

## Stringjet

Vargjet e simboleve të ndryshëm (shkronjave, numrave dhe simboleve speciale), përkatësisht tekstet e çfarëdoshme të shkruara brenda thonjzave, quhen stringje dhe llogariten si të dhëna të tipit string (ang. string). Kështu, p.sh., të dhëna të tipit string janë:

*"Gjuha programuese C"*

*"Jeta"*

*"y=3\*x+4"*

Meqënëse stringjet formohen si vargje të karaktereve, për këto të dhëna thuhet edhe se paraqesin stringje të karaktereve (ang. character string), sepse në memorien e kompjuterit ruhen si vargje karakteresh, të cilave kompjuteri në fund ua shton edhe karakterin zero (ang. null character) '\0' për ta përcaktuar fundin e tyre.

Kështu, p.sh., stringu "Jeta", në memorien e kompjuterit vendoset si varg karakteresh, në këtë mënyrë: J e t a \0 duke shfrytëzuar një bajt për çdo karakter. Në fakt, vlerat e fushave përkatëse në memorie mbushen me vlerat binare të ekuivalentëve decimale të kodeve përkatëse në kodin ASCII, ku për karakterin zero në fund të stringut ruhet ekuivalenti binar i numrit 0.

Duhet pasur kujdes gjatë operimit me stringje, kur si karakter paraqitet shifra decimale 0, sepse kodi i këtij karakteri në ASCII është i barabartë me numrin decimal 48, kurse njëkohësisht si kod i karakterit zero \0 shfrytëzohet numri decimal 0. Stringu i cili nuk përmban asnjë karakter quhet string i zbrazët (ang. empty string) dhe përcaktohet me dy thonjza të shënuara njëra pas tjetrës, kështu "". Në memorien e kompjuterit ky string ruhet në një bajt, sepse edhe stringut të zbrazët kompjuteri ia shton karakterin zero \0.

### - Deklarimi i stringjeve

Para se te punojme me stringje ,fillimisht duhet ti deklarojme. Ato deklarohen njelloj si tabelat me tip te dhene char. Duke perdorur tabelat, psh: char s[5];

Inicializimi i nje stringu duke perdorur tabelat:

```
char c[] = "abcd";
```

OR,

```
char c[50] = "abcd";
```

OR,

```
char c[] = {'a', 'b', 'c', 'd', '\0'};
```

OR,

```
char c[5] = {'a', 'b', 'c', 'd', '\0'};
```

Rregjistrimi ose dhenia vlere stringjeve nga perdoruesi behet nepermjet funksionit scanf().psh:

```
char c[20];
```

```
scanf("%s", c);
```

Shembull #1: duke perdorur funksionin scanf() per te rregjistruar nje string shkruani ne program ne C qe ti kerkoje perdoruesit te jape nje string.

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

```
int main()
```

```
{
```

```
    char emer[20];
```

```
    printf("Jepni nje emer: ");
```

```
    scanf("%s",&emer);
```

```
    printf("Emri juaj eshte %s.", emer);
```

```
    return 0;
```

```
}
```

Output

Jepni ne emer: Ester

Emri juaj eshte Ester.

Rregjistrimi i nje rresht text behet nepermjet funksionit getchar(). Me poshte po japim nje shembull.

Nderton nje program ne C, qe duke perdorur funksionin getchar(), te lejoje perdoruesin te rregjistroje nje karakter, ky veprim te perseritet derisa perdoruesi te jape karakterin null (\0) i cili vendoset ne fund te stringut.

```
#include <stdio.h>
int main()
{
    char emer[30], ch;
    int i = 0;
    printf("Jepni ne emer: ");
    while(ch != '\n')
    {
        ch = getchar();
        emer[i] = ch;
        i++;
    }
    emer[i] = '\0';
    printf("Emri: %s", emer);
    return 0;
}
```

Por dhe ky process eshte i lodhshem dhe jo eficient. Per kete arsye perdorim funksionet gets() dhe puts() te librarise standarte stdio.h per te rregjistruar nje rresht stringu. Me poshte nje shembull:

```
#include <stdio.h>
int main()
{
    char emer[30];
    printf("Jepni ne emer: ");
    gets(emer); //Funksioni qe lexon nje string nga perdoruesi
    printf("Emri: ");
}
```

```
puts(emer); //Funksioni qe afishon nje string
return 0;
}
```

Output

Jepni nje emer: Tom Hanks

Emri: Tom Hanks

### Trajtimi i stringjeve

Shpesh ju duhet të manipuloni stringje në përputhje me nevojën e një problemi. Në përgjithësi, manipulimi i një stringu mund të bëhet me dorë, por, kjo e bën kodin kompleks dhe të madh. Për të zgjidhur këtë problem përdorim librarine standarte string.h, në të cilin ndodhen funksionet e mëposhtme për të trajtuar stringjet:

Funksioni	Pershkrimi
strlen()	Llogarit gjatësinë e vargut
strcpy()	Kopjon nje string te nje string tjetër
strcat()	Bashkon dy stringje
strcmp()	Krahason dy stringje
strlwr()	Konverton nje string ne germa te vogla
strupr()	Konverton nje string ne germa te medha

Ushtrimi 1 Llogarisni gjatësinë e një stringu duke përdorur strlen().

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

```

int main()
{
    char s[1000], i;

    printf("Enter a string: ");
    scanf("%s", s);
    for(i = 0; s[i] != '\0'; ++i);
    printf("Length of string: %d", i);
    return 0;
}

```

Ushtrimi 2 Ky program i kërkon përdoruesit që të japë një string dhe një karakter dhe kontrollon sa herë përsëritet në kete string ky karakter.

```

#include <stdio.h>

int main()
{
    char str[1000], ch;
    int i, frequency = 0;

    printf("Enter a string: ");
    gets(str);
    printf("Enter a character to find the frequency: ");
    scanf("%c",&ch);
    for(i = 0; str[i] != '\0'; ++i)
    {
        if(ch == str[i])
            ++frequency;
    }
    printf("Frequency of %c = %d", ch, frequency);
}

```

```
    return 0;
}
```

Ushtrimi 3 Ky program i kërkon përdoruesit që të japë dy stringje dhe i bashkon pa përdorur funksionin strcat().

```
#include <stdio.h>
int main()
{
    char s1[100], s2[100], i, j;
    printf("Jepni string-un e pare: ");
    scanf("%s", s1);

    printf("Jepni string-un e dyte: ");
    scanf("%s", s2);
    // llogarit gjatesine e stringut te pare s1 dhe ruaje ne variablin i

    for(i = 0; s1[i] != '\0'; ++i);
        for(j = 0; s2[j] != '\0'; ++j, ++i)
            {
                s1[i] = s2[j];
            }
    s1[i] = '\0';
    printf("pas bashkimit: %s", s1);
    return 0;
}
```