

Kütahya Dumlupınar Üniversitesi
Bilgisayar Mühendisliği Bölümü
14/01/2021

1. Sorulara kendinizin cevap vermesi beklenmektedir. Kopya cevaplar kabul edilmeyecektir.
2. Sorular bu word dokümanı içerisine cevaplanacaktır. Cevaplama işlemini tamamlandıktan sonra OYS sistemine “**numaranız.pdf**” olarak (**20XX1317X0XX.pdf**) kaydetmeniz beklenmektedir. Dosya adı ve formatı farklı olduğunda ödeviniz 80 üzerinden değerlendirilecektir.
3. Özet olarak sisteme **numaranız.pdf** şeklinde bir adet dosya yükleyeceksiniz. Yazdığınız kodlar ve ekran çıktısını gösteren ekran görüntüsünü soru kısmında yer alması gerekmektedir. Kodlar da pdf dosyası içerisinde yer almalıdır.
4. **ZİP, RAR** gibi sıkıştırma dosyaları **YÜKLENMEYECEKTİR!** Bu dosyalar (zip, rar... vb.) dikkate alınmadan “**0 (sıfır)**” olarak değerlendirilecektir.
5. Ödevlerinizi **14 Ocak 2021 saat 23:00’a** kadar yüklemeniz gerekmektedir. Sistemsel sıkıntılar olabileceği için bu tarihten önce yükleme yapınız. Süre uzatımı **YAPILMAYACAKTIR.**

A. (24 puan) Aşağıdaki soruları cevaplayınız.

A1. Programlama dillerinde yer alan atama ifade türleri nelerdir?

<p>-Birleşik atama: örnek: $x = x + y$; yerine $x += y$;</p> <p>-Tekil atama: Örnek: $x = ++y$ (y arttırıldı ve arttırılan değer x'e atandı.) $++y$ y değişkeni 1 arttırıldı.</p> <p>-Şartlı atama: Örnek: $x = (z > 0) ? -1 : 1$; ($z > 0$) koşulu sağlanırsa $x = -1$ olur Aksi halde $x = 1$ ataması yapılır.</p>	<p>-Çoklu atama: perl programlama dilinden örnek; $(\\$bir, \\$iki, \\$uc) = (100, 200, 300)$;</p> <p>-Şartlı amaçlar: perl dilinde yazılmış örnek: $(\\$dogruMu ? \\$sayi1 : \\$sayi2) = 0$ dogruMu koşulu sağlanırsa(1) sayi1 degiskenine 0 atanır. Aksi halde(0) sayi2 degiskenine 0 atanır.</p> <p>-Bir ifade olarak atama: Örnek: $while((ch=getchar())!=EOF)\{\dots\}$</p> <p>-Karışık biçim ataması: Atama ifadeleri karışık biçimde de olabilir .</p>
--	--

A2. C'de yer alan Union ile Struct veri yapısını depolama ve verilere erişim açısından tartışınız.

<p>Union veri yapısı örneği:</p> <pre>union insan{ int boyCm; double kiloKg; };</pre> <p>Henüz bellekte yer ayrılmamıştır.</p> <p>union insan insan1; bildirimi yapılırken ayrılır.</p> <p>insan1.boyCm=190; insan1.kiloKg=97; ile verilere erişilir.</p> <p>→belleği daha verimli kullanır.</p> <p>→Derleyici union yapılar için bellekte yer ayracağı zaman, her zaman en büyük elemanın saklanabileceği kadar yer ayırır.</p> <p>→Verilere erişim sıkıntılı en son eklenen veri haricindekilere ulaşılıyor. Örneğin yukarıda tanımladığım insan1.boyCm değişkenine ulaşılıyor, ondan sonra kilo tanımladığım için.</p>	<p>Struct veri örneği:</p> <pre>struct insan{ int boyCm; double kiloKg; }insan1;</pre> <p>→verilere erişirken bir problem çıkmıyor.</p> <p>→erişmek için: insan1.boyCm=180; insan1.kiloKg=89;</p> <p>→verilerin erişimi ve depolanması açısından uniondan daha güvenilir.</p>
---	---

A3. Heap bölgesinde dizi tanımlamanın iki avantajını yazınız.

- büyük bellek kullanabilme
- esneklik

B. (30 puan) Aşağıdaki kavramsal olan kod parçalarının ekran çıktılarını şıklardaki durumlara göre cevaplayınız.

B1.

```
func f1(int x, int y, int z)
begin
  x = y + z;
  y = z + 1;
  print x, y, z;
end

func main
begin
  int r = 5;
  int t = 10;
  int k = 15;
  f1(r, t, t + k);
  print r, t, k;
end
```

1. F1 fonksiyonun bütün parametreleri pass-by-value ile gönderilirse ne yazdırır.
2. F1 fonksiyonun x, y pass-by-reference ve z pass-by-value ile gönderilirse ne yazdırır.
3. F1 fonksiyonun x, y pass-by-value-result ve z pass-by-value ile gönderilirse ne yazdırır.

1) 35 26 25 5 10 15	2) 35,26,25 35,26,15	3) 35,26,25 35,26,15
------------------------	-------------------------	-------------------------

B2.

```
func f2(int x, int y, int z)
begin
  y = y + 5;
  y = x + z + 4;
  print x, y, z;
end

function main
begin
  int t = 10;
  int k = 15;
  f2(t, t, t + k);
  print t, k;
end
```

1. F2 fonksiyonun bütün parametreleri pass-by-value ile gönderilirse ne yazdırır.
2. F2 fonksiyonun x, y pass-by-reference ve z pass-by-value ile gönderilirse ne yazdırır.
3. F2 fonksiyonun x, y pass-by-value-result ve z pass-by-value ile gönderilirse ne yazdırır.

1) 10 39 25 10 15	2) 44,44,25 44,15	3) 44,44,25 44,15
----------------------	----------------------	----------------------

C. (16 puan) Aşağıdaki soruları cevaplayınız.
C1. Yapıyı switch ile tekrardan kodlayınız.

```
int main() {
    int i, z;

    z = (-10) * 72 / 240;

    for (i = 0; i < 3; i++) {
        if ((z + 2) == 2) {
            z--;
        } else if ((z + 2) == 0) {
            z++;
        } else {
            z = 0;
        }

        if (z < 0)
            z = 3 - i;
    }

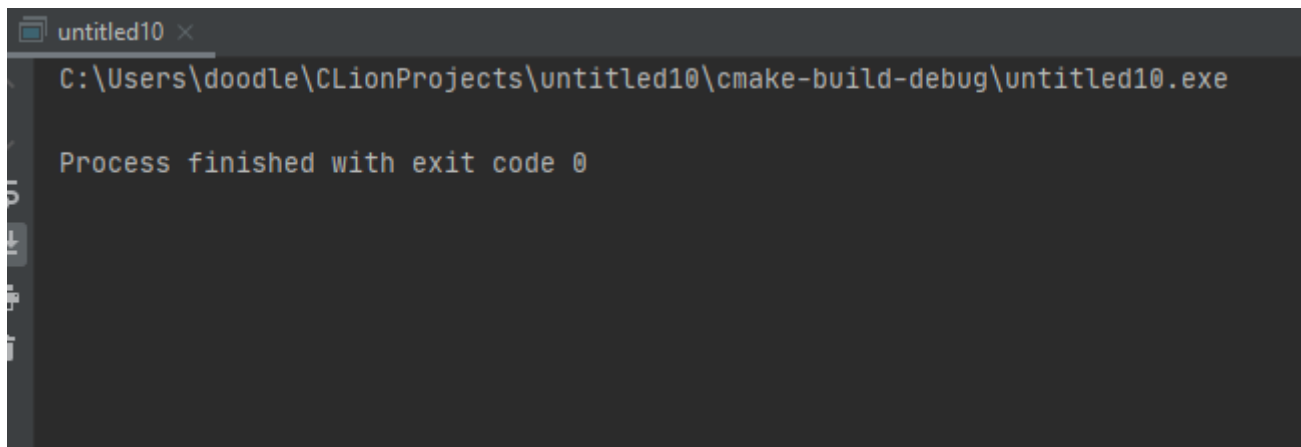
    return 0;
}
```

Cevap:

```
#include <stdio.h>

int main() {
    int i, z;
    z = (-10) * 72 / 240;
    for (i = 0; i < 3; i++) {
        switch (z + 2) {
            case 2:
                z--;
                break;
            case 0:
                z++;
                break;
            default:
                z = 0;
                break;
        }
        switch (z < 0) {
            case 1:
                z = 3 - i;
                break;
            default:
                break;
        }
    }

    return 0;
}
```



Çıktı verecek herhangi bir kod yok. Bu da kullandığım derleyicide sorunsuz çalıştığını gösteren ekran görüntüsü.

C2. Yapıyı while ile tekrardan yazınız.

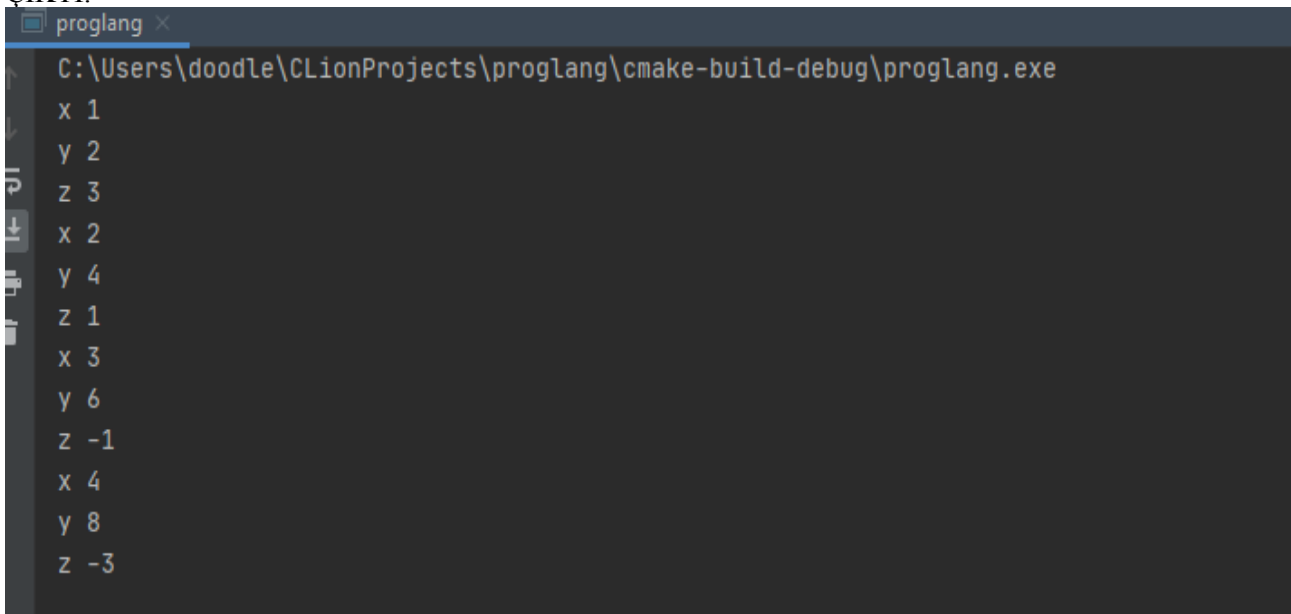
```
#include <stdio.h>
int main()
{
    int x, y, z;
    for (x=1, y=2, z=3; x<5 ; x++, y++, z++, y=y+1, z=z-1)
    {
        printf("x %d\n", x);
        printf("y %d\n", y);
        printf("z %d\n", z);
    }
    return 0;
}
```

Cevap:

```
#include <stdio.h>

int main() {
    int x, y, z;
    x = 1;
    y = 2;
    z = 3;
    while (x < 5) {
        printf("x %d\n", x);
        printf("y %d\n", y);
        printf("z %d\n", z);
        x++;
        y += 2;
        z -= 2;
    }
    return 0;
}
```

ÇIKTI:



```
proglang x
C:\Users\doodle\CLionProjects\proglang\cmake-build-debug\proglang.exe
x 1
y 2
z 3
x 2
y 4
z 1
x 3
y 6
z -1
x 4
y 8
z -3
```

D. (30 puan) C’de toplam(), ortalama(), en_küçük() adında üç tane fonksiyon yazmanız beklenmektedir. Bu üç fonksiyonda numaranızın her rakamını tutacak bir dizi tanımlanacaktır (20XX1317X0XX). Birinci fonksiyonda bu dizi static olarak, ikinci fonksiyonda heap bölgesinde tutulacak şekilde, üçüncü fonksiyonda ise stack bölgesinde yer ayrılacak şekilde tanımlayınız. Toplam() fonksiyonu dizideki elemanların toplamını hesaplamaktadır. Ortalama() fonksiyonu dizinin elemanlarının ortalamasını bulmaktadır. En_küçük() fonksiyonu ise dizideki en küçük elemanı bulmaktadır. Son olarak bu fonksiyonların geri dönüş değerlerini ekrana yazdıran main() fonksiyonunu tanımlayınız.

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

int toplam();

double ortalama();

int enKucuk();

int main() {
    int sayi = toplam();
    printf("toplami: %d\\t", sayi);
    int kucuk = enKucuk();
    printf("en kucuk: %d\\t", kucuk);
    double ortaa=ortalama();
    printf("ortalamasi: %lf",ortaa);
    return 0;
}

int toplam() {
    int toplami = 0;
    static int dizi[] = {2, 0, 1, 9, 1, 3, 1, 7, 1, 0, 4, 2};
    for (int i = 0; i < 12; i++) {
        toplami += dizi[i];
    }

    return toplami;
}

double ortalama() {
    int eleman = 12;
    double ort = 0;
    int *ptr;
    ptr = (int *) malloc(eleman * sizeof(int));
    ptr[0] = 2;
    ptr[1] = 0;
    ptr[2] = 1;
    ptr[3] = 9;
    ptr[4] = 1;
    ptr[5] = 3;
    ptr[6] = 1;
    ptr[7] = 7;
    ptr[8] = 1;
    ptr[9] = 0;
    ptr[10] = 4;
    ptr[11] = 2;
    for (int i = 0; i < 12; i++) {
        ort = ort + ptr[i] / 12.0;
    }

    free(ptr);
    return ort;
}

int enKucuk() {
```

```
int kucuk = -1;
int ogrNo[12] = {2, 0, 1, 9, 1, 3, 1, 7, 1, 0, 4, 2};
for (int i = 0; i < 12; i++) {
    if (kucuk == -1) {
        kucuk = ogrNo[i];
    }
    if (ogrNo[i] < kucuk) {
        kucuk = ogrNo[i];
    }
}
return kucuk;
}
```

F.

G. Kod çıktısı:

```
C:\Users\doodle\CLionProjects\proglang\cmake-build-debug\proglang.exe
toplami: 31      en kucuk: 0      ortalamasi:2.583333

Process finished with exit code 0
```

H.