

ISPARTA UYGULAMALI BİLİMLER ÜNİVERSİTESİ - TEKNOLOJİ FAKÜLTESİ BİLGİSAYAR MÜHENDİSLİĞİ
BÖLÜMÜ

2024-2025 BAHAR DÖNEMİ BLG-102 VERİ YAPILARI DERSİ ÖDEV RAPORU

Ad Soyad: Ceren Mıtırık

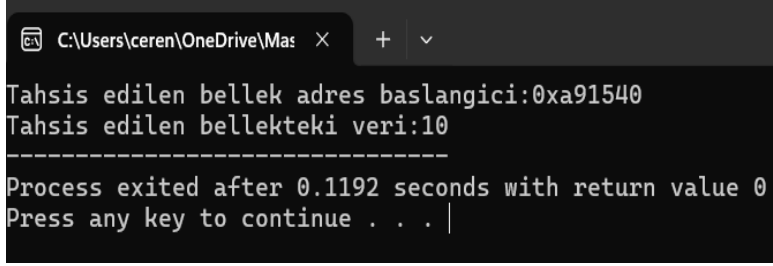
Numara: 2212721032

(Bu bölüme ödev numarasını ve ödevi yazın.)

ÖDEV:

Örnek 1:

```
#include <stdio.h>
#include <iostream>
#include <conio.h>
#include <cstdlib>
using namespace std;
int main()
{
    int *ip; // işaretçi dinamik olarak ayrılan bellek bölgesinin adresini tutacak
    ip = (int*) malloc(sizeof(int)); // malloc fonksiyonu ile int boyutunda bir bellek bölgesi ayır
    // int* ile tür dönüşümü yap. Ayrılan belleğin adresi ip işaretçisine ata
    *ip = 10; // Ayrılan bellek bölgesine 10 değerini ata. *ip, ip'nin gösterdiği bellek adresindeki DEĞERİ ifade eder.
    cout << "Tahsis edilen bellek adres baslangici:" << ip; // başlangıç ADRESİNİ yazdır
    cout << "\nTahsis edilen bellekteki veri:" << *ip; // Ayrılan bellekteki veri yani 10'u yazdır
    free(ip); // malloc ile ayrılan bellek bölgesi free fonksiyonu ile serbest bırak
}
```

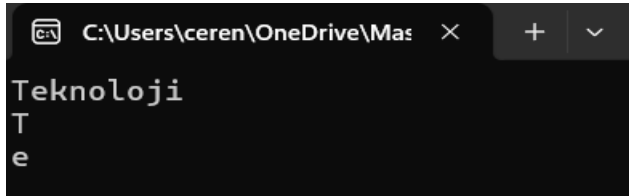


```
C:\Users\ceren\OneDrive\Mas x + v
Tahsis edilen bellek adres baslangici:0xa91540
Tahsis edilen bellekteki veri:10
-----
Process exited after 0.1192 seconds with return value 0
Press any key to continue . . . |
```

ÖRNEK 2:

```
#include <stdio.h>
#include <iostream>
#include <conio.h>
#include <cstdlib>
#include <cstring>
using namespace std;
int main()
{
    char *cp = "Teknoloji"; // karakter işaretçisi tanımla ve "Teknoloji" başlangıç adresine ata
    cout << cp << endl; // cp'nin gösterdiği string yani Teknoloji yazdır
    cout << cp[0] << endl; // string'in ilk karakterini yazdır
    cout << cp[1] << endl; // string'in ikinci karakterini yazdır
    free(cp); // String statik bellek bölgesinde saklanır ve free ile serbest bırakılmaz ??

    char *cpl;
    cpl = (char*) malloc(40); // 40 byte'lık dinamik bellek ayır ve adresi cpl'ye ata
    if (!cpl) { cout << "Bellek tahsis hatası!"; exit(1); } // ayrılan belleğin başarılı olup olmadığı kontrol et sonuç eğer Başarısız ise hata mesajı ver ve programı sonlandır
    // cpl = "Bilgisayar"; // hata
    // strcpy(cpl, "Bilgisayar"); // Doğru kullanım strcpy fonksiyonu ile "Bilgisayar" string'i cpl'nin gösterdiği dinamik belleğe kopyala ??
    strcpy(cpl, "Bilgisayar");
    cout << cpl << endl; // gösterdiği stringi yazdır
    cout << cpl[0] << endl; // gösterdiği string'in ilk karakterini yazdır
    cout << cpl[1] << endl; // string'in ikinci karakterini yazdır
    free(cpl); // malloc ile ayrılan bellek bölgesi free ile serbest bırak
}
// çıktıda BİLGİSAYAR ın çıktısını bu kod vermiyor |
```



```
C:\Users\ceren\OneDrive\Mas x + v
Teknoloji
T
e
```

ÖRNEK 3:

```
int main(){
int *ip; // işaretçi dinamik olarak ayrılan bellek bölgesinin adresini tutacak
ip = (int*) calloc(5, sizeof(int)); // calloc fonksiyonu ile 5 adet int boyutunda bellek bölgesi ayır ve (int*) ile tür dönüşümü yap.
// Ayrılan belleğin adresi ip işaretçisine ata

for (int i = 0; i < 5; i++) {
*(ip + i) = (i + 1) * 10; // 10, 20, 30, 40, 50 değerleri atanır callocla 0 la başlar çünkü
// (*(ip + i), ip'nin gösterdiği bellek adresinden i kadar ilerleyerek o adresteki değeri ifade eder.)??
}

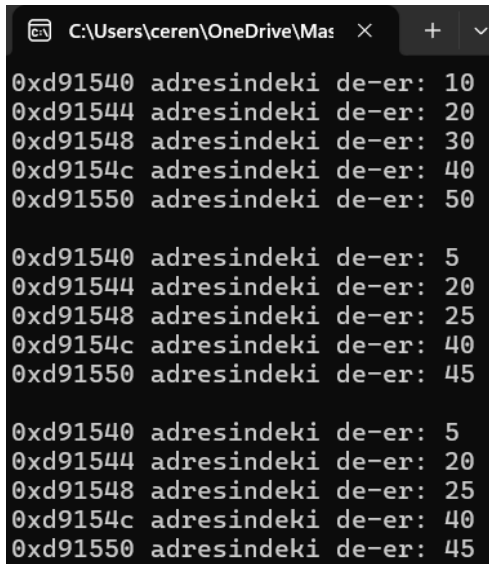
for (int i = 0; i < 5; i++) {
cout << (ip + i) << " adresindeki değer: " << *(ip + i) << endl; // adresi ve değeri yazdır
}
cout << endl;

ip[0] = 5; // ilk elemana 5 ata
ip[2] = 25; // üçüncü elemanına 25 ata
ip[4] = 45; // beşinci elemanına 45 ata

for (int i = 0; i < 5; i++) {
cout << (ip + i) << " adresindeki değer: " << *(ip + i) << endl; // adresi ve yeni değeri yazdır
}
cout << endl;

for (int i = 0; i < 5; i++) {
cout << (ip + i) << " adresindeki değer: " << ip[i] << endl; // adresi ve değeri bu sefer (ip[i] kullanarak) yazdır
}

free(ip); // serbest bırak
```



```
C:\Users\ceren\OneDrive\Mas x + v

0xd91540 adresindeki de-er: 10
0xd91544 adresindeki de-er: 20
0xd91548 adresindeki de-er: 30
0xd9154c adresindeki de-er: 40
0xd91550 adresindeki de-er: 50

0xd91540 adresindeki de-er: 5
0xd91544 adresindeki de-er: 20
0xd91548 adresindeki de-er: 25
0xd9154c adresindeki de-er: 40
0xd91550 adresindeki de-er: 45

0xd91540 adresindeki de-er: 5
0xd91544 adresindeki de-er: 20
0xd91548 adresindeki de-er: 25
0xd9154c adresindeki de-er: 40
0xd91550 adresindeki de-er: 45
```

ÖRNEK 4 :

```
int main(){
int *ip; // işaretçi ayrılan bellek bölgesinin adresini tutsun
ip = (int*) calloc(5, sizeof(int)); // 5 tane (int) boyutunda bellek bölgesi ayır ve.(int*) ile tür dönüşümü yap
// Ayrılan belleğin adresi ip işaretçisine ata
for (int i = 0; i < 5; i++) {
*(ip + i) = (i + 1) * 10; // 10, 20, 30, 40, 50 değerleri ata (*(ip + i), ip'nin gösterdiği bellek adresinden i kadar ilerleyerek o adresteki değeri ifade eder.?)
}
for (int i = 0; i < 5; i++) {
cout << (ip + i) << " adresindeki değer: " << *(ip + i) << endl; // adresi ve değeri yazdır
}
cout << endl;

ip = (int*) realloc(ip, 10 * sizeof(int)); // realloc fonksiyonu ile ip'nin gösterdiği bellek bölgesinin boyutu 10 tamsayıya çıkar
// realloc, mevcut verileri koruyarak yeni boyutta bellek ayırır.
printf("Genişletilmiş bellek değerleri:\n");
for (int i = 5; i < 10; i++) { // genişletilmiş
*(ip + i) = i + 1; // Genişletilen bellek bölgesinin her elemanına 6, 7, 8, 9, 10 değerlerini ata
}
for (int i = 0; i < 10; i++) {
cout << (ip + i) << " adresindeki değer: " << *(ip + i) << endl; // eski ve genişletilen adresi ve değeri yazdır
}

free(ip);
```

```
C:\Users\ceren\OneDrive\Mas  X + v
0xc41540 adresindeki de-er: 10
0xc41544 adresindeki de-er: 20
0xc41548 adresindeki de-er: 30
0xc4154c adresindeki de-er: 40
0xc41550 adresindeki de-er: 50

Geniřletilmiř bellek de-erleri:
0xc41540 adresindeki de-er: 10
0xc41544 adresindeki de-er: 20
0xc41548 adresindeki de-er: 30
0xc4154c adresindeki de-er: 40
0xc41550 adresindeki de-er: 50
0xc41554 adresindeki de-er: 6
0xc41558 adresindeki de-er: 7
0xc4155c adresindeki de-er: 8
0xc41560 adresindeki de-er: 9
0xc41564 adresindeki de-er: 10
```

ÖRNEK 5:

```
int main(){
char *pdizisatir[5]; // 5 elemanlı char işaretçi dizisi tanımla dizi, her biri dinamik olarak ayrılmış bir string'in adresini tutsun
int id;

for (id = 0; id < 5; id++) {
pdizisatir[id] = (char*)malloc(20 * sizeof(char)); // Her dizi elemanı için 20 karakterlik dinamik bellek ayır
// (char*) ile tür dönüşümü yap. ayrılan adres pdizisatir dizisindeki elemana ata
}

for (id = 0; id < 5; id++) {
strcpy(pdizisatir[id], "Karakter dizisi"); // "Karakter dizisi" string'i, pdizisatir dizisindeki her elemanın gösterdiği dinamik belleğe kopyala
}

for (id = 0; id < 5; id++) {
cout << pdizisatir[id] << endl; // pdizisatir dizisinin her elemanının gösterdiği stringi yazdır
}

for (id = 0; id < 5; id++) {
free(pdizisatir[id]); // serbest bırak
}
```

```
C:\Users\ceren\OneDrive\Mas  X + v
Karakter dizisi
Karakter dizisi
Karakter dizisi
Karakter dizisi
Karakter dizisi
=====
```

ÖRNEK 6:

```
int main(){
int *ip; // işaretçi ayrılan bellek bölgesinin adresini tutsun
ip = new int; // int boyutunda dinamik bellek ayır ve ayrılan belleğin adresi ip işaretçisine ata
if (!ip) { // bellek ayırımı Başarısız ise hata mesajı ver ve programı sonlandır
cout << "Bellek tahsis hatası!";
exit(1);
}

// ip = new int(21); // 22. satır yerine bu kullanım ile int değeri büyüklüğünde bellek tahsis ederek bellek adresine 21 değerini atar.

*ip = 21; // 20. satır yerine işaretçi yoluyla belleğe 21 değerini atar.
cout << ip << " bellek adresindeki değeri: " << *ip; // Ayrılan belleğin adresini ve değerini yazdır
delete ip; //serbest bırak
```

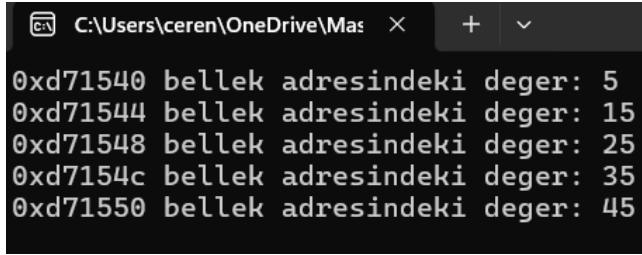
```
C:\Users\ceren\OneDrive\Mas  X + v
0xd41540 bellek adresindeki de-er: 21
=====
```

ÖRNEK 7:

```
int main(){
int *ip, id; // int işaretçi ve bir int değişkeni tanımla
ip = new int[5]; // dizi için dinamik bellek ayırır ve ayrılan belleğin adresi ip işaretçisine ata
if (!ip) { //başarılı mı kontrol et
    cout << "Bellek tahsis hatası!";
    exit(1);
}
ip[0] = 5;
ip[1] = 15;
ip[2] = 25;
ip[3] = 35;
ip[4] = 45;

for (id = 0; id < 5; id++) {
    cout << &ip[id] << " bellek adresindeki deger: " << ip[id] << endl; // Dizideki her elemanın adresini ve değerini yazdır
}

delete [] ip; // dinamik belleği serbest bırak
}
```



```
C:\Users\ceren\OneDrive\Mas
0xd71540 bellek adresindeki deger: 5
0xd71544 bellek adresindeki deger: 15
0xd71548 bellek adresindeki deger: 25
0xd7154c bellek adresindeki deger: 35
0xd71550 bellek adresindeki deger: 45
```

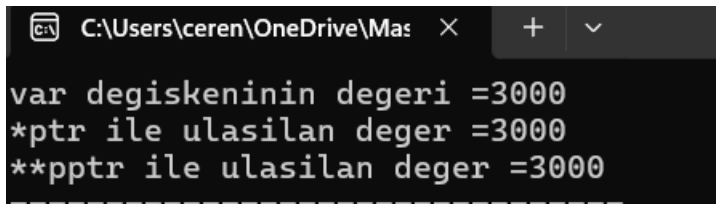
ÖRNEK 8 :

```
using namespace std;
int main(){
int var;
int *ptr; // işaretçi tamsayı değişkeninin adresini tutar
int **pptr; // işaretçi tamsayı işaretçisinin adresini tutar

var = 3000;

/* var değişkeninin adresini alıyoruz */
ptr = &var; // var değişkeninin adresini ptr işaretçisine ata
/* & kullanarak ptr işaretçisinin adresini alıyoruz */
pptr = &ptr; // ptr işaretçisinin adresini pptr işaretçisine ata

cout << "var degiskeninin degeri =" << var << endl;
cout << "*ptr ile ulasilan deger =" << *ptr << endl; // ptr işaretçisinin gösterdiği adresteki değeri yazdır
cout << "**pptr ile ulasilan deger =" << **pptr; // pptr işaretçisinin gösterdiği adresteki işaretçinin gösterdiği adresteki değeri yazdır
//[ ** operatörü, işaretçinin işaretçisinin gösterdiği adresteki değeri almak için kullanılır.]
```



```
C:\Users\ceren\OneDrive\Mas
var degiskeninin degeri =3000
*ptr ile ulasilan deger =3000
**pptr ile ulasilan deger =3000
```

(Bu bölüme sıradaki ödev numarasını ve ödevi yazın.)

ÖDEV: