

BURSA TEKNİK ÜNİVERSİTESİ
MÜHENDİSLİK VE DOĞA BİLİMLERİ FAKÜLTESİ

BİLGİSAYAR MÜHENDİSLİĞİ

BLM0111 – Algoritmalar ve Programlama
Yılıçi Sınavı

Ad&Soyad	:	
Öğrenci Numarası	:	

Akademik yıl : 2018-2019
Dönem : Güz
Tarih : 8 Kasım 2018 – 13:00
Sınav süresi : 90 dakika
Öğr. görevlisi : Dr. Öğr. Üyesi Ergün GÜMÜŞ

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!
- 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)[9p] Bir algoritmanın sahip olması gereken i) Sonluluk, ii) Kesinlik, iii) Etkinlik özelliklerinin tanımlarını yapınız. Gerekliyse örnekler veriniz.

Yanıt

Sonluluk: Algoritmanın belli miktarda adımdan sonra sonlanıp bir çıktı üretmesi. Sonsuz döngüye girmemesi.

Kesinlik: Algoritmanın her adımının anlaşılır, açık ve net şekilde tanımlanması. Belirsiz ifadeler içermemesi (örneğin: "3. adımı birkaç kez tekrarla"– birkaç kezden kasıt kaç kere?)

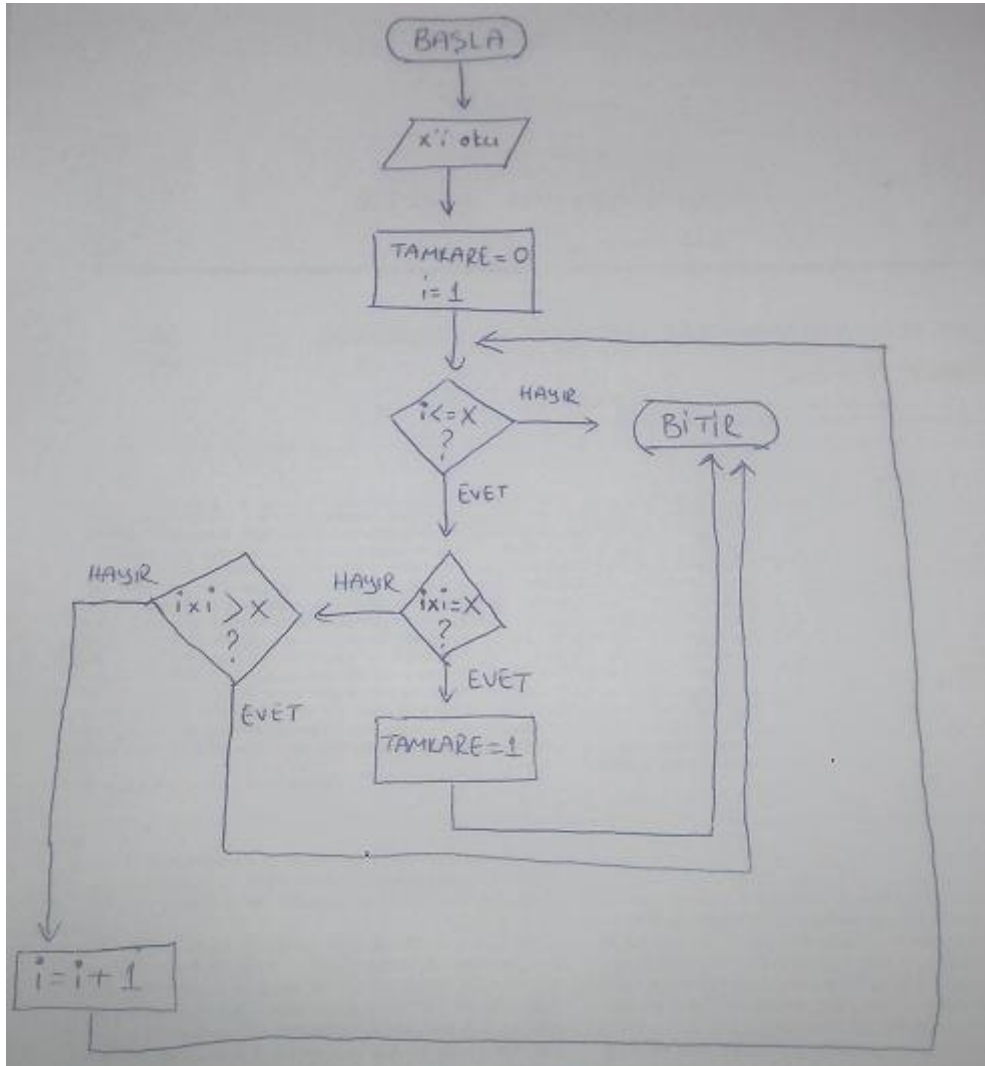
Etkinlik: Yazılan algoritmanın daha sonra başka bir projede tekrar kullanılabilir olacak şekilde yazılması

2)[11p] Kullanıcıdan alınan bir X tamsayısının karekökü bir tamsayı ise TAMKARE isimli değişkenin değerini 1, değilse 0 yapan bir algoritmanın eksiksiz akış şemasını çizin.

*Eğer X sayısının karekökü bir tamsayı ise TAMKARE=1
Aksi takdirde, TAMKARE=0*

Karekökü hesaplamak için tanımlanmış özel bir fonksiyonunuz (örneğin \sqrt{x}) gibi) YOKTUR !!!
Çözümünüzü bu duruma göre geliştirmeniz gerekmektedir.

Yanıt



3)[16p] Aşağıdaki birbirinden bağımsız kod parçacıklarını çalıştırıp ekran çıktılarını yanlarına yazınız.

Kod	Çıktı
<pre>int k=1; for(int i=1;i<16;k*=2) { for(int j=i;j<2*i;j++) printf("a"); printf("\n"); }</pre>	Sonsuz döngü, her iterasyonda ekrana a'n basıyor
<pre>int i=1; while(i<=8) { for(int j=1;j<i-(j/2);j++) printf("b"); printf("\n"); i++; }</pre>	b b bb bbb bbb bbbb bbbbbb
<pre>i=4; i*=5; printf("%d\n",i); printf("%d\n",(--i + 3)); printf("%d\n",(i++ + 10)); i+=i; printf("%d\n",++i);</pre>	20 22 29 41

4)[10p] Bu soruda bir hesap makinesi programı yazmanız istenmektedir. Öncelikle kullanıcıdan A ve B sayıları aralarında bir boşluk olacak şekilde aynı anda alınacak, ardından da işlem türü (1- Toplama, 2-Çıkarma, 3- Çarpma, 4-Bölme) sorulacaktır. Kullanıcının girdiği işlem türü Switch-Case yapısı ile kontrol edilerek dört işlemten biri yapıp sonuç ekrana yazdırılacaktır. İşlem türü 1-4 aralığında değilse uyarı verilecektir. Programın örnek ekran çıktısı aşağıdaki gibidir :

A ve B sayilarini giriniz: 85 9 (kullanıcı enter'a basar)
Islem turunu giriniz: 2 (kullanıcı enter'a basar)
>> 85 - 9 = 76

Başka bir senaryo

A ve B sayilarini giriniz: 85 9 (kullanıcı enter'a basar)
Islem turunu giriniz: 9 (kullanıcı enter'a basar)
>> Islem turunu hatali girdiniz!

Yanıt

```
#include <stdio.h>

void main(){
    int secim;
    double A,B;

    printf("A ve B sayilarini giriniz: ");
    scanf("%lf %lf",&A,&B);
    printf("Islem turunu giriniz: ");
    scanf("%d",&secim);

    switch(secim)
    {
        case 1:
            printf("%lf + %lf = %lf\n",A,B,A+B);
            break;
        case 2:
            printf("%lf - %lf = %lf\n",A,B,A-B);
            break;
        case 3:
            printf("%lf x %lf = %lf\n",A,B,A*B);
            break;
        case 4:
            printf("%lf / %lf = %lf\n",A,B,A/B);
            break;
        default:
            printf("Islem turunu hatali girdiniz");
    }
}
```

5)[20p] Kullanıcıdan aldığı 100 adet sıfırdan büyük tamsayı bir döngü yardımıyla teker teker *int* türünde bir sayı dizisine yerleştiren C programını yazınız. Eğer kullanıcı ondalık sayı veya negatif bir sayı girerse programınız bunu algılamalı ve ekrana ">> Pozitif tamsayı giriniz." yazarak az önce girilen sayının düzeltilerek tekrar girilmesini istemelidir. Programın örnek ekran çıktısı aşağıdaki gibidir.

```
1. sayi: -4 (kullanıcı enter'a basar)
>> Pozitif tamsayı giriniz.
1. sayi: 3.5 (kullanıcı enter'a basar)
>> Pozitif tamsayı giriniz.
1. sayi: 6 (kullanıcı enter'a basar)
2. sayi: 12 (kullanıcı enter'a basar)
3. sayi: -4.2 (kullanıcı enter'a basar)
>> Pozitif tamsayı giriniz.
3. sayi: ...
```

Yanıt

```
#include <stdio.h>

void main(){
    int sayilar[100],sayac=1;
    double girdi;

    while(sayac<=100)
    {
        printf("%d. sayi: ",sayac);
        scanf("%lf",&girdi);

        if((girdi>0)&&(girdi==(int)girdi))
        {
            sayilar[sayac]=(int)girdi;
            sayac++;
        }
        else
            printf(">> Pozitif tamsayı giriniz.\n");
    }
}
```

6)[10p] Aşağıdaki koşullu ifadeyi ternary operator (koşul?doğru:yanlış) yardımıyla sadece ama sadece bir deyim (satır) halinde yazınız.

```
if(c>1)
if(h>1)
    a=100;
else
    a=200;
else if(c<-2)
a=300;
else if(c== -1)
a=400;
else
a=500;
```

Yanıt

a = c > 1 ? (h > 1 ? 100 : 200) : (c < -2 ? 300 : (c == -1 ? 400 : 500));

7)[12p] Aşağıdaki kod parçasının sağdaki çıktıyı üretebilmesi için şeklinde boş bırakılan yerlere ne yazılmalıdır?

Kod	Çıktı
<pre>int i,j; for(i=1;i<10;i++){ if (i<5){ for(.....) printf("*"); printf("\n"); } else{ for(.....) printf(" "); for(.....) printf("*"); printf("\n"); } }</pre>	<pre>* ** *** **** ***** ***** **** *** ** *</pre>

Yanıt

1. boşluk: j=1; j<=i; j++
2. boşluk: j=0; j<i-5; j++
3. boşluk: j=1; j<=10-i; j++

Boşluklara yazdıklarım tek çözüm değildir. Başka çözümler de olabilir.

8)[12p] Ali, 10'ar elemanlı A ve B dizilerinin ortak elemanlarını X dizisine kaydeden ve ortak eleman sayısını ekrana yazan bir C programı yazmaya çalıştı. Ancak yazdığı kodda bazı hatalar var. Ali'nin kodundaki hatalar hangileridir ve nasıl düzeltilmeleri gerekir? İlgili satırın yanına yazınız.

```
1.   int A[10] = [ 1, 2, 3, 5, 8, 13, 21, 34, 55, 89 ]; //{ } olmalı
2.   int B[10] = [ 2, 3, 5, 7, 11, 13, 17, 19, 23, 29 ]; //{ } olmalı
3.   int X[10];
4.   int k, m, n; //n=0 olmalı
5.
6.   for ( k = 0; k < 10; ++k )
7.   {
8.       for ( m = 0; m < 10; ++m )
9.       {
10.          if (A[k] = B[m]) //== olmalı
11.          {
12.              X[k] = B[m]; // X[n] = B[m]; olmalı
13.              ++n;
14.          }
15.      }
16.  }
17.
18.  printf("Toplamda %c tane ortak eleman var.\n", n); //%c değil %d olmalı
```