

NISKE (stringovi) u C-u: podsećanje

1. Primer programa koji uvodi niske karaktera terminisane nulom

```
#include <stdio.h>
main()
{
    /* Poslednji bajt niske karaktera s se postavlja na '\0' tj. 0 */
    char s[] = {'a', 'b', 'c', '\0' };

    /* Kraci nacin da se postigne prethodno */
    char t[] = "abc";

    /* Ispis niske s karakter po karakter - I nacin*/
    int i;
    for (i = 0; s[i] != '\0'; i++) putchar(s[i]);
    putchar('\n');

    /* Ispis niske s koristeci funkciju printf - II nacin*/
    printf("%s\n", s);

    /* Ispis niske t karakter po karakter-I nacin*/
    for (i = 0; t[i] != '\0'; i++) putchar(t[i]);
    putchar('\n');

    /* Ispis niske t koristeci funkciju printf - II nacin*/
    printf("%s\n", t);
}
```

2.

```
/* strlen, strcpy, strcat, strcmp, strchr, strstr -manipulacija niskama karaktera */
```

Napisati C funkcije koje implementiraju funkcije iz biblioteke string.h i to:

1. funkciju string_length koja implementira funkciju strlen iz zaglavlja string.h
2. funkciju string_copy koja implementira funkciju strcpy iz string.h
3. funkciju string_concatenate koja implementira funkciju strcat iz string.h
4. funkciju string_char koja implementira funkciju strchr iz string.h
5. funkciju string_last_char koja pronalazi poslednju pojavu karaktera u nisci
6. funkciju string_string koja slicno funkciji strstr iz string.h proverava da li je neka niska podniska druge NCP koji ilustruje pozive ovih funkcija.

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

```
/* Izracunava duzinu stringa */
```

```
int string_length(char s[])
{
    int i;
    for (i = 0; s[i]; i++);
    return i;
}
```

```
/* Kopira string src u string dest. Prepostavlja da u dest ima dovoljno prostora. */
```

```
void string_copy(char dest[], char src[])
```

```

{
/*Kopira karakter po karakter,sve dok nije iskopiran karakter '\0' */
int i;
for (i = 0; (dest[i]==src[i]) != '\0'; i++) ;

/* Uslov != '\0' se, naravno, moze izostaviti : for (i = 0; dest[i]==src[i]; i++); */

}

/* Nadovezuje string t na kraj stringa s. Prepostavlja da u s ima dovoljno prostora.*/
void string_concatenate(char s[], char t[])
{
    int i, j;

    /* Pronalazimo kraj stringa s */
    for (i = 0; s[i]; i++);

    /* Vrsi se kopiranje, slicno funkciji string_copy, ali ne od pocetka niske s, vec od kraja niske s */
    for (j = 0; s[i] = t[j]; j++, i++) ;

}

/* Vrsi leksikografsko poredjenje dva stringa.
Vraca :
0 - ukoliko su stringovi jednaki
<0 - ukoliko je s leksikografski ispred t
>0 - ukoliko je s leksikografski iza t
*/
int string_compare(char s[], char t[])
{
    int i;

    /* Petlja tece sve dok ne nadjemo na prvi razliciti karakter */
    for (i = 0; s[i]==t[i]; i++)
        if (s[i] == '\0') return 0; /* Naisli smo na kraj oba stringa, a nismo nasli razliku */

    /* s[i] i t[i] su prvi karakteri u kojima se niske razlikuju.
    Na osnovu njihovog odnosa (razlike ASCII vrednosti), odreduje se odnos stringova < ili > ili ==*/
    return s[i] - t[i];
}

/* Pronalazi prvu poziciju karaktera c u stringu s, odnosno -1 ukoliko s ne sadrzi c */
int string_char(char s[], char c)
{
    int i;

    for (i = 0; s[i]; i++)
        if (s[i] == c) return i;

    /* NIKAKO
    else return -1;
*/
}

```

```

/* Nije nadjeno */
return -1;
}

/* Pronalazi poslednju poziciju karaktera c u stringu s, odnosno -1 ukoliko s ne sadrzi c */
int string_last_char(char s[], char c)
{
    int i;

    /* Najpre pronalazimo kraj stringa s */
    for (i = 0; s[i]; i++) ;

    /* Krecemo od kraja i trazimo c unazad */
    for (i--; i>=0; i--)
        if (s[i] == c) return i;

    /* Nije nadjeno */
    return -1;

    /* ILI drugi nacin bi bio koristeci string_length :
       for (i = string_length(s) - 1; i>0; i--)
           if (s[i] == c) return i;
       return -1;
    */
}

/* Proverava da li string str sadrzi string sub.
Vraca poziciju na kojoj sub pocinje, odnosno -1 ukoliko ga nema
*/
int string_string(char str[], char sub[])
{
    int i, j;

    /* Proveravamo da li sub pocinje na svakoj poziciji i */
    for (i = 0; str[i]; i++)
        /* Poredimo sub sa str pocevsi od poziciji i sve dok ne nadjemo na razliku */
        for (j = 0; str[i+j] == sub[j]; j++)
            /* Nismo naisli na razliku a ispitali smo sve karaktere niske sub */
            if (sub[j+1]=='\0') return i;

    /* Nije nadjeno */
    return -1;
}

main()
{
char s[100];
char t[] = "Zdravo";
char u[] = " svima";

string_copy(s, t);
printf("%s\n", s);
}

```

```
string_concatenate(s, u);
printf("%s\n", s);

printf("%d\n", string_char("racunari", 'n'));

printf("%d\n", string_last_char("racunari", 'a'));

printf("%d\n", string_string("racunari", "rac"));

printf("%d\n", string_string("racunari", "ari"));

printf("%d\n", string_string("racunari", "cun"));

printf("%d\n", string_string("racunari", "cna"));

}
```

Izlaz:

Zdravo

Zdravo svima

4

5

0

5

2

-1

Zaglavlje string.h

This header file defines several functions to manipulate *C strings* and arrays.

Functions

Copying:

[memcpy](#)

Copy block of memory (function)

[memmove](#)

Move block of memory (function)

[strcpy](#)

Copy string (function)

[strncpy](#)

Copy characters from string (function)

Concatenation:

[strcat](#)

Concatenate strings (function)

[strncat](#)

Append characters from string (function)

Comparison:

memcmp

Compare two blocks of memory (function)

strcmp

Compare two strings (function)

strcoll

Compare two strings using locale (function)

strncmp

Compare characters of two strings (function)

strxfrm

Transform string using locale (function)

Searching:

memchr

Locate character in block of memory (function)

strchr

Locate first occurrence of character in string (function)

strcspn

Get span until character in string (function)

strupr

Locate characters in string (function)

strrchr

Locate last occurrence of character in string (function)

strspn

Get span of character set in string (function)

strstr

Locate substring (function)

strtok

Split string into tokens (function)

Other:

memset

Fill block of memory (function)

strerror

Get pointer to error message string (function)

strlen

Get string length (function)

Macros

NULL

Null pointer (macro)

Types

size_t

Unsigned integral type (type)

3. Napisati funkciju

```
void zameni(char s[])
```

koja u niski s maksimalne duzine 20 karaktera zamenuje svako pojavljivanje slova a slovom b.

Na primer: niska dan se zamenjuje niskom dbn

U glavnom programu ucitati nisku sa ulaza i testirati funkciju.

4. Napisati funkciju

```
void zameni3(char s[], char ns[])
```

koja zamenjuje svaki karakter niske s sa 3 naredna karaktera po abededi i tako dobijenu nisku cuva u ns.

Na primer: nisku ANA treba modifikovati u BCDOPQBCD

Napisati potom program koji sa ulaza ucitava 5 niski maksimalne duzine 20 karaktera i ispisuje odgovarajuce modifikacije.

5. Napisati program kojim se sa ulaza ucitava ceo broj k ($k \leq 10$) i niska s maksimalne duzine 20 karaktera, a zatim se ispisuje nova niska formirana nadovezivanje polazne niske n puta.

Na primer: za unos 4 i ana izlaz je ana_ana_ana_ana

6. Napisati funkciju

```
int palindrom (char s[])
```

kojom se proverava da li je zadata niska s palindrom (za nisku kazemo da je palindrom ako se isto cita i sa leve i sa desne strane).

Na primer: niska anavolimilovana je palindrom

U glavnom programu ucitati nisku i testirati rad funkcije.

7. Napisati funkciju

```
int anagrami (char s1[], char s2[])
```

kojom se proverava da li su dve zadate niske anagrami (za dve niske kazemo da su anagrami ako se od slova prve niske moze formirati druga niska).

Na primer: ortoped i torpedo

U glavnom programu ucitati dve niske maksimalne duzine 20 i testirati rad funkcije.