

BLM-112 PROGRAMLAMA DİLLERİ II

Ders-4 İşaretçiler (Pointer) (Kısım-3)

Yrd. Doç. Dr. Ümit ATILA

umitatila@karabuk.edu.tr

<http://web.karabuk.edu.tr/umitatilla/>

Fonksiyon Gösteren Pointer

- Pointer bir fonksiyonun adresini tutabilir.
- Bir fonksiyonun ismi dizilerde olduğu gibi fonksiyonun işlevini yerine getiren kodun hafızada tutulduğu başlangıç adresini gösterir.
- `int (*fPtr) (int,int)`
 - Burada iki tamsayı parametre alan ve geriye tamsayı döndüren bir fonksiyonu gösteren `fPtr` isiminde pointer tanımlanmıştır.
- `int *fPtr (int,int)`
 - Burada ise `fPtr` isiminde iki tamsayı parametre alan ve geriye tamsayı pointer döndüren `fPtr` isimli bir fonksiyon tanımlanmıştır.

Fonksiyon Gösteren Pointer

```
1 #include <stdio.h>
2 int kare(int);
3 int kup(int);
4 int main(void)
5 {
6     /* bir int değer alıp geriye int değer gönderen bir fonksiyonun adresi */
7     int (*islem)(int);
8     int i;
9     char c;
10
11     printf("1-karealani\n2-kup hacmi\n ");
12     c = getchar();
13     printf("\nSayıyı gir : ");
14     scanf("%d", &i);
15     if (c == '1')
16         islem = kare; /* kare işlevinin adresi islem değişkenine kopyalanır */
17     else
18         islem = kup;
19     printf("Sonuc = %d\n", islem(i));
20     while( getchar() != '\n' ) { /*do nothing*/ } ;
21     getchar() ; /* wait */
22 }
23 int kare(int s)
24 {
25     return s*s;
26 }
27 int kup(int s)
28 {
29     return s*s*s;
30 }
```

Void Pointer'lar

- İşaretçilerin tipi void olabilir.
- Bu durumda işaretçinin gösterdiği adresteki değere erişirken veri tipini belirtmeliyiz.

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4 int main(void)
5 {
6     void *a;
7     a = (char*) malloc(6);
8     strcpy((char *)a, "12345");
9     printf("%s\n", a);
10    free(a);
11    a = (double*) malloc(sizeof(double));
12    /* değere erişirken veri tipi belirt */
13    *(double*)a = 3.123;
14    printf("%f\n", *(double *)a);
15    getchar();
16 }
```

Örnek-1

- Prototipi aşağıda verilen ve 5 elemanlı tamsayı dizinin elemanları toplamını pointer kullanarak bularak döndüren fonksiyonu yazınız.

int toplamBul(int *a)

```
#include <stdio.h>

void main()
{
    int sonuc;
    int dizi[5] = {10,20,30,40,50};
    sonuc = toplamBul(dizi);
    printf("Toplam = %d", sonuc);
}

int toplamBul(int *a)
{
    int i, toplam=0;
    for(i=0;i<5;i++)
        toplam += *(a+i);
    return toplam;
}
```

Örnek-2

- İki değişkenin değerini işaretçi kullanarak birbirisiyle değiştiren fonksiyonu aşağıda verilen prototipe göre yazınız.

void degistir(int *,int *)

```
#include <stdio.h>
void swap(int * q,int * p)
{
    int temp = *p;
    *p = *q;
    *q = temp;
}

int main()
{
    int a = 10, b = 2, x = 3, y = 5;
    printf("a,b,x,y: %d,%d,%d,%d\n", a, b, x, y);
    swap(&x, &y);
    swap(&a, &b);
    printf("a,b,x,y: %d,%d,%d,%d\n", a, b, x, y);
}
```

Örnek-3

- Bir dikdörtgenin alanı ve çevresini hesaplayan fonksiyonu aşağıda verilen prototipe göre yazınız.

void dortgen (int a,int b, int *alan, int *cevre)

```
#include <stdio.h>
void dortgen(int a, int b, int *alan, int *cevresi);

int main()
{
    int x, y;
    int alan, cevresi;
    printf("Boşlukla ayrılmış iki değer giriniz: " );
    scanf("%d %d", &x, &y);
    dortgen(x, y, &alan, &cevresi);
    printf("Alanı %d ve çevresi %d dir\n", alan, cevresi);
}

void dortgen(int a,int b, int *alan,int *cevresi)
{
    *alan = a * b;
    *cevresi = 2 * (a + b);
}
```

Örnek-4

- Strlen fonksiyonu bir string'in uzunluğunu verir. Bu fonksiyonu aşağıda verilen prototipe göre yazınız.

int strlen (char *)

```
int uzunluk(char *p)
{
    int n=0;
    while(*(p+n) != '\0')
    {
        n++;
    }
    return n;
}
```

```
#include <stdio.h>

int uzunluk(char *);

void main()
{
    char dizi[50];
    printf("String gir:");
    gets(dizi);

    printf("String uzunlugu: %d", uzunluk(dizi));
}
```


Örnek-5

- Strchr fonksiyonu bir string içerisinde belli bir karakteri arar. İlk bulunduğu yerin indisini döndürür. Bulamaz ise -1 döndürür. Bu fonksiyonu aşağıda verilen prototipe göre yazınız.

`int strchr (char *, char)`

```
#include <stdio.h>
void main()
{
    int sonuc;
    char dizi[50], aranan;
    printf("String gir:");
    gets(dizi);

    printf("Karakter gir:");
    scanf("%c", &aranan);

    sonuc = karakterAra(dizi, aranan);
    if(sonuc == -1)
        printf("Aradiginiz bulunamadi...");
    else
        printf("Aradiginiz %c karakteri %d indisinde bulundu...", aranan, sonuc);
}
```

```
int karakterAra(char *p, char c)
{
    int i=0;
    while(*(p+i) != '\0')
    {
        if(*(p+i) == c)
            return i;
        i++;
    }
    return -1;
}
```

Örnek-6

- Verilen işaretsiz değişkeni binary sayıya çeviren fonksiyonu aşağıda verilen prototipe göre yazınız.

void convertToBinary (unsigned)

```
#include <stdio.h>
#include <conio.h>

void convertToBinary(unsigned);

int main()
{
    unsigned sayi;
    printf("Enter number");
    scanf("%u", &sayi);
    convertToBinary(sayi);

    getch();
}
```

```
void convertToBinary (unsigned x)
{
    int i=0,k;
    unsigned *p;
    p = (unsigned *)malloc(sizeof(unsigned));
    *p = x;
    while(1)
    {
        *(p+i) = x % 2;
        x = x/2;

        if(x == 1)
        {
            p = (unsigned *) realloc(p, (i+1)*sizeof(unsigned));
            *(p+i+1) = 1;
            break;
        }
        i++;
        p = (unsigned *) realloc(p, (i+1)*sizeof(unsigned));
    }
    for(k=i+1;k>=0;k--)
        printf("%d", *(p+k));
}
```

Örnek-7

- Klavyeden ENTER tuşuna basılana kadar karakter okuyan, okuduğu her karakter yerine ekrana * karakterini yazdıran, ENTER girildiğinde ise o ana kadar girilmiş tüm karakterleri ekrana yazdıran programı yazınız.
- ENTER tuşunun karakter kodu 13 tür.

```
#include <stdio.h>
#include <stdlib.h>
void main()
{
    char *p;
    int i=0, k;

    p = (char *) malloc(sizeof(char));
    while(1)
    {
        *(p+i) = getch();
        if(*(p+i) == 13) break;
        putchar('*');
        i++;
        p = (char *) realloc(p, (i+1)*sizeof(char));
    }

    putchar('\n');
    for(k=0;k<i;k++)
        putchar(*(p+k));
}
```

Örnek-8

- Her bir elemanı bir ay isminin adresini gösteren dört tane mevsim dizileri oluşturunuz.
- Oluşturduğunuz mevsim dizilerini tutan **table** isminde başka bir dizi oluşturunuz.
- Dizilere table dizisini kullanarak erişiniz ve elemanları yazdırınız.

```
1  #include <stdio.h>
2  int main()
3  {
4      int i,j;
5      char * ilkBaharAylar[3] ={"Mart","Nisan","Mayis"};
6      char * yazAylar[3] ={"Haziran","Temmuz","Agustos"};
7      char * sonBaharAylar[3] ={"Eylul","Ekim","Kasim"};
8      char * kisAylar[3] ={"Aralik","Ocak","Subat"};
9
10     char ** table[4]; //char pointer(string) tutan dizileri tutan dizi
11     table[0] = ilkBaharAylar;
12     table[1] = yazAylar;
13     table[2] = sonBaharAylar;
14     table[3] = kisAylar;
15
16     for(i=0;i<4;i++)
17     {
18         for(j=0;j<3;j++)
19         {
20             printf("%s\n",table[i][j]);
21         }
22     }
23
24     getchar();
25     return 0;
26 }
```

Kaynaklar

- Paul J. Deitel, "C How to Program", Harvey Deitel.
- Kaan Aslan, "A'dan Z'ye C Klavuzu 8. Basım", Pusula Yayıncılık, 2002.