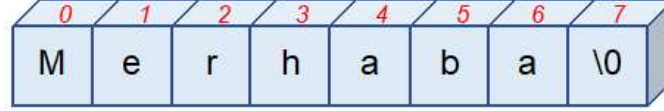


char kelime[] = "Merhaba";

Aslında yukarıdaki tanımlamada gördüğünüz üzere null karakteri bir katar sabitinin sonuna yerleştirmezsiniz. C++ derleyicisi diziyi oluşturduğunda '\0' değerini dizinin sonuna otomatik olarak ekler. Yukarıdaki tanımlama sonucu bellekte şu şekilde bir yerleşim söz konusu olacaktır.



C. Sonuç

Süre: 10 dk.

Materyal: [EK 4 Hafta 7 Süreli Ödev](#)

Hazırlık: EK 4 öğrencilere ÖYS ortamında süreli ödev olarak ders öncesinde açılmalıdır.

Ders içi uygulama: Dersin son 10 dk.'sında öğrencilere ödevin nasıl yapılacağı ve gönderileceği açıklanır. Gelecek hafta "Nesneler" konusunun işleneceği hakkında bilgi verilir.

Ders dışı uygulama (50 dk.): Bu uygulama asenkron olarak ÖYS üzerinden yürütülecektir. Öğrenciler ders sonunda istedikleri zaman başlatabilecekleri ancak süreli olacak şekilde tamamlayacakları görevleri ders dışında ödev olarak ÖYS'den yapacaktır. EK 4 ödev sırasında öğrencilere açılır. Ödevin süresi 50 dk. olarak belirlenir. Öğrenciler görevleri istedikleri sırada ve sayıda kendi tercihlerine bağlı olarak verilen süre içinde asenkron olarak uygulamaya başlar ve ÖYS ortamında ödev olarak gönderir. Bir görevi doğru yapan öğrenciye, o göreve ilişkin beceri rozeti ÖYS üzerinden atanacaktır. Ödevler iletildikten sonra eğitimci görevleri ve yanıtlarını GitHub üzerinden öğrencilere gönderir. Doğru yanıtların buradan kontrol edilmesi için Github ortamını gösterir. Doğru yanıtlanan görevler ile ilgili rozetler öğrenci hesabına gönderilir. Süreli ödevler ile öğrencilerin performanslarının değerlendirilmesi, kalıcı öğrenme ve transferin sağlanması amaçlanmaktadır.

Tasarlayıcı: Fadime, 10 arkadaşının Facebook beğeni sayılarının toplamını ve beğeni ortalamalarını merak etmektedir. Bu amaçla bir program yazmak ister, ancak yardıma ihtiyacı vardır. Fadime için bu programı sen tasarlar mısın?

```
#include <iostream>
```

```

using namespace std;

int main()
{
    int sayilar[] = {17, 13, 12, 9, 6, 11, 3, 14, 2, 19};
    int toplam = 0, n = 10;
    float ort;
    cout << "Dizi: " << endl;
    for (int i=0; i < n; i++) {
        cout << sayilar [i] << " ";
        toplam += sayilar[i];
    }
    cout << "\nDizinin toplami: " << toplam << endl;
    ort = (float)toplam / n;
    cout << "Dizinin ortalamasi: " << ort << endl;
    return 0;
}

```

Analizci: Arkadaşın (bilgisayar) aklından 1-9 arasında rastgele bir sayı tutar. Sen de tutulan bu sayıyı 3 tahminde bulmaya çalışan bir program yazıyorsun. Kural gereği tutulan sayıyı 3 tahminde bulamazsan oyun sona erer. Eğer 3 tahminden birinde sayıyı bulursan program tutulan sayıyı kaçınıcı tahmin hakkında bulduğunu ekrana yazdırır.

```

#include <iostream>
#include <ctime>
using namespace std;

int main()
{
    int sayi;
    int tahmin = -1;
    int tahmin_sayisi = 0;
    int tahmin_limiti = 3;
    bool outOfGuesses = false;
    srand(time(NULL));

    sayi = rand() % 9 + 1;
    cout << sayi;
}

```

```

while(tahmin != sayi && tahmin_sayisi < tahmin_limiti){
    cout << "Tahmininizi girin: ";
    cin >> tahmin;
    tahmin_sayisi++;
}
if(tahmin == sayi){
    cout << "Tebrikler, " << tahmin_sayisi << ". denemede kazandınız!" << endl;
} else {
    cout << "Uzgunum, 3 hakkınızda bilemediniz!" << endl;
}
return 0;
}

```

Kodlayıcı: Aşağıdaki matris yapısında voleybol oyuncularının numaraları verilmektedir. Koç, oyuncuları iki maç öncesi aşağıdaki gibi aynı sırada görmek istiyor. İki maçtada sıralamanın aynı olduğundan emin olmak için bir kod tasarlamayı düşünüyor, ancak yardıma ihtiyacı var. Koç için bu programı sen tasarlar mısın? Program içerisinde tanımlama bölümünde oyuncuların ilk dizilimi aşağıdaki matristeki gibi olmalıdır.

1. Maç Oyuncu Sırası:

1	3	5	7	9
11	13	15	17	19
2	4	6	8	10
12	14	16	18	20

2. Maç Oyuncu Sırası:

1	3	5	7	9
11	13	15	17	19
2	4	6	8	10
12	14	16	18	20

```
#include <iostream>
using namespace std;
int main()
{
    int matris1[4][5] = { {1, 3, 5, 7, 9},
                          {11, 13, 15, 17, 19},
                          {2, 4, 6, 8, 10},
                          {12, 14, 16, 18, 20}};

    int matris2[4][5] = { {1, 3, 5, 7, 9},
                          {11, 13, 15, 17, 19},
                          {2, 4, 6, 8, 10},
                          {12, 14, 16, 18, 20}};

    bool durum = true;
    int i, j;
    for (i = 0; i < 4; i++)
        for (j = 0; j < 5; j++)
            if (matris1[i][j] != matris2[i][j])
                durum = false;

    if (durum)
        cout << "Sıralama aynı";
    else
        cout << "Sıralama farklı";

    return 0;
}
```