

**BURSA TEKNİK ÜNİVERSİTESİ**  
**MÜHENDİSLİK VE DOĞA BİLİMLERİ FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ**

**BLM0111 – Algoritmalar ve Programlama**  
**Bitirme Sınavı**

Ad&Soyad	: <b>CEVAP ANAHTARI</b>
Öğrenci Numarası	:

Akademik yıl : 2022-2023  
Dönem : Güz  
Tarih : 18 Ocak 2023 – 11:00  
Sınav süresi : 75 dakika  
Öğr. görevlisi : Dr. Öğr. Üyesi Ergün GÜMÜŞ

Soru	1	2	3	4	5	6	7	8	Toplam
Puan	10	25	10	5	10	10	10	20	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) dosya.txt isimli metin dosyasında A-Z aralığındaki harfler yan yana yazılıdır. A harfinin 0. harf olduğunu kabul edersek, bu dosyadan sırasıyla 1. (B), 19. (T) ve 20. (U) harfleri çekerek yan yana "BTU" şeklinde konsola yazdıran koddaki boş bırakılan iki satırı doldurunuz. Koda yeni deyim eklemeniz halinde puan alamazsınız.

<b>dosya.txt</b>
ABCDEFGHIJKLMNOPQRSTUVWXYZ
<pre>void main(){     FILE *fptr = fopen("dosya.txt","r");     int poz[] = {1, 19, 20}; //İstenen harflerin pozisyonları      for(int i = 0; i &lt; 3; i++){         fseek(fptr, poz[i], SEEK_SET); [5p]         printf("%c", fgetc(fptr)); [5p]     }     fclose(fptr); }</pre>

2) Aşağıdaki C kodunda "Kayit" isimli bir veri tipi tanımlıdır. Bu veri tipinden oluşturulan her bir kayıt belirli bir uzunluktaki metinden oluşmaktadır. Bu metnin uzunluğu "uzunluk" isimli değişkene bağlı olduğundan ve bu uzunluk sonradan ayarlandığından, "metin", char \* türü bir pointer ile gösterilmektedir.

Kodun görevi,

- 20 kayıt tutan "x" statik dizisindeki her bir kayda, sonlandırıcı karakter ('\0') dahil 2 ila 10 karakter arası rasgele bir uzunluk vermek (kaydın "uzunluk" isimli değişkenini doldurmak),
- Kaydın içindeki char \*metin göstergeci, kaydın tam uzunluğu kadar karakter içeren bir dinamik diziye bağlamak,
- Kaydın içindeki metnin her bir karakterini A-Z aralığından rasgele bir harf ile doldurmak,
- Kaydın içindeki metni konsola yazdırmaktır.

Kodda eksik ya da fazla deyim yoktur. Ancak, bazı deyimlerde yazım hataları bulunmaktadır. Bu hataları tespit ederek ilgili deyimın yanına olması gereken doğru halini yazınız. Koda yeni deyim ekleyemezsiniz!

```
01. typedef struct{      //İlk 6 satırda hata yoktur. Değiştirmeyiniz.
02.     int uzunluk;
03.     char *metin;
04. } Kayit;
05.
06. void main(){
07.     srand(time(NULL));
08.     Kayit x[20];
09.
10.     for(int i = 0; i < 20; i++){
11.         int sayi = rand();      =rand()%9+2 olmalı [3p]
12.         uzunluk = sayi;        x[i].uzunluk= [3p]
13.         x[i].metin = (char*)malloc(sizeof(Kayit) * sayi);
14.         13. satırda (char *) [3p],sizeof(char) [3p]
15.         for(int j = 0; j < sayi; j++) j < sayi - 1 [3p]
16.             x[i].metin[j] = rand();  ='A'+rand()%26 [4p]
17.
18.         x[i].metin[sayi] = '\0';      [sayi-1] [3p]
19.         puts(x[i].metin[j]);        x[i].metin [3p]
20.     }
21. }
```

3) 2. sorudaki tüm hataları düzelttiğinizi ve kodun doğru bir şekilde çalıştığını varsayalım. Buna göre, **hiçbir şekilde [ ] (dizi parantezi) kullanmadan**, x dizisinin 5 indisli elemanındaki metnin 1 indisli karakterine erişip konsola yazdırmak istersek aşağıdaki deyimdeki boşluğu nasıl doldurmamız gerekir?

```
printf("%c", *( ( * ( x + 5 ) ).metin + 1 ) ); [10p]
```

4) Aşağıdaki kod, "dosya2.txt" isimli bir metin dosyasına 26 satır yazdırmaktadır. Her satırda bir sayı, TAB kadar boşluk ve bir harf bulunmaktadır. Buna göre koddaki boş satırı doldurunuz. Birden fazla deyim yazmanız halinde puan alamazsınız.

dosya2.txt		void main(){
1	A	FILE *fptr = fopen("dosya2.txt","w");
2	B	for(int i = 1; i <= 26; i++)
3	C	
4	D	fprintf(fptr, "%d\t%c\n", i, 'A'+i-1); [5p]
...		
25	Y	fclose(fptr);
26	Z	}

5) Aşağıdaki kodu incelediğinizde bir göstergeç zinciri kurulduğunu (ppp → pp → p → harfler) göreceksiniz. Sadece ppp göstergecini kullanarak (diğer göstergeçleri kullanmak yasak!) "harfler" dizisindeki 'S' harfini konsola yazdırmanın iki farklı yolunu 6. ve 7. satırdaki boşlukları doldurarak gösteriniz.

```
01. void main(){
02.     char harfler[] = "BURSA";
03.     char *p = harfler;
04.     char **pp = &p;
05.     char ***ppp = &pp;

06.     printf("%c\n", (**ppp)[3] ); //1. yol [5p]
07.     printf("%c\n", *(*ppp+3) ); //2. yol [5p]
08. }
```

6) >> programimiz 6 5 BURSA\_TEKNIK\_UNIVERSITESI <<  
>> \_TEKN

Yukarıda C dilinde yazılmış "programimiz" isimli bir programın komut satırından çalıştırılışı görülmektedir. Program, aralarında birer boşluk olacak şekilde üç parametreyi komut satırından almaktadır. İlk parametre başlangıç pozisyonu (**bas**), ikincisi miktar (**miktar**) ve üçüncüsü de bir metindir. Program, son parametre olarak verilen metnin, başlangıç pozisyonuncu harfinden itibaren miktar tane karakteri konsola yazdırılmaktadır. Bu işi yapan C kodu aşağıda verilmiştir. Koda deyim eklemeyen/silmeden boş bırakılan alanları doldurunuz. Kütüphane tanımlamalarını ihmal ediniz.

```
void main( int argc, char *argv[] ){ [3p] değişken isimleri farklı olabilir

    int bas = atoi(argv[1]) ; [2p]

    int miktar = atoi(argv[2]) ; [2p]

    for(int i = bas; i < bas + miktar; i++)

        printf("%c", argv[3][i-1] ); [3p]

}
```

7) Aşağıdaki kodun konsol çıktısını ... ile gösterilen yere yazınız.

```
void main(){
    int sayilar[20] = {0, 1};
    for(int i = 2; i < 20; i++)
        sayilar[i] = sayilar[i-1] + sayilar[i-2];
    int *p = &sayilar[3];
    printf("%d, %d", *p, *(p + p[3] - p[1]));
}
```

2, 21

[5p+5p]

8) Aşağıdaki kod, satir ve sutun sayısı #define anahtar kelimesi ile tanımlanan “matris” isimli bir matrisin içeriğini rasgele tamsayılarla doldurmakta ve “ortalama\_hesapla” isimli özyinelemeli fonksiyonla matrisin sütun ortalamalarını hesaplamaktadır. Buna göre “ortalama\_hesapla” fonksiyonunun imzasını değiştirmeden gövdesini tamamlayınız. Fonksiyon imzasını değiştirmeniz veya içerisinde herhangi bir döngü yapısı kullanmanız durumunda sorudan puan alamazsınız!

```
#define satir 6 //bunlar global değişken gibi düşünülebilir
#define sutun 4
```

```
void main(){
    srand(time(NULL));
    int matris[satir][sutun];
    double ortalama[sutun] = { }; //sütun ortalamalarını bu dizi tutacak

    for(int i = 0; i < satir; i++) //matrisi dolduran döngü
        for(int j = 0; j < sutun; j++)
            matris[i][j] = rand();

    ortalama_hesapla(matris, ortalama, 0, 0); // matris[0][0]'dan başla
}

void ortalama_hesapla(int matris[][sutun], double ortalama[], int a, int b){

    if(a != satir){
        ortalama[b] += matris[a][b] / (double)satir;
        b++;
        if(b == sutun){
            a++;
            b = 0;
        }
        ortalama_hesapla(matris, ortalama, a, b);
    }

}
```