**STRING (KATAR) DİZİLER**

Katar dediğimiz şey de aslında bir dizidir. Değişken tipi *char* yani karakter olan diziler, '*katar*' ya da İngilizce adıyla '*string*' olarak isimlendirilirler.

char *katar\_adı*[*eleman\_sayısı*];

veya

char \**katar\_adı*;

Örnek

char ktr[10];

veya

char \*ktr;

**BAŞLANGIÇ DEĞERLERİ VERME**

char ktr[5]={'K','o','n','y','a','\0'};

char ktr[5]="Konya";

Birinci satırdaki bildirimde '\0' (NULL) sonlandırıcı karakter dizisinin sonlandığını gösterir. Daha önce de bahsedildiği gibi sonlandırıcı karakter, karakter dizileri üzerinde işlemlerin hızlı ve etkin bir biçimde yapılabilmesine olanak sağlar. İkinci bildirimde buna gerek yoktur.

Eğer bir karakter dizisinin kaç eleman sayısı belirtilmezse, başlangıçta bildirilen karakter sayısı kaç tane ise dizinin eleman sayısı o kadar olduğu varsayılır.

char ktr[] = "Konya"; /\* 5 elemanlı \*/

Eğer bildirim gösterici ile yapılmak isterirse:

char \*ktr = "Ankara"; /\* 5 elemanlı \*/

veya

char \*ktr;

ktr = "Ankara";

Ancak

char ktr[6];

ktr = "Ankara"; //??? Ktr bir değişken değil dizidir.

Elemanları katar olan diziler tanımlamak mümkündür. Örneğin en uzunu 7 karakter olan 5 farklı isim bir çatı altında şöyle toplanabilir:

char isim[5][8] = { "Semra", "Mustafa", "Ceyhun", "Asli", "Leyla" };

veya

char isim[][8] = { "Semra", "Mustafa", "Ceyhun", "Asli", "Leyla" };

veya

char \*isim[5] = { "Semra", "Mustafa", "Ceyhun", "Asli", "Leyla" };

**#include <stdio.h>**

**#include <stdlib.h>**

**#include <conio.h>**

**#include <time.h>**

**int main(){**

**char isim[30];**

**printf( "İsim giriniz> ");**

**scanf( "%s", isim );**

**printf( "Girdiğiniz isim: %s\n", isim );**

**getch();**

**return 0;**

**}**

30 karakterlik bir karakter dizisi tanımlayarak işe başladık. Bunun anlamı girdileri saklayacağımız '*isim*' katarının 30 karakter boyutunda olacağıdır. Ancak bu katara en fazla 29 karakterlik bir kelime atanabilir. Çünkü katarlarda, kelime bitiminden sonra en az bir hücre boş bırakılmalıdır. Bu hücre '*Boş Karakter*' ( *NULL Character* ) tutmak içindir. Boş karakter "**\0**" şeklinde ifade edilir. C programlama dilinde, kelimelerin bittiğini boş karakterlerle anlarız.

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

**int main(){**

**char isim[30];**

**int i;**

**printf( "İsim giriniz> ");**

**scanf( "%s", isim );**

**printf( "Girdiğiniz isim:\n ");**

**for( i = 0; isim[i]!='\0'; i++ ) // for( i = 0; i<30; i++ )**

**printf( "%c\n", isim[i] );**

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

**getch();**

**return 0;**

**}**

**int main(){**

**char isim[25], soyad[30];**

**printf( "Ad ve soyad giriniz> ");**

**scanf( "%s%s", isim, soyad );**

**printf( "Sayın %s\n%s\n, hoş geldiniz!\n", isim, soyad );**

**getch();**

**return 0;**

**}**

**Katarları farklı yazdırma yöntemleri**

**int main(){**

**char dizi[7] = {'S', 'e', 'l', 'a', 'm', '!', '\0'};**

**int i;**

**/\* Herbir karakteri ayrı ayrı alt alta yaz \*/**

**printf("Dizi elemanlari:\n");**

**for (i=0; i<7; i++)**

**printf("dizi[%d] icerigi: %c\n", i, dizi[i]);**

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

**/\* 1. yöntem: her elemanı yanyana yaz \*/**

**printf( "Butun dizi (1.yontem): ");**

**for (i=0; i<7; i++)**

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

**/\* 2. Yöntem: bütün diziyi yaz \*/**

**printf("\nButun dizi (2.yontem): ");**

**printf("%s\n", dizi);**

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

**getch();**

**return 0;**

**}**

**Girilen stringde kaç adet ‘m’ karakteri olduğunu saymaktadır.**

**int main(){**

**char str[20];**

**int i,sayac=0;**

**printf("Bir string girin: ");**

**gets(str);**

**for(i=0; str[i] != '\0'; i++)**

**if( str[i] == 'm') sayac++;**

**printf("'m' karakteri sayisi = %d\n",sayac);**

**getch();**

**return 0;**

**}**

**Çok boyutlu katar dizisinin yazdırılması**

**int main(){**

**char \*gun[7] = { "Pazartesi", "Sali", "Carsamba", "Persembe", "Cuma", "Cumartesi", "Pazar" };**

**int i;**

**for(i=0; i<7; i++)**

**printf("%d. %s\n",i+1,gun[i]);**

**getch();**

**return 0;**

**}**

**int main(){**

**//char ktr[100];**

**char \*ktr;**

**ktr=(char \*)malloc(30);**

**int i,sayac=0;**

**printf("Bir Cümle Giriniz....:");**

**scanf("%s",ktr);**

**//for( i = 0; ktr[ i ]!='\0'; i++ ) sayac++;**

**for( i = 0; \*(ktr+i)!='\0'; i++ ) sayac++;**

**printf("%d",sayac);**

**getch();**

**return 0;**

**}**

**Tersten yazma**

**int main(){**

**char ktr[100];**

**printf("Bir Cümle Giriniz....:");**

**//scanf("%s",ktr);**

**gets(ktr);**

**int i, boyut;**

**for( boyut = 0; ktr[ boyut ]!='\0'; boyut++ );**

**for( i = 0; i < boyut; i++ )**

**printf("%c", ktr[ boyut - 1 - i ] );**

**printf(" ");**

**getch();**

**return 0;**

**}**

**<string.h>**

**Strlen**

**Strchr**

**Strcpy**

**Strcmp**

**ÖDEVLER:**

1. Kullanıcının klavyeden girdiği metin içerisinde kaç kelime olduğunu bulan program kodunu yazınız.
2. Kullanıcının klavyeden girdiği metin içerisinde, yine kullanıcının girdiği kelimenin olup olmadığını bulan, varsa hangi pozisyonda olduğunu döndüren program kodunu yazınız.
3. Kullanıcının klavyeden girdiği metin içerisindeki büyük harfleri küçüğe, küçük harfleri büyüğe çeviren program kodunu yazınız.