

Süre: **24 Saat**

ALGORİTMALAR ve PROGRAMLAMA

FİNAL SINAVI ÖDEVİ

* Ödev/sınav dosyasını kendi öğrenci numaranız ve kendi ad & soyadınıza göre aşağıda gösterildiği gibi adlandırınız. (Dosyayı isimlendirirken Türkçe karakter kullanmayınız, boşluk bırakmayınız, alt çizgiler ile ayırmayı yapınız).

181234567_Cenk_Ozdemir.doc

* Dosya türü (formatı) olarak sadece Microsoft Office Word (**doc, docx**) gönderiler kabul edilecektir. Farklı uzantılar (örneğin; .txt, .pdf, .rtf, .odt, .rtf, ... vb) seklinde ödevini gönderen öğrencinin sınavı değerlendirmeye alınmayacaktır.

* Sadece Haruzem Moodle üzerinden gönderilmiş ödevler kabul edilecektir. Taslak olarak kaydedilmiş ama submit edilmemiş ödevler değerlendirmeye alınmayacaktır.

* Ödevinizi Haruzem Moodle üzerinden sisteme yükleyiniz, eposta olarak atmanız durumunda değerlendirmeye alınmayacaktır.

* Verilen soruda kod yazılması istenildiğinde, cevaplarınızı doğrudan Word'de yazabileceğiniz gibi derleyicilerde de yazıp word'e aktarabilirsiniz. Fakat derleyicilerde yazdığınız kodunuzun word'e düz metin olarak (screenshot değil) doğru aktardığınızdan emin olunuz. Word de kodunuzu yazarken paragraf girintisi, noktalı virgül kullanımı, eğer varsa Int yerine int kullanımı, " " yerine " " ve benzeri ayrıntılara dikkat ediniz. Çok fazla bu tür hatalar içeren cevaplar değerlendirmeye alınmayacaktır.

* Derleyiciden kaynaklı hataların önüne geçmek için yazdığınız ve çalışan kodunuzun çıktılarının görüntülediği konsol (console) ekran görüntüsünü küçük boyutlu bir resim olarak ilgili soruya cevabınızın sonuna ekleyebilirsiniz.

*** Kopya çekilmesi, web'den başka(ları/sı)na ait kod, resim, grafik, tablo veya düz metin materyalin kullanılması, herhangi bir compiler'da yazdığınız kodun veya farklı bir kaynaktan (internet, kitap, vb) alınan materyalin (örneğin, düz metin, tablo, grafik gibi) bir ekran görüntüsü ya da resim şeklinde ödev dosyasına eklenmesi, ... vb etik olmayan durumlarda sınavınız geçersiz sayılıp, gereği yapılacaktır.

* Ödevinizi doğru şekilde (sadece bir defa yükleme şansınız olacaktır) ve verilen süre içerisinde yüklediğinizden emin olunuz. Son teslim tarih ve saatinden sonra hiç bir ödev kabul edilmeyecektir. Ayrıca ek süre verilmeyecektir.

* Cevaplarınızda yazı tipi rengi olarak kırmızı renk yerine siyah, mavi renkleri tercih ediniz.

* Boşlukları tamamlama sorularında cevaplarınızı verilen boşluklara yazmayınız. Yeni baştan yazınız.

* Yazı tipi olarak (Times New Roman, Calibri, Arial, Consolas, Courier New vb. gibi) kolay okunabilir formatları tercih ediniz ve yazı boyutu olarak 12 puntodan büyük olmamasına dikkat ediniz. Ayrıca satırlar arası gereksiz fazla boşluk bırakmayınız.

1) Aşağıda ekran çıktıları yandaki kutucukta paylaşılan kodlarda boşlukları tamamlayınız.

a)

KOD:	ÇIKTI:
<pre>..... for (i = 0; i < ;) if ((i %) == 0) printf("A"); else if ((i %) == 0) </pre>	ACABACACAB

b)

KOD:	ÇIKTI:
<pre>int { char []="BADGERS"; char*ptr=string; *ptr=*ptr+2; ptr=ptr+2; printf("%c",); ptr--; printf("%c",*ptr); ptr=..... ; </pre>	DAD

2) Girilen tüm sayıların pozitif olduğunu varsayılarak 50 pozitif tamsayı sayısını okumak ve çift ve tek sayıların sayısını ayrı ayrı saymak için bir program yazın. Örneğin, program şu şekilde bir çıktı yazdırmalıdır: 23 çift sayı ve 27 tek sayı girdiniz. Örnek program çıktısı:

```
Toplam sayı adedi giriniz
5
Sayi giriniz
14
.
.
.
Sayi giriniz
03

Cift sayi adedi: 2
Tek sayi adedi: 3
```

3) Girilen bir dizideki en büyük ve en küçük elemanı fonksiyon ile bulan bir C programı yazınız. Program aşağıda yer alan fonksiyon prototiplerini içermelidir:

```
int EB(int *dizi) //en buyuk elemani bulan fonksiyon
int EK(int *dizi) //en kucuk elemani bulan fonksiyon
```

Örnek program çıktısı aşağıdaki gibidir:

```
Sayıyı giriniz
25
Sayıyı giriniz
100
Sayıyı giriniz
80
Sayıyı giriniz
53
Sayıyı giriniz
88
En buyuk sayi: 100
En kucuk sayi: 25
```

4) Insertion Sort Algoritması, düzensiz olan dizi elemanlarını tek tek ele alarak her birini dizinin sıralanmış kısmındaki uygun yerine yerleştirme esasına dayanır.

Aşağıda bir programcı tarafından kodlanan Insertion Sort programı ve örnek ekran çıktısı verilmiştir. **Sadece** boşluklara gelecek kodları/kısımları numara sırasına göre yazınız.

Sıralanmadan önce 3, 5, 8, 2, 1, 6, 4, 7, 12, 10, 9, 1, 5, 11, 14, 13 Sıralandıktan sonra 1-2-3-4-5-6-7-8-9-...
--

```
#include <stdio.h>
#include <stdlib.h>
#define _____ (1)
//prototype
void insertion_sort_____ (2)
int main()
{
    (3) _____ //Sıralanacak dizi
    insertion_sort(array_to_be_sorted, SIZE); // Sıralanacak dizi
fonksiyona gönderilir
    return 0;
}
void insertion_sort _____ (4)
    int counter = 1;

    int i, temp;
    int pivot = sorted_array[1];
    while(counter < _____ (5)) {
        for(i = counter - 1; i >= 0; i--){
            if(pivot < sorted_array[i]) {

                temp = sorted_array[i];
                _____ (6)
                _____ (7)
            }
            _____ (8)
        }
        pivot = _____ (9) ;
    }
}
```

5) Öğrencilerin notları ve isimleri aşağıda verilen yapı kullanılarak tanımlanmalıdır:

```
struct ogr
{
    char adi[50];
    float odev1=0.05, odev2=0.05;
    float quiz1=0.08, quiz2=0.08, quiz3=0.08;
    float vize1=0.13, vize2=0.13;
    float final=0.40;
    float gecme_notu;
};
```

- Öğrencilerin isimleri ve notları kullanıcıdan alınıp dinamik bir dizide tutulmalıdır. Dinamik dizinin boyutu kullanıcıdan alınmalıdır.
- Öğrenci notları, büyükten küçüğe sıralanmalıdır.
- Öğrenci geçme notu ve öğrenci isimleri “notlar.txt” isimli bir dosyaya işaretçi kullanılarak kaydedilmelidir.

Buna göre, yukarıda belirtilen ilkelere bağlı kalarak, ders notlarının hesaplanmasını sağlayan bir C programı yazınız.