

BURSA TEKNİK ÜNİVERSİTESİ
MÜHENDİSLİK VE DOĞA BİLİMLERİ FAKÜLTESİ

BİLGİSAYAR MÜHENDİSLİĞİ

BLM0111 – Algoritmalar ve Programlama
Yılıçi Sınavı

Ad&Soyad	: CEVAP ANAHTARI
Öğrenci Numarası	:

Akademik yıl : 2023-2024
Dönem : Güz
Tarih : 23 Kasım 2023 – 17:00
Sınav süresi : 75 dakika
Öğr. görevlisi : Doç. Dr. Ergün GÜMÜŞ

Soru	1	2	3	4	Toplam
Puan	25	25	25	25	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) Aşağıda, parametre olarak aldığı bir sayı dizisinin içindeki en büyük sayıyı geriye döndüren “enbuyuk” isimli bir özyinelemeli (recursive) fonksiyonun taslağı görülmektedir. Fonksiyonun ilk iki parametresinin ne olması gerektiği verilmiştir (dizi: sayı dizimiz, n: dizideki eleman sayısı). Gerekli gördüğünüz ek parametreleri parametre listesine ekleyerek fonksiyonun gövdesini yazınız. Herhangi bir döngü yapısı ya da global değişken kullanmanız durumunda sorudan puan alamazsınız!

```
int enbuyuk(int dizi[], int n, int p,int e ){
    if(p==n)
        return e;
    else if(dizi[p]>e)
        return enbuyuk(dizi,n,p+1,dizi[p]);
    else
        return enbuyuk(dizi,n,p+1,e);
}

void main(){
    int dizi[] = {5, 1, 9, 7, -4, 6, 2, 8};
    int n = sizeof(dizi) / sizeof(int);    //dizinin eleman sayısı
    int e = enbuyuk(dizi, n, 1,dizi[0] );
    printf("Dizideki en buyuk sayi %d", e); //Doğru çalışırsa 9 yazdıracak
}
```

2) DİKKAT!: Bu bir “ya hep ya hiç” sorusudur. Çözümde istenen tüm adımları doğru bir şekilde yazmanız durumunda sorudan tam puan alırsınız. Herhangi bir hata yapmanız durumunda sorudan puan alamazsınız.

a = 2, b = 9, c = 1 olmak üzere C dilinde yazılan aşağıdaki ifadeyi işlem öncelik sırasına göre adım adım çalıştırınız. Her adımın sonunda ifadenin sadeleşen halini ile gösterilen yere yazınız.

İfade: $a > b \% c != 0 > a / b * 2 \& c$

Adım1: $// a > 0 != 0 > a / b * 2 \& c$

Adım2: $// a > 0 != 0 > 0 * 2 \& c$

Adım3: $// a > 0 != 0 > 0 \& c$

Adım4: $// 1 != 0 > 0 \& c$

Adım5: $// 1 != 0 \& c$

Adım6: $// 1 \& c$

Adım7: ← İşlem sonucu $//1$

3) 5×5 boyutlu bir matrisin içerisindeki bir alt matrisin tüm elemanlarının ortalamasını bulan bir C kodumuz var. Bu kodun aşağıdaki çıktıyı vermesini istiyoruz:

2	8	x → 5	1	10
5	9	9	3	5
6	6	2	8	2
2	6	3	8	7
2	5	3	4	3 ← y

Sol-üst elemanın satır sütun değerlerini giriniz: 1 2

Sag-alt elemanın satır sütun değerlerini giriniz: 3 4

Ortalama: 5.222222

Üstünde çalıştığımız matris şekilde görüldüğü gibidir. Satır ve sütun indisleri 0'dan başlamak üzere 1. satır ve 2. sütunda bulunan x elemanından, 3. satır ve 4. sütunda bulunan y elemanına kadar oluşan alt matrisin (dikdörtgen içindeki alan) tüm elemanlarının ortalaması 5.22 olarak hesaplanmalıdır. Bu işi yapması gereken C kodumuz aşağıdadır, ancak kodda bazı hatalar vardır. Yeni bir satır eklemeyen, kodun hatalı satırlarını bularak (ilk 11 satırda hata yoktur) ilgili satırın yanına gerekli düzeltmeleri yazınız.

```

01. void main() //İLK 11 SATIRDA HATA YOKTUR!!!
02. {
03.     int matris[5][5];
04.
05.     for(int i = 0; i < 5; i++){
06.         for(int j = 0; j < 5; j++){
07.             matris[i][j] = rand() % 10 + 1;
08.             printf("%d\t", matris[i][j]);
09.         }
10.         printf("\n");
11.     }
12.
13.     int x_satir, x_sutun, y_satir, y_sutun, n;
14.     int ortalama = 0;    float ortalama = 0.0; (veya double)
15.
16.     printf("\nSol-üst elemanın satır sütun değerlerini giriniz: ");
17.     scanf("%d %d", &x_satir, &x_sutun);
18.     printf("Sag-alt elemanın satır sütun değerlerini giriniz: ");
19.     scanf("%d %d", &y_satir, &y_sutun);
20.
21.     n = (y_satir - x_satir) * (y_sutun - x_sutun);
22.     //n = (y_satir - x_satir + 1) * (y_sutun - x_sutun + 1);
23.     for(int i = x_satir; i < y_satir; i++) //<=
24.         for(int j = x_sutun; j < y_sutun; j++) //<=
25.             ortalama = matris[i][j]; //+=
26.
27.     ortalama /= n;
28.     printf("\nOrtalama: %d", ortalama); //%f (double ise %lf)
29. }

```

4)

Kullanıcıdan aldığı taban uzunluğuna göre konsola üçgen piramit çizdiren bir C kodu yazınız.

-Taban uzunluğunun tek sayı olup olmadığı program tarafından kontrol edilmelidir. Değilse uyarı verilip tekrar sorulmalı.

-Döngü işlemleri için istediğiniz döngü türünü kullanabilirsiniz.

-Koşullu ifadeler için sadece ama sadece ternary operatör kullanabilirsiniz. **if grubu veya switch/case yapısı kullanmak yasaktır.**

Örnek konsol çıktısı

```
Taban uzunlugunu giriniz: 10
>>Taban uzunlugu tek sayi olmalı
```

```
Taban uzunlugunu giriniz: 9
```

```
  *
 ***
*****
*****
*****
```

```
void main() {
    int t, y;

    do{
        printf("Taban uzunlugunu giriniz: ");
        scanf("%d", &t);
        getchar();
        printf("%s\n", (t%2==0)?">>Taban uzunlugu tek sayi olmalı\n":"" );
    }while(t % 2 == 0);

    y = t / 2 + 1; //piramidin yuksekligi

    for(int i = 0 ; i < y ; i++){
        for(int j = 0 ; j < t ; j++){
            printf("%c", ( j < y - i - 1 || j > y + i - 1) ? ' ' : '*' );
            printf("\n");
        }
    }
}
```