

Tema 4

Char tip, stringovi

char tip

- Promenljiva tipa char koristi se za čuvanje tekstualnih znakova
- Znakovi mogu biti:
 - slova
 - cifre
 - specijalni znaci
 - neštampajući (beli) znaci
- Može se koristiti i za čuvanje malih, celobrojnih vrednosti
 - U slučaju signed tipa od -128 do 127
 - U slučaju unsigned tipa od 0 do 255
- Primeri char konstanti: 'a', 'A', '1', '?', '\n'

char je samo broj

- Svakom karakteru pridružuje se numerički kod
- Postoje različiti skupovi kodova (standardi):
 - ASCII (American Standard Code for Information Interchange)
 - Najčešće korišćen
 - EBCDIC
 - zastareo, danas se retko koristi
 - Unicode
 - predstavnik novijih skupova karaktera
- Koristićemo ASCII

Primer 1

Korišćenje char kao znakovnog tipa i za malu numeričku vrednost

```
#include <stdio.h>

int main()
{
    char znak;

    printf("Unesite znak: ");
    scanf("%c", &znak);

    printf("Znak kao karakter je: %c\n", znak);
    printf("Numericka vrednost znak-a je: %d\n", znak);
    printf("Karakter posle %c je %c\n", znak, znak + 1);

    return 0;
}
```

Stringovi

- Nizovi karaktera sa specijalnom oznakom za kraj stringa
 - `'\0'` terminator, označava kraj stringa, "žrtvovana" jedna lokacija kako ne bi bilo potrebe da se posebno prati zauzeće niza
- Unos i ispis sa standardnog ulaza
 - `scanf` i `printf` sa format specifikatorom `%s`
 - Specijalizovane funkcije za unos i ispis stringova `gets` i `puts`
- Biblioteka sa funkcijama za rad sa stringovima `string.h`
 - `strlen` - vraća dužinu stringa
 - `strcat` - spajanje stringova
 - `strcpy` - kopiranje stringa
 - `strstr` - pokazivač na prvu pojavu stringa unutar većeg stringa
 - `strcmp` - poređenje stringova
- Za više informacija o funkciji, koristiti man stranice, na primer: `man strlen`

Primer 1

Funkcije za unos i ispis stringova

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

#define MAX_STRING 101

int main()
{
    char str1[MAX_STRING], str2[MAX_STRING];

    printf("Unesite prvi string: ");
    scanf("%s", str1);
    __fpurge(stdin);

    printf("Unesite drugi string: ");
    fgets(str2, MAX_STRING, stdin);
    int duzina_str2 = strlen(str2);
    if(str2[duzina_str2 - 1] == '\n') str2[duzina_str2 - 1] = '\0';
```

Programski jezici i strukture podataka - Tema 4

```
printf("\n%s\n", str1);  
printf("%s\n\n", str2);  
  
puts(str1);  
puts(str2);  
  
return 0;  
}
```

Primer ispisa na standardnom izlazu:

```
Unesite prvi string: Primer prvog stringa  
Unesite drugi string: Primer drugog stringa  
  
Primer  
Primer drugog stringa  
  
Primer  
Primer drugog stringa
```

- scanf je funkcija osetljiva na whitespace karaktere

Primer 2

Korišćenje funkcije strcat za spajanje stringova

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

#define MAX_STRING 31

int main()
{
    char str1[MAX_STRING] = "Veni, ";
    char str2[] = "vidi, ";

    printf("%s\n", strcat(str1, str2));
    strcat(str1, "vici.");
    printf("%s\n", str1);

    return 0;
}
```

Primer 3

Korišćenje funkcije strcpy za kopiranje stringova

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

#define MAX_STRING 31

int main()
{
    char str1[MAX_STRING];

    strcpy(str1, "Znanje");
    puts(str1);
    puts(strcpy(str1, "Imanje"));

    return 0;
}
```

Napomene

- Nikada ne koristiti operator dodele vrednosti (=) nad stringovima!
 - Ukoliko se pokuša dodela stringa drugom stringu, to će biti sintaksna greška
 - Dodela pokazivačkoj promenljivoj tipa `char *` je moguća, ali u pitanju je referenca na isti string
 - Svaka modifikacija preko pokazivačke promenljive promeniće originalni string
- Korišćenjem funkcije `strcpy`, vrši se kopiranje sadržaja jednog stringa u drugi
 - Karakteri će biti kopirani jedan po jedan
 - Izmena sadržaja u drugom stringu neće uticati na prvi string

Primer 4

Korišćenje funkcije `strstr` za prebrojavanje pojave stringa unutar većeg stringa (`strstr` vraća `NULL` kad mali string ne postoji unutar velikog)

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

int main()
{
    char str_veliki[] = "tatatatira", str_mali[] = "ta";
    char *pstr_veliki = str_veliki;
    int br = 0;

    while(strstr(pstr_veliki, str_mali) != NULL)
    {
        br++;
        pstr_veliki = strstr(pstr_veliki, str_mali) + strlen(str_mali);
    }

    printf("Broj pojavljivanja stringa %s u stringu %s je: %d\n", str_veliki, str_mali, br);

    return 0;
}
```

Primer 5

Korišćenje funkcije strcmp

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

#define MAX_STRING 101

int main()
{
    char str1[MAX_STRING] = "";

    while(strcmp(str1, "exit") != 0)
    {
        fgets(str1, MAX_STRING, stdin);
        int duzina_str1 = strlen(str1);
        if(duzina_str1 < MAX_STRING - 1) str1[strlen(str1) - 1] = '\0';
        printf("Unet je string: %s\n", str1);
    }

    return 0;
}
```

Napomene

- Nikada ne koristiti relacioni operator provere jednakosti (==) za poređenje stringova!
 - Uporediće se adrese nizova, relacioni izraz će biti tačan samo ukoliko se poredi string sa samim sobom
- Stringovi su nizovi karaktera, njihov sadržaj je moguće porediti samo karakter po karakter
 - Što funkcija strcmp u osnovi i radi, a vodi računa i o dužini poređenih stringova

Zadatak 1

Napisati program koji ispisuje string obrnutim redosledom od unetog.

- Koristiti funkciju `strlen`

Zadatak 2

Napisati program koji proverava da li je uneti string palindrom. Ukoliko je string palindrom, u pitanju su reči ili rečenice koje se mogu isto čitati s leva na desno i obrnuto.

- Napraviti program tako da prvo radi za reči koje su palindromi (na primer, "rotor")
- Proširiti funkcionalnost tako da je moguće isto uraditi i za skup reči (na primer, "Ana voli Milovana")
 - Ignorirati razmake i odgovarajuća mala i velika slova smatrati istim prilikom provere
- Proširiti funkcionalnost tako da je moguće isto uraditi i za rečenice (na primer, "Ana voli Milovana.")
 - Ignorirati znake interpunkcije

Zadatak 3

Napisati program koji kao parametre uzima jedan string i karakter. Program ispisuje broj pojavljivanja karaktera u stringu.

- Za primer stringa "tatatatira" i karaktera 'a', broj pojavljivanja karaktera je 4

Zadatak 4

Napisati program koja modifikuje string tako što njegova mala slova transformiše u velika i obrnuto.

- Znakove koji nisu slova program ne treba da obrađuje, već ih ostavlja takve kakvi su