**DİZİLER**

Diziler aynı tipte verilere tek bir isimle ulaşmak için kullanılırlar. Diziler bellekte peş peşe alanlarda saklanırlar. Bir dizinin bildirim işleminin genel biçimi şöyledir:

*veriTipi* *dizi\_adı*[*eleman\_sayısı*];

*int*  vize[10];

vize[0] =125

vize[2] =2

vize[3] =0

printf(“%d”,vize[8]);

k=vize[0] ;

**Klavyeden girilen 10 adet sayıyı bir diziye aktararak dizideki elemanların toplamını ve ortalamasını bulma:**

**#include <stdio.h>**

**#include <stdlib.h>**

**#include <conio.h>**

**#include <time.h>**

**int main(){**

**int dizi[10];**

**int toplam=0;**

**float ort;**

**for (int i=0;i<10;i++){**

**printf("Dizinin %d. Elemanını Giriniz....:",i+1);**

**scanf("%d",&dizi[i]);**

**}**

**for (int k=0;k<10;k++){**

**toplam+=dizi[k];**

**}**

**ort=(float)toplam/10;**

**printf("%d , %3.2f",toplam,ort);**

**getch();**

**return 0;**

**}**

**rasgele üretilen 10 adet sayının bir diziye aktarılması:**

**#include <stdio.h>**

**#include <stdlib.h>**

**#include <conio.h>**

**#include <time.h>**

**int main(){**

**int vize[10];**

**srand(time(NULL));**

**for (int i=0;i<10;i++){**

**vize[i]=rand()%10;**

**printf("%d\n",vize[i]);**

**}**

**getch();**

**return 0;**

**}**

**Dizilere Başlangıç Değerlerinin Verilmesi**

*veriTipi* *dizi\_adı*[*eleman\_sayısı*]={ *sabitler* };

int vize[10]={5,6,8,2,25,6,9,7,0,12} ; dizi boyutu kadar sabit verilmek zorunda değildir.

Verilmeyen indislerin değeri başlangıç değeri alır.

Başlangıç değeri verilen dizilerde boyut verilmesi zorunlu değildir. Dizinin boyutu verilen başlangıç değeri sayısı kadar olur.

**int main(){**

**int dizi[10]={5,6,8,2,25} ;**

**for (int k=0;k<10;k++){**

**printf("%d , %d\n",k,dizi[k]);**

**}**

**getch();**

**return 0;**

**}**

**Çok Boyutlu Diziler**

Tek boyutlu dizilere vektör, iki boyutlu dizilere matris denilmektedir.

Matris tanımı: *int matris[satır\_sayısı][sütun\_sayısı]*

int matris[3][4];

bellekte öncelikli olarak 1 satırın elemanları daha diğer satırların elemanları peş peşe gelecek şekilde sıralı olarak yerleştirilmektedir.

matris[0][0]=8; //matrisin ilk elemanı;

matris[2][3]=0; //matrisin son elemanı

**girilen elemanları matris olarak gösterme**

**int main(){**

**int dizi[2][3];//={1,2,3,4,5,6};**

**for (int i=0;i<2;i++){**

**for (int j=0;j<3;j++){**

**printf("(%d,%d)-->",i+1,j+1);**

**scanf("%d",&dizi[i][j]);**

**}**

**}**

**for (int i=0;i<2;i++){**

**for (int j=0;j<3;j++){**

**printf(" %d ",dizi[i][j]);**

**}**

**printf("\n");**

**}**

**getch();**

**return 0;**

**}**

**Rastgele sayılardan oluşturulan 5x5 lik bir matrisin diyagonal elemanlarının toplamını bulma**

**int main(){**

**int matris[5][5];**

**int toplam=0;**

**srand(time(NULL));**

**for (int i=0;i<5;i++){**

**for(int j=0;j<5;j++){**

**matris[i][j]=rand()%10;**

**printf(" %d ",matris[i][j]);**

**if (i==j) toplam+=matris[i][j];**

**}**

**printf("\n");**

**}**

**printf("\n\n\n Diyagonal Toplamı....:%d",toplam);**

**getch();**

**return 0;**

**}**

**Rastgele oluşturulmuş 20 elemanlı dizideki en büyük elemanı bulma**

**int main(){**

**int matris[20];**

**int eb;**

**srand(time(NULL));**

**for (int i=0;i<20;i++){**

**matris[i]=rand()%100;**

**printf(" %d ",matris[i]);**

**}**

**eb=matris[0];**

**for (int j=0;j<20;j++){**

**if (matris[j]>eb)**

**eb=matris[j];**

**}**

**printf("\n\n\n En Büyük .....:%d",eb);**

**getch();**

**return 0;**

**}**

**Sıralama**

**int main(){**

**int matris[20];**

**int temp,kontrol,count=0;**

**srand(time(NULL));**

**// diziyi doldurma**

**for (int i=0;i<20;i++){**

**matris[i]=rand()%100;**

**printf(" %d ",matris[i]);**

**}**

**//diziyi sıralama**

**while (1){**

**count++;**

**kontrol=0;**

**for (int j=0;j<19;j++){**

**if (matris[j]<matris[j+1]){**

**temp=matris[j];**

**matris[j]=matris[j+1];**

**matris[j+1]=temp;**

**kontrol=1;**

**}**

**}**

**if (kontrol==0) break;**

**}**

**printf("\n");**

**//sıralanmış diziyi yazdırma**

**for (int i=0;i<20;i++){**

**printf(" %d ",matris[i]);**

**}**

**printf("\n\n\n %d",count);**

**getch();**

**return 0;**

**}**