

BURSA TEKNİK ÜNİVERSİTESİ
MÜHENDİSLİK VE DOĞA BİLİMLERİ FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ

BLM0111 – Algoritmalar ve Programlama
Bitirme Sınavı

Ad&Soyad	:	CEVAP ANAHTARI
Öğrenci Numarası	:	

Akademik yıl : 2021-2022
Dönem : Güz
Tarih : 27 Ocak 2022 – 09:00
Sınav süresi : 100 dakika
Öğr. görevlisi : Dr. Öğr. Üyesi Ergün GÜMÜŞ

Soru	1	2	3	4	5	Toplam
Puan	25	25	15	20	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) [25p] Bu soruda bir kodun parçalarını yazmanız istenmektedir. Arada bazı parçaların da verildiğini göreceksiniz. Buna göre aşağıdaki şıkları cevaplayınız.

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

#define LIMIT 20
#define KAYIT_SAYISI 5
```

a) [5 puan] “Kayıt” adında bir struct yapısı oluşturunuz (Adı “struct Kayıt” değil sadece “Kayıt” olmalı). Bu yapının içinde her biri en çok LIMIT adet harf tutabilen “ad” ve “soyad” isminde iki karakter dizisi tanımlayınız.

```
typedef struct {
    char ad[LIMIT+1];
    char soyad[LIMIT+1];
} Kayıt;
```

b) [5 puan] Parametre olarak aldığı “metin” isimli bir karakter dizisinin içerisini dolduran “uret” isimli fonksiyonu yazınız. Bu fonksiyonun imzası `void uret(char *metin)` şeklinde olmalı ve şu iki maddeye dikkat edilmelidir:

- i) Üretilcek metnin uzunluğu 1 ile LIMIT arasında rastgele bir sayıda olmalı.
- ii) Üretilcek metin A-Z aralığından rastgele seçilen büyük harflerle oluşturulmalı.

```
void uret(char *metin) {
```

```
    int j = 0, i = rand() % LIMIT + 1;

    for(; j < i; j++)
        metin[j] = rand() % 26 + 'A';

    metin[j] = '\0';
```

```
}
```

```
void main() {
    srand(time(NULL));
```

c) [5 puan] KAYIT_SAYISI adet kaydı tutacak statik bir dizi oluşturunuz ve bu dizinin her bir elemanını b şıkında yazdığınız uret isimli fonksiyondan faydalanarak dolduracak bir döngü yazınız. Bu döngü ürettiği her bir kaydın elemanlarını ekrana da yazdırmalıdır.

```

Kayit kayitlar[KAYIT_SAYISI];

for(int i = 0; i < KAYIT_SAYISI; i++){
    uret(kayitlar[i].ad);
    uret(kayitlar[i].soyad);
    printf("Kayit%2d\nAd: %s\nSoyad: %s\n\n", i+1, kayitlar[i].ad,
kayitlar[i].soyad);
}

```

d) [4 puan] *fopen()* fonksiyonunu kullanarak “kayitlar.txt” adında bir dosya açınız. Ardından, ürettiğiniz KAYIT_SAYISI adet kaydın tamamını bu dosyaya *fwrite()* fonksiyonunu kullanarak binary formatta yazdırınız. *fwrite()* fonksiyonunun imzası şu şekilde tanımlıdır:

```
fwrite(void *ptr, size_t size, size_t count, FILE *stream);
```

```
FILE *fptr;
```

```

fptr = fopen("kayitlar.txt", "w+");
fwrite(kayitlar, sizeof(Kayit), KAYIT_SAYISI, fptr);

```

e) [6 puan] *Dosyayı kapatmayınız. Kayıtları yazdırdıktan sonra fseek()* fonksiyonuyla baştan 4. kayda dönüp sadece bu kaydı *fread()* fonksiyonu ile okutunuz. Bu fonksiyonların imzaları şu şekilde tanımlıdır:

```
fseek(FILE *stream, long offset, int origin);
```

```
fread(void *ptr, size_t size, size_t count, FILE *stream);
```

Okuttuğunuz tekil kaydı ekrana yazdırıp dosyayı kapatınız.

```

Kayit okunan_kayit;
fseek(fptr, 3 * sizeof(Kayit), SEEK_SET);
fread(&okunan_kayit, sizeof(Kayit), 1, fptr);
fclose(fptr);

printf("Okunan Kayit: %s %s\n", okunan_kayit.ad, okunan_kayit.soyad);

```

2) [25p] Ali, basit bir şifreleme programı yazmak istiyor. Buna göre, kullanıcı tamamı büyük harflerden oluşan maksimum 900 karakterlik bir metin girecektir. Program bu metnin harflerini, olabilecek en küçük boyutlu karesel matrisin içerisine sütunlar halinde yerleştirecektir. Matrisin boş kalan kısımlarını da X karakteri ile dolduracaktır. Ardından bu matrisin satırlarını ucuca ekleyerek şifreli metin oluşturulacaktır. Aşağıda, programın örnek konsol çıktısı görülmektedir:

```
Sifrelenecek metni giriniz (max 900 karakter): ALGORITMA VE  
PROGRAMLAMA DERSI FINAL SINAVI
```

```
Olusturulan matris (7x7):
```

A	M	R	A	S	L	I
L	A	O	M	I		X
G		G	A		S	X
O	V	R		F	I	X
R	E	A	D	I	N	X
I		M	E	N	A	X
T	P	L	R	A	V	X

```
Sifreli metin: AMRASLILAOMI XG GA SXOVR FIXREADINXI MENAXTPLRAVX
```

Aşağıdaki şıklardaki işlemleri yaparak Ali'ye yardım ediniz.

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

```
void main() {  
    char metin[901];  
    char sifreli_metin[901];  
    int uzunluk, i, j, n, sayac = 0;  
    char **matris;  
  
    printf("Sifrelenecek metni giriniz (max 900 karakter): ");  
    gets(metin);  
    uzunluk = strlen(metin);  
    n = ceil(sqrt(uzunluk));    //n, matrisin satır/sütun sayısını gösteriyor.
```

a) [8p] $n \times n$ boyutlu karesel matrisi dinamik yoldan oluşturmak için aşağıdaki kodda şeklinde boş bırakılan yerleri doldurunuz.

```
matris = (char **) malloc( sizeof( char * ) * n );  
for(i = 0; i < n; i++)  
    matris[i] = (char *) malloc( sizeof( char ) * n );
```

b) [8p] “matris”in sütunlarını “metin” isimli dizinin elemanları ile dolduracak döngü yapısını yazınız.

```
for(i = 0; i < n; i++)  
    for(j = 0; j < n; j++)  
        if(sayac < uzunluk)  
            matris[j][i] = metin[sayac++];  
        else  
            matris[j][i] = 'X';
```

```
printf("\nOlusturulan matris (%dx%d):\n\n", n, n);
```

c) [9p] İçi doldurulan “matris”i, örnek konsol çıktısındaki gibi ekrana yazdıran ve aynı zamanda da bu matrisin satırlarındaki elemanları tek tek “sifreli_metin” dizisine ekleyen döngüyü yazınız.

```
sayac = 0;  
for(i = 0; i < n; i++) {  
    for(j = 0; j < n; j++) {  
        printf("%c\t", matris[i][j]);  
        sifreli_metin[sayac++] = matris[i][j];  
    }  
    printf("\n");  
}  
sifreli_metin[sayac] = '\0';
```

```
printf("\nSifreli metin: %s", sifreli_metin);
```

```
}
```

3) [15p] Bilgisayar sistemlerinde *int* veri tipi 4 bayt ile ifade edilir ve bu tipteki her tamsayı belleğe yerleştirilirken Little-Endian veya Big-Endian formatlarından biri kullanılır. Bu yerleşimin bir örneği aşağıdaki şekilde görülmektedir (0x..... sayının 16'lık sayı formatında olduğunu göstermektedir.) :

	Little-Endian	Big-Endian
Adres	Bellek gözü içeriği	Bellek gözü içeriği
0xAAAA1234	78	12
0xAAAA1235	56	34
0xAAAA1236	34	56
0xAAAA1237	12	78

Buna göre sayıları Little-Endian formatıyla belleğe kaydeden bilgisayar sistemindeki *eleman_sayisi* adet tamsayıyı Big-Endian formatına çeviren bir C fonksiyonu yazınız (main bloğunu yazmanız istenmiyor). Fonksiyonunuzun imzası aşağıdaki gibi olmalıdır:

```
void littleToBIG(int sayilar[ ], int eleman_sayisi) {
// sayı dizisi aktarılırken pass-by-reference tekniği kullanılmış
```

```

    char *p; //her byte'ı gösterecek olan pointer
    char val[4];

    for(int i = 0; i < eleman_sayisi; i++) {
        p = &sayilar[i]; //i. tamsayının ilk byte'ının adresi (mesela 0xAAAA1234)
        for(int j = 0; j <= 3; j++)
            val[j] =* (p + j); //bu adresin j göz ötesindeki değeri val dizisine atalım
        for(int j = 0; j <= 3; j++)
            *(p + j) = val[3 - j]; //val dizisini takas işlemi için kullanalım
    }
}
```

4) [20p] Pozitif bölenlerinin sayısına tam bölünen sayma sayılarına “Tau sayısı” denir. Örneğin, 9 sayısının 1, 3 ve 9 şeklinde 3 adet pozitif böleni vardır. 9 sayısı da 3’e tam bölündüğü için bir Tau sayısıdır. Benzer şekilde 40 sayısı da bir Tau sayısıdır. Çünkü 1, 2, 4, 5, 8, 10, 20, 40 şeklinde 8 adet pozitif böleni vardır ve 40 sayısı 8’e tam bölünür.

Bu soruda sizden, herhangi bir sayıyı parametre olarak alan ve bu sayı bir Tau sayısı ise geriye 1, aksi takdirde geriye 0 döndüren bir özyinelemeli fonksiyon yazmanız istenmektedir. Bu fonksiyonun parametrelerini istediğiniz gibi ayarlayabilirsiniz, ancak eğer bu konuda bir fikriniz yoksa aşağıdaki fonksiyon imzasını da kullanabilirsiniz:

```
int taumu(int sayi, int a, int b) {  
    // sayi: Tau sayısı olup olmadığı kontrol edilecek olan sayı  
    // a: sayi'yi tam bölüp bölmediği kontrol edilecek olan başka bir sayı  
    // b: Herhangi bir anda sayi'yi o ana kadar tam bölen sayıların adedi  
}  
  
void main() {  
    int x = 40;  
    printf("%d", taumu(x, x, 0) );  
}
```

NOT: SORUDA HERHANGİ BİR ŞEKİLDE DÖNGÜ KULLANMANIZ DURUMUNDA PUAN ALAMAZSINIZ!

```
//bu kod, sayi isimli değişkeni [sayi → 1] aralığındaki tüm a sayılarına bölmeye çalışır.  
int taumu( int sayi, int a, int b ) {  
    if( a == 1 ) //temel durumda 1, sayi'yi tam böler. O yüzden aşağıda b+1 diyoruz.  
        if( sayi % (b + 1) == 0 )  
            return 1;  
    else  
        return 0;  
    else  
        if( sayi % a == 0 ) //a, sayi'yi tam bölüyorsa bölen sayısını (b) 1 artırarak çağır  
            return taumu( sayi, a - 1, b + 1 );  
        else //aksi takdirde bölen sayısı artmamıştır.  
            return taumu( sayi, a - 1, b );  
}
```

5) [15p] main() fonksiyonuna parametre aktarımı tekniğini kullanarak bir hesap makinesi yapmanız isteniyor. Programınız konsoldan çalıştırılırken aralarında bir boşluk olacak şekilde 3 parametre alacaktır. Bu parametreler sırasıyla; 1. sayı (ondalık sayı), 2. sayı (ondalık sayı), işlem sembolü (+, -, *, /) olmalıdır. Argüman sayısı ya da aritmetik işlem işleci hatalı verilirse programınız aşağıdaki örnek çıktıda görüldüğü gibi hata mesajları üretmelidir.

```
> hesapla.exe 12.0 4.0 5.0 *
Hatali sayida arguman!
> hesapla.exe 12.0 4.0 @
Hatali islec!
> hesapla.exe 12.0 4.0 /
Sonuc: 3.00
> hesapla.exe 12.0 4.0 +
Sonuc: 16.00
```

Not: Metin formatındaki bir sayıyı double tipi bir değişkene çevirmek için atof() fonksiyonunu kullanabilirsiniz: *double atof(const char *str)*

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

int main(int argc, char *argv[]) {
    if(argc != 4)
        printf("Hatali sayida arguman!");
    else{
        double a, b, sonuc, hata = 0.0;
        a = atof(argv[1]);
        b = atof(argv[2]);

        if(strcmp(argv[3], "+") == 0)
            sonuc = a + b;
        else if(strcmp(argv[3], "-") == 0)
            sonuc = a - b;
        else if(strcmp(argv[3], "*") == 0)
            sonuc = a * b;
        else if(strcmp(argv[3], "/") == 0)
            sonuc = a / b;
        else
            hata = 1.0;

        if(hata == 1.0)
            printf("Hatali islec!");
        else
            printf("Sonuc: %0.2lf", sonuc);
    }
    return 0;
}
```