", &j);
```

```
    if( i+satir<=N+1 && j+sutun <= N+1 && satir>0 && sutun>0 && i>0 && j>0 ){
        for(int k = i-1; k < i-1+satir; k++){
            for(int t = j-1; t < j-1+sutun; t++)
                printf("%d\t", matris[k][t]);
            printf("\n");
        }
    }
    else
        printf("Istediginiz matris olusturulamaz!");
```

```
    return 0;
}
```

- 4) [15p] Aşağıda verilen işlemi gerçekleştirecek ve *toplam* sonucunu konsola yazdıracak algoritmanın eksiksiz akış şemasını (flowchart) çiziniz. Gerek duyduğunuz değişkenleri istediğiniz şekilde isimlendirebilirsiniz.

$$\text{toplam} = \frac{1}{2} - \frac{2}{3} + \frac{3}{4} - \frac{4}{5} + \frac{5}{6} - \dots - \frac{98}{99} + \frac{99}{100}$$



- 5) [15p] Aşağıda C dilinde yazılmış bir kodun konsol çıktısı görülmektedir. Bu çıktının oluşması için gerekli olan kodu tamamlayınız.

```

x sayisinin sekizlik sayi tabanindaki karsiligi: 113
x sayisinin %50'si: 37.500
x sayisinin karakter tablosundaki karsiligi: K
  
```

```

int main(){
    int x = 75;
    printf("x sayisinin sekizlik sayi tabanindaki karsiligi: %o\n", x);

    printf("x sayisinin %50'si: %.3f\n", x/2.0);

    printf("x sayisinin karakter tablosundaki karsiligi: %c", x);

    return 0;
}
  
```