

Ana programı da aşağıdaki gibi yazabiliriz. İlk olarak dizilerimizi tanımlıyoruz. Ardından dizi1'i fonksiyona gönderip en büyüğünü buluyoruz. Sonra dizi2'nin en büyük elemanını bulup ekrana yazdırıyoruz. Son olarak yapmak istediğimiz toplama işlemini gerçekleştiriyoruz.

```
int main()
{
    int dizi1[] = {5,3,4,5,8};
    int dizi2[] = {9,3,4,5,8};

    int en_buyuk_1 = en_buyuk(dizi1);
    cout << "1. dizinin en buyugu:" << en_buyuk_1 << endl;
    int en_buyuk_2 = en_buyuk(dizi2);
    cout << "2. dizinin en buyugu:" << en_buyuk_2 << endl;

    cout << en_buyuk_1 << "+" << en_buyuk_2 << "=" << en_buyuk_1 + en_buyuk_2;
}
```

Eğitmene Öneriler: Yukarıdaki etkinlikteki amaç öğrencilerin kod yazarken, fonksiyonları nerede, ne zaman, nasıl kullanılacağına yönelik onların farkındalığını artırmaktır. Yukarıdaki görevlerin tamamını öğrencilerin kendi bilgisayarlarında kodlanması beklenmektedir. Bunun için öğretmenlerin etkinlikler esnasında sürekli farklı öğrencilerinden isteyeceği ekran paylaşımı onların her etkinliğe hazır girmesi için motivasyonunun artmasını sağlayacaktır.

C. Sonuç

Süre: 10 dk.

Materyal: [Hafta 8 Süreli Ödev](#)

Hazırlık: EK 7 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 bildirilir.

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. "Hafta 8 Süreli Ödev" adlı dosya 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 eğitmen 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.

Süreli Ödev Yanıtlar: Görevler, öğrenciye verilecek beceri rozetleri ile isimlendirilmiştir. Her bir rozetin altında ilgili görevin yanıtları verilmektedir. Eğitmen bu yanıtları uygulama süresinin sonunda GitHub üzerinden öğrencilerle paylaşır.

1. **KODLAYICI:** Parametre olarak gönderilen iki sayının toplamını geriye döndüren bir fonksiyonu oluşturmak isteseydiniz nasıl bir kod yazardınız?

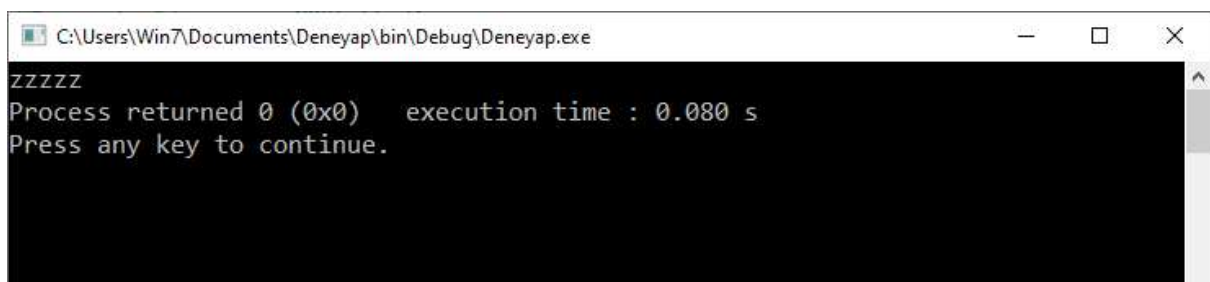
```
#include <iostream>

using namespace std;

int toplama(int sayi1, int sayi2)
{
    int toplam = sayi1 + sayi2;
    return toplam;
}

int main()
{
    int sonuc = toplama(5,4);
    cout<< sonuc;
}
```

2. **KODLAYICI:** Parametre olarak gönderilen harfi, yine parametre olarak gönderilen sayı kadar ekrana yazdıran bir fonksiyon tanımlamak isteseydiniz nasıl kodlardınız?
Örnek kullanım: tekrar_yaz('z',5)



Resim 29. Ekran çıktısı**Cevap:**

```
#include <iostream>

using namespace std;

void tekrar_yaz(char a, int adet)
{
    for(int i=0; i<adet; i++)
        cout << a;
}

int main()
{
    tekrar_yaz('z', 5);
}
```

3. **KODLAYICI:** Bir kullanıcının sosyal medyada paylaştığı fotoğraflardaki toplam beğeni sayılarının bir dizide saklandığı düşünülürse; paylaşılan fotoğraflardaki beğenilerin toplamını geriye döndüren bir fonksiyonu nasıl kodladınız?

Cevap:

```
#include <iostream>

using namespace std;

int dizi_topla(int dizi[5])
{
    int toplam = 0;

    for(int i=0; i<5; i++)
        toplam = toplam + dizi[i];

    return toplam;
}

int main()
{
    int sayilar[5] = {5, 6, 9, 3, 2};
    int sonuc = dizi_topla(sayilar);
}
```

```
cout << sonuc;  
}
```