

# BLM 112- Programlama Dilleri II

## Hafta 6

### İşaretçiler (Pointers) Örnekler

Dr. Öğr. Üyesi Caner Özcan

# Malloc() Fonksiyonu

- ▶ Malloc fonksiyonu bir değişken için hafızadan bir blok yer ayrılması için kullanılır.
- ▶ Eğer hafızada yeterli alan yoksa fonksiyon NULL döndürür.

```
int *ptr;
```

```
ptr = (int *) malloc(n*sizeof(int));
```



# Calloc() Fonksiyonu

- ▶ Calloc fonksiyonu da hafıza bloğu almak için kullanılabilir.
- ▶ Eğer hafızada yeterli alan yoksa fonksiyon NULL döndürür.
- ▶ Malloc fonksiyonundan farklı olarak argüman listesi değişmektedir.

```
char *ptr;
```

```
ptr = (char *)calloc(10, sizeof(char));
```

# Realloc() Fonksiyonu

- ▶ Realloc fonksiyonu hafızadan ayrılan bir alanı yeniden boyutlandırmak için kullanılır.
- ▶ Tekrar ayarlanacak hafıza alanının başlangıcını işaret edecek bir pointer ve yeni boyut bilgisini parametre olarak alır.

```
void *realloc(void *ptr, size_t size);
```



# Free() Fonksiyonu

- ▶ Büyük boyutta ve kapsamlı bir program söz konusuysa, efektif bellek yönetiminin ne kadar önemli olduğunu daha iyi anlarsınız.
- ▶ Gereksiz tüketilen bellekten kaçınmak gerekmektedir.
- ▶ Malloc ve calloc fonksiyonlarıyla hafızadan ayrılan alanın tekrar heap alanına serbest bırakılması için kullanılır.

```
int *ptr;  
ptr = (int *) malloc(n*sizeof(int));  
free(ptr);
```



# Dizilerde Dinamik Bellek Kullanımı

```
1 #include<stdio.h>
2 #include<stdlib.h>
3 int main( void )
4 {
5     // Dinamik bir dizi oluşturmak için pointer kullanırız.
6     int *dizi;
7     // Dizimizin kaç elemanlı olacağını eleman_sayisi isimli değiskende tutuyoruz.
8     int eleman_sayisi;
9     int i;
10    printf( "Eleman sayısını giriniz> ");
11    scanf( "%d", &eleman_sayisi );
12    // malloc( ) fonksiyonuyla dinamik olarak dizimizi istediğimiz boyutta oluşturalım.
13    dizi = (int *)malloc( eleman_sayisi * sizeof( int ) );
14    //dizi = (int *)calloc( eleman_sayisi, sizeof( int ) );
15
16    for( i = 0; i < eleman_sayisi; i++ )
17        printf( "Adres:%d\tDeğer:%d\n", &dizi[i],dizi[i] );
18
19    // hafızadan temizleme
20    free( dizi );
21
22    while( getchar() != '\n' ) { /*do nothing*/ } ;
23        getchar() ; /* wait */
24    return 0;
25 }
```

# Dizilerde Dinamik Bellek Kullanımı

```

1 #include<stdio.h>
2 #include<stdlib.h>
3 int main( void )
4 {
5     // Dinamik bir dizi oluşturmak için pointer kullanırız.
6     int *dizi;
7     // Dizimizin kaç elemanlı olacağını eleman_sayisi isimli degiskende tutuyoruz.
8     int eleman_sayisi;
9     int i;
10    printf( "Eleman sayısını giriniz> ");
11    scanf( "%d", &eleman_sayisi );
12    // malloc( ) fonksiyonuyla dinamik olarak dizimizi istedigimiz boyutta oluşturalım.
13    dizi = (int *)malloc( eleman_sayisi * sizeof( int ) );
14    //dizi = (int *)calloc( eleman_sayisi, sizeof( int ) );
15
16    for( i = 0; i < eleman_sayisi; i++)
17        printf( "Adres:%d\tDeger:%d\n", dizi[i], dizi[i] );
18
19    // hafizadan temizleme
20    free( dizi );
21
22    while( getchar() != '\n' )
23        getchar(); /* wait for next input */
24    return 0;
25 }

```

D:\Akademik\2016\Programlama Dilleri 2\Prog2\_Le

```

Eleman sayisini giriniz> 5
Adres:201968 Deger:209360
Adres:201972 Deger:196800
Adres:201976 Deger:1836008284
Adres:201980 Deger:929984365
Adres:201984 Deger:1869567068

```

# Fonksiyon Pointer'lar

- ▶ Pointer'lar bir fonksiyonun tutulduğu adresi gösterebilir.
- ▶ Normal pointer gibi fonksiyonun adresini içeren değişken tanımı yapılmalıdır.

**int (\*fPtr) (int,int)**

Bu tanımda fPtr değişkeni iki tane int tipinde parametre alan ve geriye int tipinde değer döndüren bir fonksiyonun adresini içerir.

**int \*fPtr (int,int)**

Bu tanımda ise iki tane int tipinde parametre alan ve geriye int pointer döndüren bir fonksiyon tanımlanmıştır.



# Fonksiyon Pointer'lar

```
1  #include <stdio.h>
2  int kare(int);
3  int kup(int);
4  int main(void)
5  {
6      /* bir int deęer alıp geriye int deęer 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 deęişkenine kopyalanır */
17     else
18         islem = kup;
19     printf("Sonuç = %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 }
```

# Fonksiyon Pointer'lar

```

1  #include <stdio.h>
2  int kare(int);
3  int kup(int);
4  int main(void)
5  {
6      /* bir int deęer alıp geriye int deęer 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 deęişkenine
17     else
18         islem = kup;
19     printf("Sonuç = %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 }

```

D:\Akademik\2016\Program

```

1-karealani
2-kup hacmi
2
Sayıyı gir : 5
Sonuç = 125

```

D:\Akademik\2016\Progra

```

1-karealani
2-kup hacmi
1
Sayıyı gir : 3
Sonuç = 9

```

# Void Pointer'lar

- ▶ Pointer'lar void olarak tanımlanabilir.
- ▶ Bu şekilde tanımlanan pointerların gösterdiği adresteki değere erişmek için veri tipi belirtilmelidir.

```
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

- ▶ ENTER tuşuna basılana kadar sürekli olarak klavyeden girilen karakterleri alan ve her bir karakter için ekrana '\*' basan bir program yazınız.
- ▶ Kullanıcı ENTER tuşuna bastığı zaman girilen karakterleri girildiği sırada ekrana yazdırınız. ENTER için karakter kodu 13'tür.

# Örnek 1

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

int main(){
    char giris[50];
    char *p;
    int i=0,k;
    p=giris;
    while(1){
        *(p+i)=getch();
        if(*(p+i)==13)
            break;
        putchar('*');
        printf("Adres[%d]: %d\n",i, p+i);
        i++;
    }
    printf("\n");
    for(k=0;k<i;k++){
        printf("Adres[%d]: %d\n",k, p+k);
        putchar(*(p+k));
    }
    getch();
    return 0;
}
```

# Örnek 1

```
#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 2

- ▶ Aşağıda verilen prototipi kullanarak iki değişkenin değerini yer değiştiren bir fonksiyon yazınız.

```
void swap (int *, int *)
```

## Örnek 2

- İki değişkenin değerini yer değiştiren bir fonksiyon yazınız.

```
#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

- ▶ Aşağıda verilen prototipi kullanarak bir dikdörtgenin çevresi ve alanını hesaplayan bir fonksiyon yazınız.

```
void rectangle(int a,int b, int *area, int *perimeter)
```

# Örnek 3

- Bir dikdörtgenin çevresi ve alanını hesaplayan bir fonksiyon yazınız.

```
#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

- ▶ Aşağıda verilen prototipi kullanarak strlen fonksiyonuna benzer bir fonksiyon yazınız.

int uzunluk(char \*)

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

int uzunluk(char *);

int main()
{
    char str[100];
    printf("Enter string");
    gets(str);
    printf("Length of string : %d", uzunluk(str));
    getch();
}
```

# Örnek 4

- ▶ Aşağıda verilen prototipi kullanarak strlen fonksiyonuna benzer bir fonksiyon yazınız.

int uzunluk(char \*)

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

int uzunluk(char *);

int main()
{
    char str[100];
    printf("Enter string");
    gets(str);
    printf("Length of string : %d", uzunluk(str));
    getch();
}
```

```
int uzunluk(char * p)
{
    int n =0;

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

# Örnek 5

- ▶ Aşağıda verilen prototipi kullanarak verilen bir katarıda bir karakter arayan bir fonksiyon yazınız.

char \* ara (char \*, char)

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

char *ara(char *, char);

int main() {
    char *sonuc;
    char aranan;
    char str[100];
    printf("Enter string:\n");
    gets(str);
    printf("Enter character to search:\n");
    aranan = getchar();
    printf("Aranan: %c\n", aranan);
    printf("Aranan: %d\n", aranan);
    sonuc = ara(str, aranan);
    if(sonuc==NULL)
        printf("Character not found\n");
    else
        printf("Character found\n");
    getch();
}
```

# Örnek 5

- Aşağıda verilen prototipi kullanarak verilen bir katarde bir karakter arayan bir fonksiyon yazınız.

```
char * ara (char *, char)
```

```
char *ara(char *p, char c){
    while(*p != '\0'){
        if(*p==c)
            return p;
        p++;
    }
    if(c=='\0')
        return p;
    return NULL;
}
```

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

```
char *ara(char *, char);
```

```
int main() {
    char *sonuc;
    char aranan;
    char str[100];
    printf("Enter string:\n");
    gets(str);
    printf("Enter character to search:\n");
    aranan = getch();
    printf("Aranan: %c\n", aranan);
    printf("Aranan: %d\n", aranan);
    sonuc = ara(str, aranan);
    if(sonuc==NULL)
        printf("Character not found\n");
    else
        printf("Character found\n");
    getch();
}
```

# Örnek 5

- Aşağıda verilen prototipi kullanarak verilen bir katarıda bir karakter arayan bir fonksiyon yazınız.

```
#include<stdio.h>
C:\Use<conio.h>
```

```
Enter string:
Today weather is good.
Enter character to search:
i
Aranan: i
Aranan: 105
Character found
```

```
while(*p != '\0'){
    if(*p==c)
        return p;
    p++;
}
if(c=='\0')
    return p;
return NULL;
}
```

```
ara(char *, char);
```

```
ara() {
    char *sonuc;
    char aranan;
    char str[100];
    printf("Enter string:\n");
```

```
gets(str);
```

```
printf("C:");
```

```
ara
```

```
printf("Enter string:");
```

```
printf("today weather is good.");
```

```
printf("Enter character to search:");
```

```
sonuc = ara(c);
if(sonuc != NULL)
```

```
printf("Aranan: %c", *sonuc);
```

```
printf("Aranan: %d", sonuc - str);
```

```
else
```

```
printf("Character not found");
```

```
getch();
}
```

# Örnek 6

- ▶ Unsigned integer değeri binary değere çeviren bir fonksiyon yazınız.

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

void convertToBinary(unsigned);

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

    getch();
}
```



# Örnek 6

- ▶ Unsigned integer değeri binary değere çeviren bir fonksiyon yazınız.

```
#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=1, k;
    unsigned *p;
    p = &x;

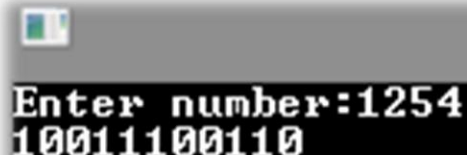
    while(1)
    {
        *(p+i) = x%2;
        x = x/2;

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

# Örnek 6

```
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));
}
```



Enter number:1254  
10011100110

# Çalışma Soruları

1)

```

  1
 1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1

```

Pascal Üçgeninin ilk 6 satırı yanda verilmiştir. Kullanıcının girdiği N değeri için pascal üçgeninin ilk N satırını alt alta ekrana yazdıran, algoritmayı çiziniz. Not: Çözümde sadece tek bir dizi kullanılmalıdır.

2)  $\cos(X) = 1 - x^2/2! + x^4/4! + \dots + x^N/N!$  şeklinde tanımlanmıştır. Kullanıcının girdiği X ve N için  $\cos(X)$ 'i hesaplayan algoritmayı çiziniz.

3) Bir sayı dizisinin ardışık elemanlarının arasındaki mutlak değerce en büyük farkı ve yerini bulan algoritmayı çiziniz.

4)

X		X		X	
	X		X		X
X		X		X	
	X		X		X
X		X		X	
	X		X		X

Kullanıcının girdiği N\*N'lik bir matrisin sadece yandaki şekildeki çarpı işaretli hücrelerinin toplamını bulup ekrana yazdıran, algoritmayı çiziniz.

5) Bir dizideki çift sayılara başa, tekleri sona başka bir dizi kullanmadan atan algoritmayı çiziniz.

Örnek:

Giriş dizisi: 5 7 2 9 5 3 8 6

Çıkış dizisi: 2 8 6 3 5 9 7 5

# Gelecek Hafta

- ▶ Struct, Enum ve Typedef
- ▶ Tek Bağlı Doğrusal Listeler

# Kaynaklar

- ▶ Doç. Dr. Fahri Vatansever, “Algoritma Geliştirme ve Programlamaya Giriş”, Seçkin Yayıncılık, 12. Baskı, 2015.
- ▶ Kaan Aslan, “A’dan Z’ye C Klavuzu 8. Basım”, Pusula Yayıncılık, 2002.
- ▶ Paul J. Deitel, “C How to Program”, Harvey Deitel.
- ▶ “A book on C”, All Kelley, İra Pohl

S o r u l a r  
?



Dinlediğiniz için teşekkürler

CANER ÖZCAN



canerozcan@karabuk.edu.tr

“

Empty your memory with  
a free() like a pointer, If you  
cast a pointer to be a integer,  
it becomes an integer. If you  
cast a pointer to a struct, it  
becomes a struct. The pointer  
can crash...And can Overflow...  
**BE A POINTER MY FRIEND.**

”

-Dennis Ritchie