Programiranje i programski jezici

Stringovi

String je jednodimenzionalni niz tipa **char** koji ne mora zauzimati sve elemente niza u kojima se čuva.

```
"Ja sam string"
        "" /* prazan string */
        "hello, " "world"
```

je isto što i `"hello, world"`

Stringovi su nizovi karaktera koji se završavaju znakom **\0**

```
/* strlen: return length of s */
int strlen(char s[])
{
    int i;

    while (s[i] != '\0')
            ++i;

    return i;
}
```

Funkcije za rad sa stringovima: <string.h>

'a' nije isto što i **"a"**

Stringovi

Inizijalizacija stringova:

- Statički nizovi
- Spoljašnji nizovi
- Automatski nizovi

```
static char s[]="IBM PC"   ili
static char s[]={'I','B','M',' ','P','S','\0'}
```

Kao i kod drugih nizova ime s predstavlja pokazivač na nulti element niza, tako da važi:

```
S==&s[0],  *s=='I',    *(s+1)==s[1]=='B'
```

Primer 1:

Napisati program koji ispisuje string, inicijalizovan kao statički niz, u direktnom i inverznom poretku.

```
#include<stdio.h>
main()
{
    char s[]="IBM PC-486";
    char *pok;
    pok=s;
    while(*pok)
        putchar(*pok++);

    while(--pok>=s)
        putchar(*pok);
    putchar('\n');
}
```

Stringovi

Ako je:

```
char s[10]="IBM PC"
```

tada se on registruje na sledeći način:

I	B	M		P	C	\0	\0	\0	\0

Ispravno je napisati i:

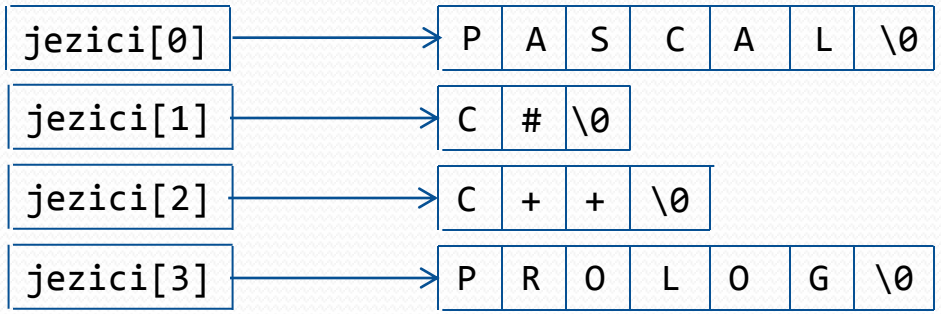
```
char s*="IBM PC-486";
```

Nizovi stringova

```
static char *jezici[4]={“PASCAL”,”C#”,”C++”,”PROLOG”};
```

definisan je niz pokazivača jezici na stringove. Nulti pokazivač `jezici[0]` pokazuje na nulti string, prvi pokazivač `jezici[1]` pokazuje na prvi string, tako da važi:

```
*jezici[0]==‘P’
*jezici[1]==‘C’
*jezici[2]==‘C’
*jezici[3]==‘P’
```



Dužina stringa se može eksplicitno zadati na sledeći način:

```
static char jezici[4][10]={“PASCAL”,”C#”,”C++”,”PROLOG”};
```

Primer 2:

Napisati funkciju kojom se na osnovu parametra koji je pokazivač na string ispituje da li je string palindrom. String je palindrom ako se isto čita s leva kao i s desna. Funkciju testirati sledećom **main()** funkcijom:

```
#include<stdio.h>
main()
{
    if (palindrom(“ANAVOLIMILOVANA”))
        printf(“String 1 je palindrom\n”);
    if (palindrom(“ASTERIX”))
        printf(“String 2 je palindrom\n”);
}
```

Primer 2:

```
int palindrom(char *strpok)
{
    char *strpok1=strpok;
    while(strpok1)
        ++strpok1;

    --strpok1;
    while(strpok<strpok1)
        if(*strpok++!=*strpok1--)
                return 0;

    return 1;
}
```

Učitavanje stringova

gets() - prihvata znake sa tastature dok ne naiđe na znak za novu liniju (‘\n’), ENTER

scanf() - prihvata znake sa tastature do praznog znaka (praznina, tabulacija, nova linije)

```
#include <stdio.h>
main()
{
      char ime[100];
      char *pok;
      pok=gets(ime);
      printf("%s",pok);
}
```

ili

```
#include <stdio.h>
main()
{
      char ime[100];
      gets(ime);
      printf("%s",ime);
}
```

Primer 3:

Napisati program koji učitava reč sa standardnog ulaza koja ima ne više od 100 karaktera i ispisuje je na standardni izlaz. Reč je ma koji niz karaktera koji ne sadrži beline (blanko, tab, prelaz u novi red,...). Učitanu reč čuvati u stringu (nizu).

```
#include <stdio.h>
#include <ctype.h>
void get_word(char s[]) {
    int c, i = 0;
    while (!isspace(c=getchar())) s[i++] = c;
    s[i] = '\0'; /*OBAVEZAN kraj stringa u C-u */
}
main() {
    char s[100];
    get_word(s);
    printf("%s\n", s);
}
```

Ispisivanje stringova

puts() - prihvata znake sa tastature dok ne naiđe na znak za novu liniju (‘\n’), ENTER

printf() - prihvata znake sa tastature do praznog znaka (praznina, tabulacija, nova linije)

```
#include <stdio.h>
main()
{
        char s1[100]=“Prvi”;
        char *s2=“Drugi”;
        puts(s1);
        puts(s2);
        printf("%s",s1);
}
```

Funkcije za rad sa stringovima

strlen() - na osnovu argumenta koji je pokazivač na string vraća broj znakova stringa bez završnog nul znaka.

Da bi se u programu mogla krostiti ova funkkcija potrebno je uključiti zaglavlje <string.h>

```
#include <stdio.h>
#include <string.h>
main()
{
        char s1[]=“Oda radosti”;
        printf("%d",strlen(s1));
        printf("%d",strlen(“Oda radosti”));
        printf("%d",strlen(“Oda\0radosti”));
        printf("%d",strlen(“”));
}
```

Funkcije za rad sa stringovima

strstr() - vraća pokazivač na prvi znak u str1 počev od koga je str2 sadržan u str1, inače vraća NULL.

```
char *strstr(char *str1, char *str2)
```

```
printf("%s\n", strstr("programiranje","gra")   ->   gramiranje
```

strcat() - povezuje dva stringa tako sto na string str1 dodaje str2, string str2 ne menja.

```
char *strcat(char *str1, char *str2)
```

-voditi računa da je za prvi string rezervisano dovoljno prostora.

strcmp() - poredi se znak po znak dok se ne ustanovi razlika ili dok se jedan od stringova ne završi. Ako su stringovi jednaki vraća nulu. U protivnom vraća -1 ukoliko prvi različit znak ima manju ASCII vrednost od drugog, u suprotnom vraća 1.

```
int strcmp(char *str1, char *str2)
```

Funkcije za rad sa stringovima

```
#include <stdio.h>
#include <string.h>
main()
{

      printf("%d", strcmp(“ABC”,”ABC”));
      printf("%d", strcmp(“ABC”,”BCD”));
      printf("%d", strcmp(“BCD”,”ABC”));
      printf("%d", strcmp(“ABC”,”ABCD”));
}
```

strcpy() - string na koji pokazuje str

char *strcpy(char *str1, char *str2) se kopira znak po znak u oblast memorije na koju pokazuje str1.

Funkcije za rad sa stringovima

atoi() - vraća ekvivalentan ceo broj. Proizvoljan broj nula, tabulacija, praznina se ignorisu. Broj se završava sa '\0', ili bilo kojim znakom koji nije cifra. Mora se uključiti <stdlib.h>

```
int atoi(char *string)
```

```
#include <stdio.h>
#include <stdlib.h>
main()
{

      printf("%d\n", atoi("423"));
      printf("%d\n", atoi("4.32"));
      printf("%d\n", atoi("632xy"));
      printf("%d\n", atoi("000023"));
      printf("%d\n", atoi("-1.25"));
      printf("%d\n", atoi("1234567812325"));
}
```

Ukoliko se želi krosititi funkcija koja za dati broj vraća ekvivalentan string: sprintf(string,"%d",broj); (nekada je postojala funkcija itoa())

Zadatak 1:
Napisati funkciju kojom se na osnovu parametra koji je pokazivač na string ispituje broj pojavljivanja datog znaka u stringu.

Zadatak 2:
Napisati funkciju koja od stringa **s** formira novi u kome se redom izbacuje svaki znak *, a svaki znak koji se od njega razlikuje duplira.

Zadatak 3:
Napisati funkciju kojom se određuje dužina najvećeg neprekidnog niza praznina stringa **s**.

Zadatak 4:
Napisati funkciju kojom se učitanu niz stringova sortira u alfabetskom poretku.