



## 6. Bilgisayar Olimpiyatları Final Yarışması

4 Mayıs 2019 Cumartesi, saat 13:00

Toplam Süre: 200 dk.

Toplam Soru Sayısı: 4

**Her Soru İçin Kullanılması Zorunlu Şablon Dosya (Süre Ölçümü İçin Gerekli)**

```
#include <time.h>
#include <stdio.h>
int main()
{
    clock_t tic = clock();

    //kodunuzu buraya yazın.....
    // fonksiyon çağrılarını yapabilirsiniz.

    clock_t toc = clock();

    printf("Toplam Süre: %f saniye\n", (double)(toc - tic) / CLOCKS_PER_SEC);
    return 0;
}
```

**Yarışmanın kurallar listesi aşağıdaki gibidir;**

1. Yarışmacılar laboratuvarında bulunan bilgisayarları kullanacaklar. (Kendi PC 'nizi getiremezsiniz)
2. Yarışma esnasında internet bağlantınız olmayacak ve telefonlarınız alınacaktır.
3. Yardımcı kaynak (kitap) getirebilirsiniz.
4. Cevapların değerlendirilebilmesi için kodun derlenebilmesi şarttır.

**Soruların Değerlendirme Kriterleri**

1. Sonucun doğruluğu
2. Çözüm üretme süresi
3. Çalıştırılabilir dosyanın boyutu

**Başarılar...**

### Soru 1)

input.txt giriş dosyasının ilk satırı matrisin satır ve sütun sayısını ve dönme sayısını, diğer satırlar ise sırasıyla matrisin elemanları içermektedir. Dosyadan bu bilgileri okuyup matrisin dış kenarında bulunan değerleri dönme sayısı kadar döndüren ve sonucu output.txt dosyasına yazan C/C++ kodunu yazınız.

**Not:** Dönme sayısı pozitif ise saatin dönme yönü, negatif ise saatin dönme yönünün tersi alınmalıdır.

### Örnek 1:

| input.txt  | output.txt                       |
|--|----------------------------------|
| 3 4 3<br>1 2 3 4<br>5 6 7 8<br>9 10 11 12                                      | 10 9 5 1<br>11 6 7 2<br>12 8 4 3 |
| 3 satır 4 sütunlu bir matrisin dış elemanları saat yönünde 3 kez kaydırılacak. |                                  |

### Örnek 2:

| input.txt   | output.txt                                 |
|---|--|
| 4 4 -2<br>-1 0 2 3<br>1 5 9 8<br>4 5 6 7<br>-3 0 5 8                                  | 2 3 8 7<br>0 5 9 8<br>-1 5 6 5<br>1 4 -3 0 |
| 4 satır 4 sütunlu bir matrisin dış elemanları saatin ters yönünde 2 kez kaydırılacak. |  |

### Soru 2)

Giriş dosyasında verilen bir stringin içerisinde alt string olarak palindrome stringlerin araştırılması istenmektedir. 3 ve daha çok karaktere sahip tüm palindrome stringlerin en büyük boyutludan en küçüğe doğru çıkış dosyasına yazılması beklenmektedir. Boyutlar aynı olduğunda alfabetik sıraya göre yazdırılmalıdır.

**Not:** Tersten yazılışı kendine eşit olan string palindrome bir string tir.

**Örn;** "ey edip adanada pide ye".

### Örnek:

| input.txt            | output.txt                    |
|----------------------|-------------------------------|
| abbaadeabadebbedyeda | debbed<br>abba<br>ebbe<br>aba |

### Soru 3)

Giriş dosyasında değerleri verilen tek boyutlu tam sayılar içeren bir dizi içerisinde yan yana elemanlardan oluşan alt dizilerin elemanların çarpılması istenmektedir. Bu şekilde çarpımı maksimum olan alt dizinin elemanlarının en küçüğü ve en büyüğü arasındaki değer farkını çıkış dosyasına yazınız. **Not:** Alt dizilerin en küçük boyutu 2 olmalıdır. Dosyadan okunacak dizinin boyutu maksimum 10.000 olabilir.

### Örnek 1:

| input.txt                | output.txt |
|--------------------------|------------|
| -3 0 -10 5 -4 7 -1 6 1 0 | 17         |

### Örnek 2:

| input.txt               | output.txt |
|-------------------------|------------|
| 0 5 -4 0 -1 1 6 0 -8 20 | 5          |

### Soru 4)

Basit bir trafik simülatörü geliştirmeniz istenmektedir. Simülatörde araçlar çapı 1 br olan dairelerle temsil edilmektedir. Araçların peşe peşe dizilmiş yani konvoy halinde olduğu varsayılmaktadır. Örnek bir input.txt aşağıda verilmiştir. Dosyadaki 7 satır, 7 konvoy olduğunu göstermektedir. Her bir konvoy (satır) için sırasıyla araç sayısı, araçların rengi ve konvoydaki ilk aracın konum bilgisi verilmiştir. Örneğin ilk satırdaki 10 B -20 -30 ile konvoyun 10 araçlık olduğu, araçların renklerinin Beyaz (B) olduğu ve ilk aracın konumunun (-20, -30) olduğu bilgisi verilmektedir. İkinci ve sonraki araçların konumları belirlenirken ilk aracın konumunun X değeri dikkate alınacaktır. Eğer X değeri 0'dan küçükse örneğin -20 ise ikinci aracın konumu (-21, ..), üçüncünün (-22, ..), .. şeklinde devam edecektir. Eğer X değeri 0'dan büyükse örneğin 20 ise benzeri şekilde ikinci aracın konumu (21, ..), üçüncünün (22, ..), .. şeklinde devam edecektir. Y değerleri ilk araçla aynı olacaktır.

Araçların hareket yönleri öncelikle konumlarının X değerini 0'lamaya yönelik olarak +X veya -X eksenleri boyunca olacaktır. X'ler 0'lanınca araçlar +Y eksenini boyunca yollarına devam edeceklerdir. Araçlar 1br/sn hızla hareket edeceklerdir. Herhangi bir anda çarpışma ihtimali söz konusu olunca geçiş önceliği Y ekseninde hareket eden araca ait olacaktır. Bütün araçlar Y ekseninde hareket etmeye başladığında Y değeri en büyük olan en öndekinden en sondakine kadar araçların renkleri output.txt 'ye aşağıdaki formatta (renk adedi, peşine renk çiftleri halinde) yazılacaktır. K:Kırmızı, M:Mavi, B:Beyaz renkleri temsil etmektedir.

Örnek input.txt için output.txt'nin elde edilmesine dair video da ekte verilecektir. Çözmeniz için verilecek olan input.txt'de 1'er satır aralıklı verilmiş 6 farklı araç topolojisi vardır. İlk 5 topoloji 8'er konvoydan, 6. topoloji 10 konvoydan oluşmaktadır. Dolayısıyla output.txt 6 satırdan oluşacaktır.

| input.txt  | output.txt                                  |
|--|---|
| 10 B -20 -30<br>8 M 20 -25<br>10 B 10 -20<br>12 K -20 -15<br>10 M -20 -10<br>12 M 20 -5<br>20 K 10 0 | 15 K 5 M 10 B 5 K 5 M 10 B 3 M 7 K 17 M 5 K |