

Programski jezik I

– Primjeri sa 5. termina predavanja –

Nizovi. Stringovi. Funkcije

1. Napisati program koji učitava dva niza, **x** i **y**, i koji provjerava da li je niz **y** podniz niza **x**. Ispisati odgovarajuću poruku.

Primjer: niz **y**=[2 3 4] je podniz niza **x**=[1 2 3 4 5].

Rješenje:

```
01 | #include <stdio.h>
02 |
03 | int main()
04 | {
05 |     int i, j, x[20], y[20], N, M, ind;
06 |
07 |     printf("Unesite broj elemenata niza x: ");
08 |     scanf("%d", &N);
09 |
10 |     printf("Unesite niz x: ");
11 |     for(i = 0; i < N; i++)
12 |         scanf("%d", &x[i]);
13 |
14 |     printf("Unesite broj elemenata niza y: ");
15 |     scanf("%d", &M);
16 |
17 |     printf("Unesite niz y: ");
18 |     for(i = 0; i < M; i++)
19 |         scanf("%d", &y[i]);
20 |
21 |     for(i = 0; i <= N - M; i++)
22 |     {
23 |         ind = 0;
24 |         for(j = 0; j < M; j++)
25 |             if(x[i+j] != y[j])
26 |                 ind = 1;
27 |         if (ind == 0)
28 |         {
29 |             printf("Jeste podniz");
30 |             break;
31 |         }
32 |
33 |     }
34 |     if (ind == 1)
35 |         printf("Nije podniz");
36 | }
```

Za vježbu, modifikovati program tako da izbjegnute upotrebu naredbe **break**.

2. Napisati program koji određuje i vraća koliko se u stringu **s** nalazi slova (malih i velikih), koliko cifara i koliko ostalih karaktera.

Rješenje:

```

01 | #include <stdio.h>
02 | #include <string.h>
03 | int main()
04 | {
05 |     char s[20];
06 |     int bm = 0, bv = 0, bo = 0, bc = 0, i;
07 |     puts("Unesite string"); // automatski dodaje novi red
08 |     gets(s); // koristimo u radu sa stringovima, preporučuje se fgets
09 |     // scanf("%s") čita do prvog razmaka ili Enter-a
10 |
11 |     for (i = 0; i < strlen(s); i++)
12 |         // alternativno, moglo se koristiti s[i] != '\0'
13 |         if (s[i] >= 'a' && s[i] <= 'z')
14 |             bm++;
15 |         else if (s[i] >= 'A' && s[i] <= 'Z')
16 |             bv++;
17 |         else if (s[i] >= '0' && s[i] <= '9')
18 |             bc++;
19 |         else
20 |             bo++;
21 |     printf("Broj velikih slova: %d\n", bv);
22 |     printf("Broj malih slova: %d\n", bm);
23 |     printf("Broj cifara: %d\n", bc);
24 |     printf("Broj ostalih: %d\n", bo);
25 | }

```

3. (Za vježbu) Napisati program koji svaki space (razmak) zamjenjuje karakterom #, te koji određuje i štampa sumu svih cifara iz datog stringa.
4. Napisati program čiji je ulazni argument string `s`, i koji ispituje da li je `s` palindrom, odnosno da li se jednako čita sa obje strane. Obratiti pažnju da položaj razmaka ne utiče na donošenje odluke da li je `s` palindrom i da se ne pravi razlika između malih i velikih slova.

Rješenje:

```

01 | #include <stdio.h>
02 | #include <string.h>
03 |
04 | int main()
05 | {
06 |     char s[20], s1[20];
07 |     int i, k = 0, ind;
08 |
09 |     puts("Unesite string:");
10 |     fgets(s, sizeof(s), stdin);
11 |
12 |     for(i = 0; i < strlen(s); i++)
13 |     {
14 |         if(s[i] >= 'a' && s[i] <= 'z')
15 |             s1[k++] = s[i];
16 |         else if(s[i] >= 'A' && s[i] <= 'Z')
17 |             s1[k++] = s[i] + ('a' - 'A');
18 |     }
19 |
20 |     ind = 0;
21 |     for(i = 0; i <= strlen(s1); i++)
22 |         if (s1[i] != s1[strlen(s1)-1 - i])
23 |             ind = 1;
24 |
25 |     if(ind == 0)
26 |         printf("Jeste palindrom.");

```

```

27 |         else
28 |             printf("Nije palindrom.");
29 |     }

```

Posebnu pažnju obratiti na liniju 17, gdje se radi konverzija velikih slova u mala slova, primjenom ASCII tabele. Za zadatke, potrebno je samo da zapamtite da prvo idu cifre, pa velika slova, pa onda mala slova u ASCII tabeli. Dakle, malim slovima odgovaraju najveće ASCII vrijednosti.

U nastavku je data i **alternativna izrada zadatka**. Uporediti za vježbu oba rješenja.

```

01 | #include <stdio.h>
02 | #include <string.h>
03 | int main()
04 | {
05 |     char s[20], s1[20];
06 |     int ind, i, k = 0;
07 |     puts("Unesite string");
08 |     fgets(s, sizeof(s), stdin);
09 |     // pretvaranje velikih u mala slova
10 |     for(i = 0; i < strlen(s); i++)
11 |     {
12 |         if(s[i] >= 'A' && s[i] <= 'Z')
13 |             s[i] = s[i] + 'a' - 'A';
14 |     }
15 |     // napravimo novi string sa samo malim slovima
16 |     for(i = 0; i < strlen(s); i++)
17 |         if(s[i] >= 'a' && s[i] <= 'z')
18 |             s1[k++] = s[i];
19 |     // ispitajmo čita li se jednako sa obje strane
20 |     ind = 0;
21 |     for(i = 0; i < strlen(s1); i++)
22 |         if(s1[i] != s1[strlen(s1)-1-i])
23 |             ind = 1;
24 |     if (ind == 0)
25 |         puts("String je palindrom!");
26 |     else
27 |         puts("String nije palindrom!");
28 | }

```

5. Napisati program koji sadrži funkciju `ima_li_5`, kojoj se proslijeđuje kao argument cio broj, i koja kao rezultat vraća 1, ako broj sadrži cifru 5, a vraća 0 ako cifra 5 ne postoji u posmatranom broju. U glavnom programu, korisnik unosi cio broj, pa se pozivom implementirane funkcije ispituje i štampa da li unijeti broj sadrži cifru 5.

Rješenje:

```

01 | #include <stdio.h>
02 |
03 | int ima_li_5(int);
04 |
05 | int main()
06 | {
07 |     int N;
08 |
09 |     printf("Unesite cio broj: ");
10 |     scanf("%d", &N);
11 |
12 |     if(ima_li_5(N) == 1)
13 |         printf("Broj sadrzi cifru 5.");
14 |     else
15 |         printf("Broj ne sadrzi cifru 5.");

```

```

16 | }
17 |
18 | int ima_li_5(int N)
19 | {
20 |     int ind, cif;
21 |     ind = 0;
22 |     while (N != 0)
23 |     {
24 |         cif = N%10;
25 |         if(cif == 5)
26 |             ind = 1;
27 |         N /= 10;
28 |     }
29 |     return ind;
30 | }

```

Napomenimo da se linije koda u funkciji nakon **return** ne izvršavaju. Ukoliko se **return** nalazi u petlji (unutar selekcije, odnosno **if** dijela, a ako se ispuni uslov iz selekcije, prekida se izvršavanje petlje u funkciji, i funkcija vraća odgovarajući rezultat). U tom smislu, **return** se ponaša kao **break**. Prethodni zadatak možemo riješiti **drugačije**, koristeći ovo svojstvo **return**-a, te na taj način možemo izbjeći upotrebu indikatorске varijable u funkciji. Alternativna, nešto elegantnija implementacija funkcije **ima_li_5** je data u nastavku:

```

01 | int ima_li_5(int a)
02 | {
03 |     int cif;
04 |     while (a != 0)
05 |     {
06 |         cif = a%10;
07 |         if (cif == 5)
08 |             return 1; // prekida petlju, a funkcija vraća 1 u main
09 |         a /= 10;
10 |     }
11 |     return 0;
12 | }

```

6. Napisati program koji sadrži funkciju **ima_li_5**, kojoj se proslijeđuje kao argument cio broj, i koja kao rezultat vraća 1, ako broj sadrži cifru 5, a vraća 0 ako cifra 5 ne postoji u posmatranom broju. U glavnom programu korisnik unosi niz cijelih brojeva **x**. Program koristeći funkciju računa sumu elemenata niza **x**, koji sadrže cifru 5.

Rješenje:

```

01 | #include <stdio.h>
02 |
03 | int ima_li_5(int);
04 |
05 | int main()
06 | {
07 |     int i, x[20], N, s;
08 |
09 |     s = 0;
10 |
11 |     printf("Unesite dužinu niza: ");
12 |     scanf("%d", &N);
13 |
14 |     printf("Unesite niz x: ");
15 |     for(i = 0; i < N; i++)
16 |         scanf("%d", &x[i]);
17 |

```

```
18 | // Određivanje sume je moglo biti odrađeno u sklopu gornje petlje.
19 |
20 | for(i = 0; i < N; i++)
21 |     if (ima_li_5(x[i]) == 1)
22 |         s += x[i];
23 |
24 | printf("Tražena suma je: %d", s);
25 | }
26 |
27 | int ima_li_5(int a)
28 | {
29 |     int cif;
30 |     while (a != 0)
31 |     {
32 |         cif = a%10;
33 |         if (cif == 5)
34 |             return 1;
35 |         a /= 10;
36 |     }
37 |     return 0;
38 | }
```