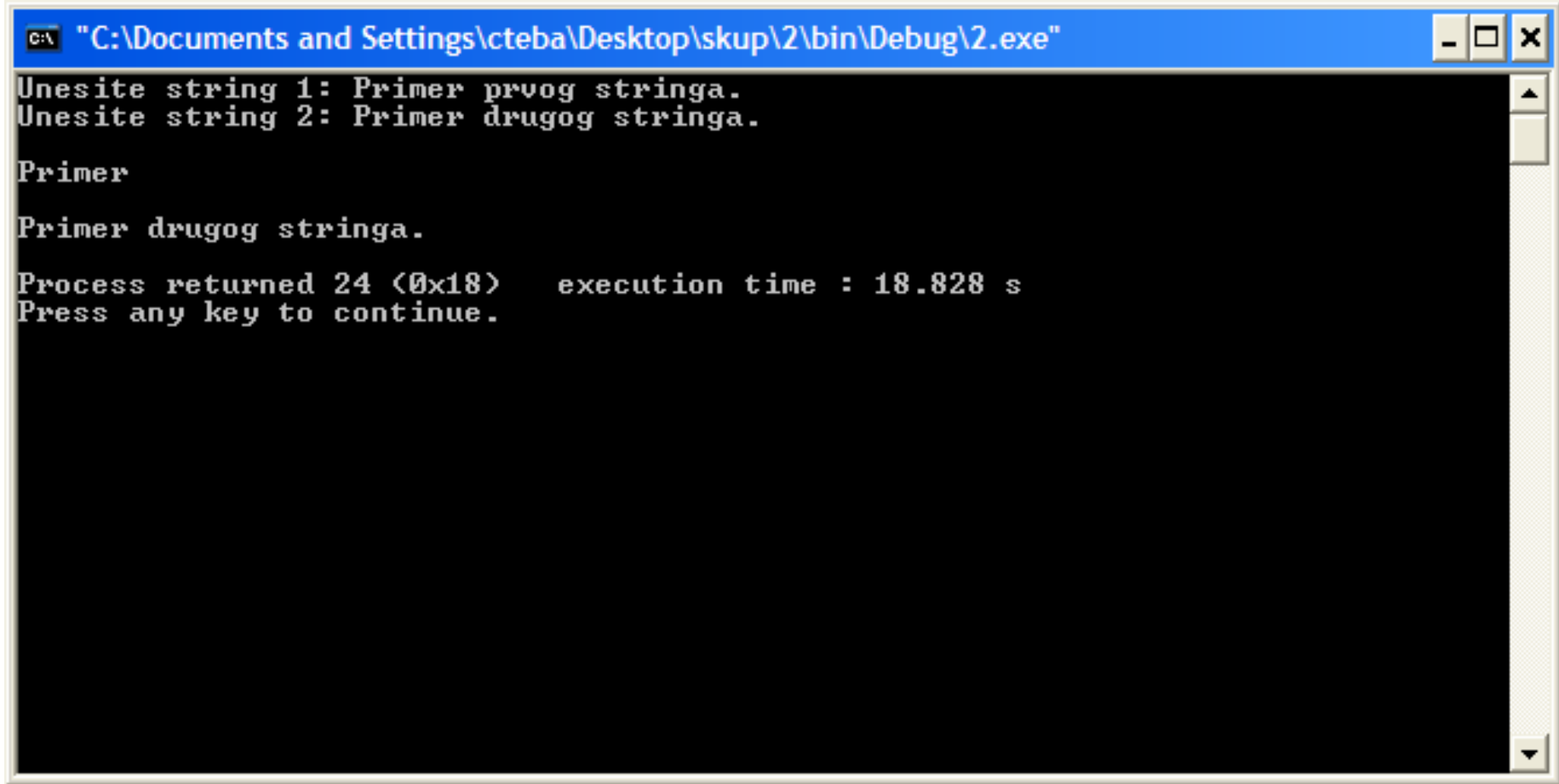


Tema 5

Stringovi

```
1  #include <stdio.h>
2
3  int main() {
4      char str1[10], *str2;
5
6      printf("Unesite string: ");
7      scanf("%s", str1);
8      str2 = str1;
9
10     while (*str2)
11         printf("%c ", *str2++);
12
13     printf("\n%s\n", str1);
14
15     return 0;
16 }
```

```
1  #include <stdio.h>
2
3  int main() {
4      char str1[101], str2[101];
5
6      printf("Unesite string 1: ");
7      scanf("%s", str1);
8
9      //fflush(stdin); //pod windows-om
10     __fpurge(stdin); //pod linux-om
11
12     printf("Unesite string 2: ");
13     gets(str2);
14
15     printf("\n%s\n", str1);
16     printf("\n%s\n", str2);
17
18     puts(str1);
19     puts(str2);
20
21     return 0;
22 }
```



A screenshot of a Windows command prompt window. The title bar is blue and contains the text "C:\Documents and Settings\cteba\Desktop\skup\2\bin\Debug\2.exe" along with standard window control buttons (minimize, maximize, close). The command prompt area has a black background with white text. The output shows two lines of string assignment, followed by two lines of output, and then a return code and execution time. The window has a vertical scrollbar on the right side.

```
C:\Documents and Settings\cteba\Desktop\skup\2\bin\Debug\2.exe
Unesite string 1: Primer prvog stringa.
Unesite string 2: Primer drugog stringa.

Primer
Primer drugog stringa.

Process returned 24 (0x18)    execution time : 18.828 s
Press any key to continue.
```

Zadatak 1

Napisati funkciju koja ispisuje string obrnutim redom. Napraviti kratak test program (učitati string i ispisati obrnutim redosledom karaktere) sa i bez upotrebe f-ije **strlen()**.

Primer 3

Napisati funkciju kojom se ispisuje:

- a) prvih **k** znakova string promenljive **s**;
- b) poslednjih **k** znakova string promenljive **s**;
- c) deo stringa **s** od pozicije **p** na dužini **k**.

Primer 3

```
// A)
void levi(char *s,int k)
{
    while(k && *s)
    {
        putchar(*s++);
        k--;
    }
    putchar('\n');
}
```

```
// B)
void desni(char *s, int k)
{
    if(strlen(s)>=k)
        puts(s+strlen(s)-k);
    else
        puts(s);
}
```

```
// C)
void kopi(char *s, int p, int k)
{
    levi(s+p,k);
}
```

Zadatak 2

Napisati program koji proverava da li je učitani string palindrom. Program napraviti tako da radi za reči a potom ga proširiti da radi i sa rečenicama (ignorišući znakove razmaka) kako bi string “Ana voli Milovana” program registrovao kao palindrom.

Zadatak 3

Šta ispisuje sledeći program?

```
#include <stdio.h>
int main() {
    static char film[]="Tango argentino";
    char *pok;
    pok=film;
    puts(pok);
    puts(++pok);
    film[5]='\0';
    puts(film);
    puts(++pok);
    return 0;
}
```

Zadatak 4

Napisati funkciju koja kao parametre uzima jedan string i karakter, a vraća broj pojavljivanja tog karaktera u string. Napisati test program. (Primer: str="tatatatira", c='a', povratna vrednost je 4).

Zadatak 5

Šta ispisuje sledeći program?

```
#include <stdio.h>
int main() {
    static char ime[] = "Bingo";
    char *pok;
    pok=ime+strlen(ime);
    while(--pok>=ime)
        puts(pok);
    return 0;
}
```

Zadatak 6

Napisati funkciju **int jednaki(char *s1, char *s2)** koja proverava da li su stringovi *s1* i *s2* jednaki. U glavnom programu od korisnika učitati dva stringa (maksimalne dužine 20 karaktera) i ispisati da li su jednaki.

Primer 4 – funkcija strstr()

Napisati funkciju kojom se određuje broj pojavljivanja stringa **s1** u stringu **s**.

```
int broj(char *s, char *s1)
{
    int br=0;
    while(strstr(s,s1) != NULL)
    {
        br++;
        s=strstr(s,s1)+strlen(s1);
    }
    return br;
}
```

Primer 5– funkcija **strcat()**

Sledeći primer ilustruje korišćenje funkcije **strcat()**:

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

int main() {
    static char s1[30] = "Veni, ";
    static char s2[] = "vidi, ";

    printf("%s\n", strcat(s1,s2));
    strcat(s1,"vici.");
    printf("%s\n",s1);

    return 0;
}
```

Primer 6 – funkcija **strcmp()**

Napisati program za učitavanje stringova koje se prekida kada naiđe na EOF (tada **gets()** vraća NULL), prazan string ili dostigne predviđen broj članova niza koji se zadaje konstantom MAX. Obezbediti ispis unetog niza.

Primer 6 – funkcija strcmp()

```
#include <stdio.h>
#include <string.h>
#define DUZINA 81
#define MAX 50
#define STOP ""

int main() {
    static char str[MAX][DUZINA];
    int i=0,j;

    while(gets(str[i])!=NULL && strcmp(str[i],STOP)!=0 && i++<MAX);

    for(j=0;j<i;j++)
        puts(str[j]);

    return 0;
}
```


Primer 7 – funkcija **strcpy()**

Sledeći primer ilustruje korišćenje funkcije **strcpy()**:

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

int main() {
    char str1[31];

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

    return 0;
}
```

Zadatak 7

Napisati funkciju **void updown(char *s)** koja mala slova u stringu *s* pretvara u velika, i obrnuto. Ostale karaktere u stringu (cifre, znakove interpunkcije, itd) funkcija ne sme modifikovati. U glavnom programu od korisnika učitati string. Nakon toga, string obraditi implementiranom funkcijom i ispisati rezultat.

Zadatak 8

Napisati funkciju **int atof(char *s, float *f)** koja string *s* pretvara u realan broj i rezultat vraća u parametru *f*. Realan broj se definiše kao niz od jedne ili više cifara, nakon koje može biti tačka, nakon koje može biti još nula ili više cifara. Ukoliko string nije dobro formatiran (npr. sadrži dve tačke, zarez, slovo, itd) funkcija vraća 0. Inače, vraća 1. U glavnom programu od korisnika učitati dužinu stringa, pa i sam string, a zatim ga konvertovati u realan broj i ispisati rezultat. Ukoliko je prilikom konverzije došlo do greške (tj. ako string sadrži nedozvoljeni karakter), ispisati odgovarajuću poruku.

Zadatak 9

Napisati program koji od korisnika učitava niz stringova i ispisuje ih na ekran. Program treba od korisnika najpre da učitava dužinu niza (odnosno broj stringova), a zatim svaki string.

Zadatak 10

U datom stringu:

- a) odrediti broj reči;
- b) broj reči koje počinju sa M;
- c) broj reči čiji su prvi i poslednji znak jednaki;
- d) odrediti dužinu najkraće reči