BLM111 Programlama Dilleri I

Hafta 13
Fonksiyonlar

Mehmet Zahid YILDIRIM

Dizilerin Fonksiyonlara Gönderilmesi

- Bir diziyi bir fonksiyona parametre olarak göndermek için parantez kullanmadan sadece dizinin ismi belirtilir.
  - **int** myArray [**24**];
  - **myFunction** (myArray, 24);
- Char dizilerinin aksine diğer türdeki diziler her hangi bir sonlandırma karakteri içermezler.
- Bu sebeple fonksiyonlara dizideki eleman sayısıda parametre olarak gönderilir ki, fonksiyon uygun sayıda eleman üzerinde işlem yapsın.

Dizilerin Fonksiyonlara Gönderilmesi

- Dizilerin fonksiyonlara gönderilmesi referans ile çağırma işlemidir (call by reference).
- Dizinin adı aslında ilk elemanının adresidir.
- Fonksiyon böylece dizinin ilk elemanının hafıza nerede olduğunu bilir.
  - Orijinal hafıza bölgesinde işlem yapılır.
- Dizideki her hangi bir elemanın fonksiyona gönderilmesi ise değer ile çağırmadır (call by value).
  - Fonksiyona herhangi bir indisteki elemanın değeri gönderilir
    - myArray [3]
- Bir int dizi ve bir int değeri parametre olarak alan fonksiyon prototipi;
  - void myArray (int [ ], int)

Dizilerin Fonksiyonlara Gönderilmesi

```c
#include <stdio.h>
#define SIZE 5

//Function prototypes
void function1(int []);
void function2(int );

int main()
{
    int array[SIZE]={1,2,3,4,5}, i;
    printf("\nthe array before the funtions\n");
    for(int i=0;i<SIZE;i++)
        printf("%d ",array[i]);

    function1(array);
    function2(array[0]);

    printf("\nthe array after the functions\n");
    for(i=0;i<SIZE;i++)
        printf("%d ",array[i]);
    printf("\n\n");
    return 0;
}
```

Dizilerin Fonksiyonlara Gönderilmesi

```
void function1 (int A[])
{
    int i;
    for(i=0;i<SIZE;i++)
        A[i]+=i;
}
void function2(int x)// will value of array[0] change and Why?
{
    x=50;
}
```

Dizilerin Fonksiyonlara Gönderilmesi

```
void function1 (int A[])
{
    int i;
    for(i=0;i<SIZE;i++)
        A[i]+=i;
}
void function2(int x)// will value of array[0] change and Why?
{
    x=50;
}
```

```
the array before the funtion
1 2 3 4 5
the array after the function
1 3 5 7 9
```

Çok Boyutlu Dizileri Fonksiyonlara Gönderme

- Tek boyutlu dizileri fonksiyona göndermekten farklı değildir.
- Her bir boyut için köşeli parantez kullanın, ilk boyut hariç diğerleri için büyüklük belirtin.
  - void writeMatrice (int [ ] [4], int rowNumber);
  - Bu tanımlama 4 sütuna sahip her matris için farklı satır numaralarına sahip olsalar da geçerli olur.
  - void writeMatrice (int [ ] [3] [4], int rowNumber);

Çok Boyutlu Dizileri Fonksiyonlara Gönderme

```c
#include <stdio.h>

void printArray( int [][3]);
void SumArray(int [][3], int [][3]);
int main()
{
    int array1[2][3]={{1,2,3},{4,5,6}};
    int array2[2][3]={1,2,3,4,5,6};

    printf("\nbefore SumArray function\n");
    printArray(array1);

    SumArray(array1,array2);

    printf("\n\nafter Sumarray function\n");
    printArray(array1);
    printf("\n\n");
    return 0;
}
```

Çok Boyutlu Dizileri Fonksiyonlara Gönderme

```
void printArray(int arr[][3])
{
    int i, j;
    for(i=0;i<2;i++)
    {
        for(j=0;j<3;j++)
            printf("%d ",arr[i][j]);
        printf("\n");
    }
}
void SumArray(int arr1[][3], int arr2[][3])
{
    int i, j;
    for(i=0;i<2;i++)
        for(j=0;j<3;j++)
            arr1[i][j]+=arr2[i][j];
}
```

Çok Boyutlu Dizileri Fonksiyonlara Gönderme

```
void printArray(int arr[][3])
{
    int i, j;
    for(i=0;i<2;i++)
    {
        for(j=0;j<3;j++)
            printf("%d ",arr[i][j]);
        printf("\n");
    }
}
void SumArray(int arr1[][3], int arr2[][3])
{
    int i, j;
    for(i=0;i<2;i++)
        for(j=0;j<3;j++)
            arr1[i][j]+=arr2[i][j];
}
```

```
before SumArray function
1 2 3
4 5 6


after Sumarray function
2 4 6
8 10 12
```

Örnek 1

```
1   #include <stdio.h>
2   void function(char, char);
3   int main()
4   {
5       char i='A';
6       function(i++, ++i);
7       return 0;
8   }
9   void function(char x, char y)
10  {
11      char tmp;
12      tmp=x;
13      x=y;
14      y=tmp;
15      printf("%c %c", x, y);
16  }
```

Örnek 1

```
1   #include <stdio.h>
2   void function(char, char);
3   int main()
4   {
5       char i='A';
6       function(i++, ++i);
7       return 0;
8   }
9   void function(char x, char y)
10  {
11      char tmp;
12      tmp=x;
13      x=y;
14      y=tmp;
15      printf("%c %c", x, y);
16  }
```

C A

Örnek 2

```c
#include <stdio.h>

void MatrisToArray(int aa[][5], int a[]);
void incremant(int x);
int decremant(int y);

int main()
{
    int array[10], matris[2][5]={1,2,3,4,5,6,7,8,9,10}, i;

    MatrisToArray(matris, array);

    for(i=0;i<10;i++)
        printf("%d ",array[i]);

    return 0;
}
```

Örnek 2

```
19  void MatrisToArray(int aa[][5], int a[])
20  {
21      int i, j;
22      for(i=0;i<2;i++)
23          for(j=0;j<5;j++)
24              a[i*5+j]=aa[i][j];
25
26      for(i=0;i<10;i++)
27          incremant(a[i]);
28
29      for(i=0;i<10;i++)
30          a[i]=decremant(a[i]);
31  }
32  void incremant(int x)
33  {
34      x+=10;
35  }
36  int decremant(int y)
37  {
38      return y-10;
39  }
```

Örnek 2

```
19  void MatrisToArray(int aa[][5], int a[])
20  {
21      int i, j;
22      for(i=0;i<2;i++)
23          for(j=0;j<5;j++)
24              a[i*5+j]=aa[i][j];
25
26      for(i=0;i<10;i++)
27          incremant(a[i]);
28
29      for(i=0;i<10;i++)
30          a[i]=decremant(a[i]);
31  }
32  void incremant(int x)
33  {
34      x+=10;
35  }
36  int decremant(int y)
37  {
38      return y-10;
39  }
```

```
-9 -8 -7 -6 -5 -4 -3 -2 -1 0
```

Örnek 3

```
1   /* Simplifies consecutive elements in array 5552211221 */
2
3   #include <stdio.h>
4
5   void shifting(char[],int);
6   void simplifying(char []);
7
8   int main()
9   {
10      int i=0;
11      char array[]="5552211221";
12      simplifying(array);
13      printf("\n\n");
14      puts(array);
15      printf("\n\n");
16      return 0;
17  }
18
```

Örnek 3

```
19  void simplifying(char array[])
20  {
21      int i;
22      while(array[i]!='\0')
23      {
24          if(array[i]==array[i+1])
25              shifting(array,i);
26          else
27              i++;
28      }
29  }
30
31  void shifting(char a[],int i)
32  {
33      for(;a[i]!='\0';i++)
34          a[i]=a[i+1];
35  }
```

Örnek 4

```c
#include <stdio.h>

int sum(int array[], int size)
{
    int total=0, i;
    for(i=0;i<size;i++)
        total+=array[i];
    return total;
}

void multiply(int array[], int x)
{
    int i;
    for(i=0;i<5;i++)
        array[i]*=x;
}

void print(int array[], int size)
{
    int i;
    for(i=0;i<size;i++)
        printf("%d ",array[i]);
}

int main()
{
    int array[5]={1,2,3,4,5};

    multiply(array, sum(array, 5));

    print(array, 5);

    return 0;
}
```

Örnek 4

```c
#include <stdio.h>

int sum(int array[], int size)
{
    int total=0, i;
    for(i=0;i<size;i++)
        total+=array[i];
    return total;
}

void multiply(int array[], int x)
{
    int i;
    for(i=0;i<5;i++)
        array[i]*=x;
}

void print(int array[], int size)
{
    int i;
    for(i=0;i<size;i++)
        printf("%d ",array[i]);
}

int main()
{
    int array[5]={1,2,3,4,5};

    multiply(array, sum(array, 5));

    print(array, 5);

    return 0;
}
```

```
15 30 45 60 75
```

Örnek 5

```c
#include <stdio.h>
float function(void);
float calculate_area(float);
int main()
{
    float area;
    area=function();
    printf("%f",area);
    return 0;
}
float function(void)
{
    float r;
    printf("please enter the radius");
    scanf("%f",&r);
    return calculate_area(r);
}

float calculate_area(float r)
{
    float x, pi=3.14;
    x=pi*r*r;
    return x;
}
```

Örnek 6

```c
#include <stdio.h>

int Anagram(char str1[], char str2[]);

int main()
{
    char str1[100], str2[100];
    printf("\n\n Function : whether two given strings are anagram :\n");
    //Example : pears and  spare, stone and tones, listen and slient

    printf(" Input the  first String : ");
    scanf("%s",str1);
    printf(" Input the  second String : ");
    scanf("%s",str2);

    if(Anagram(str1, str2) == 1)
       printf(" %s and %s are Anagram.\n\n",str1,str2);

    else
       printf(" %s and %s are not Anagram.\n\n",str1,str2);

    return 0;
}

```

Örnek 6

```
25  //Function to check whether two passed strings are anagram or not
26  int Anagram(char str1[], char str2[])
27  {
28      int str1ChrCtr[256] = {0}, str2ChrCtr[256] = {0};
29      int ctr,a,b;
30
31      /* check the length of equality of Two Strings */
32      for(a=0;str1[a]!='\0';a++);
33      for(b=0;str2[b]!='\0';b++);
34      if(a!=b)
35          return 0;
36
37      //count frequency of characters in str1
38      for(ctr = 0; str1[ctr] != '\0'; ctr++)
39          str1ChrCtr[str1[ctr]]++;
40
41      //count frequency of characters in str2
42      for(ctr = 0; str2[ctr] != '\0'; ctr++)
43          str2ChrCtr[str2[ctr]]++;
44
45      //compare character counts of both strings
46      for(ctr = 0; ctr < 256; ctr++)
47          if(str1ChrCtr[ctr] != str2ChrCtr[ctr])
48              return 0;
49
50      return 1;
51  }
```

Kaynaklar

- Doç. Dr. Fahri Vatansever, “Algoritma Geliştirme ve Programlamaya Giriş”, Seçkin Yayıncılık, 12. Baskı, 2015.
- J. G. Brookshear, “Computer Science: An Overview 10th Ed.”, Addison Wisley, 2009.
- Kaan Aslan, “A’dan Z’ye C Klavuzu 8. Basım”, Pusula Yayıncılık, 2002.
- Paul J. Deitel, “C How to Program”, Harvey Deitel.
- Bayram AKGÜL, C Programlama Ders notları