

2. masovne instrukcije 2010./11.

Programiranje i programsko inženjerstvo

Održali:

Kristijan Franković

Alen Rakipović

Siniša Matetić

Asistenti:

Željko Tomić

Renato Barukčić

18. studeni 2010.

1. zadatak

Učitavati cijele brojeve u polje dok se ne učitava nula (nulu ne učitavati u polje). Potom na ekran ispisati sve elemente polja čija je vrijednost manja od aritmetičke sredine polja! Najveći broj elemenata koji se mogu učitati iznosi 25. Ukoliko se odmah učitava nula, potrebno je ispisati poruku („Nije učitano ni jedan broj!“).

Primjer:

Ulaz: 1 2 3 4 5 6 7 8 9 10 11 12 0

Izlaz: 1 2 3 4 5 6

Rješenje:

```
#include <stdio.h>

int main ()
{
    int polje [25], broj, i=0, ucitano, suma=0;

    do
    {
        scanf ("%d", &broj);
        if (broj!=0)
        {
            polje[i] = broj;
            suma += broj;
            i++;
        }
    } while (i < 25 && broj != 0);

    if (i == 0)
    {
        printf ("Nije ucitano ni jedan broj\n");
        return 0;
    }

    ucitano = i;

    for (i=0;i<ucitano;i++)
        if (polje[i] < (float)suma/ucitano)
            printf ("%d ", polje[i]);

    printf ("\n");
    return 0;
}
```

2. zadatak

S tipkovnice učitavati cijele brojeve dok god se naizmjenice upisuje jedan paran, jedan neparan, jedan paran broj, itd. Ako se dva puta za redom upiše paran broj ili se dva puta za redom upiše neparan broj ili se upiše broj 0, program ispisuje najveći do tada upisani parni i neparni broj i prekida se njegovo izvršavanje. Ukoliko nije bio upisan ni jedan paran ili neparan broj, potrebno je dojaviti korisniku tu informaciju. Prvi broj koji se upiše s tipkovnice može biti ili paran ili neparan.

Primjer:

Ulaz: 1 2 3 4 5 6 7 26 25 25

Izlaz: Najveci uneseni parni broj iznosi: 26. Najveci uneseni neparni broj iznosi 25.

Rješenje:

```
#include <stdio.h>

int main ()
{
    int broj, najparni = -1, najneparni = -1, uvjet = -1;
    while (1)
    {
        scanf ("%d", &broj);
        if (broj == 0)
            break;
        if (broj < 0)
            continue;
        if (broj % 2 == 0)
        {
            if (najparni == -1)
                najparni = broj;
            if (najparni < broj)
                najparni = broj;
            if (uvjet == 0)
                break;
            uvjet = 0;
        }
        if (broj % 2 == 1)
        {
            if (najneparni == -1)
                najneparni = broj;
            if (najneparni < broj)
                najneparni = broj;
            if (uvjet == 1)
                break;
            uvjet = 1;
        }
    }
    if (najparni == -1)
        printf ("Nije bilo uneseno parnih brojeva!\n");
    else
        printf ("Najveci uneseni parni broj iznosi: %d\n", najparni);
    if (najneparni == -1)
        printf ("Nije bilo uneseno neparnih brojeva!\n");
    else
        printf ("Najveci uneseni neparni broj iznosi: %d\n", najneparni);
    return 0;
}
```

3. zadatak

Napišite program koji učitava broj na intervalu između 0 i 9 i na ekran ispisuje poruku „Učitani broj je manji od broja 9“ i tako redom za sve brojeve od kojih je učitani broj manji. Ukoliko se unese broj koji je izvan ovoga intervala dojaviti poruku o pogrešci i završiti rad programa. Ukoliko je uneseni broj bio upravo 9, tada ne treba ispisati ništa na ekran. Obavezno koristite naredbu switch-case i propadanje po labelama.

Primjer:

Ulaz: 6

Izlaz:

Ucitani broj je manji od broja 7.

Ucitani broj je manji od broja 8.

Ucitani broj je manji od broja 9.

Rješenje:

```
#include <stdio.h>

int main ()
{
    int broj;

    scanf ("%d", &broj);

    switch (broj)
    {
        case 0: printf ("Ucitani broj je manji od broja 1\n");
        case 1: printf ("Ucitani broj je manji od broja 2\n");
        case 2: printf ("Ucitani broj je manji od broja 3\n");
        case 3: printf ("Ucitani broj je manji od broja 4\n");
        case 4: printf ("Ucitani broj je manji od broja 5\n");
        case 5: printf ("Ucitani broj je manji od broja 6\n");
        case 6: printf ("Ucitani broj je manji od broja 7\n");
        case 7: printf ("Ucitani broj je manji od broja 8\n");
        case 8: printf ("Ucitani broj je manji od broja 9\n");
        case 9: break;
        default: printf ("Unijeli ste broj izvan intervala!\n");
    }
    return 0;
}
```

4. zadatak

Napišite program koji uz pomoć funkcije `gets` učitava polje duljine maksimalno 50 znakova (može se pretpostaviti da korisnik neće unijeti više). Za definiranje duljine obavezno koristiti simboličke konstante. Program kao rezultat treba ispisati učestalost pojavljivanja pojedinih slova. Pritom je potrebno brojati i velika i mala slova, ali brojači ne moraju biti razdvojeni. Sve ostale znakove zanemariti.

Primjer:

Ulaz: ABCabc <ENTER>

Izlaz:

Slovo A se pojavilo 2 puta!

Slovo B se pojavilo 2 puta!

Slovo C se pojavilo 2 puta!

Slovo D se pojavilo 0 puta!

...

Rješenje:

```
#include <stdio.h>
#define MAX 50
#define BR 26

int main ()
{
    char polje [MAX+1];
    int brojac [BR] = {0}, i, d=0;

    gets (polje);

    while (polje[d]!='\0')
        d++;

    for (i=0;i<d;i++)
    {
        if (polje[i] >= 'A' && polje[i] <= 'Z')
            brojac[polje[i]-'A']++;
        if (polje[i] >= 'a' && polje[i] <= 'z')
            brojac[polje[i]-'a']++;
    }

    for (i=0;i<BR;i++)
        printf ("Slovo %c se pojavilo %d puta!\n", 'A' + i, brojac[i]);

    return 0;
}
```

5. zadatak

Napišite program koji učitava dva niza znakova. Pritom se može pretpostaviti da prvi učitani niz neće biti veći od 50 znakova, a drugi će biti velik najviše 5 znakova. Učitavanje obaviti uz pomoć naredbe `scanf` sa formatom `%s`. Označimo prvi niz sa A, a drugi niz sa B. Pronaći sva pojavljivanja podniza B u nizu A i na ekran ispisati koliko je bilo ovakvih pojavljivanja! Može se pretpostaviti da je u nizu B barem jedan znak učitani!

Primjer:

Ulaz: anananalaaaaaana ana

Izlaz: Niz ana se u prvom nizu pojavio 4 puta.

Rješenje:

```
#include <stdio.h>
#include <string.h>
#define MAX 50

int main ()
{
    char nizA[50+1], nizB[5+1];
    int duljinaA, duljinaB, i, j, jednaki, brojac=0;

    scanf ("%s", nizA);
    scanf ("%s", nizB);

    duljinaA = strlen(nizA);
    duljinaB = strlen(nizB);

    for (i=0;i<duljinaA-duljinaB+1;i++)
    {
        jednaki = 1;
        for (j=0;j<duljinaB;j++)
            if (nizA[i+j] != nizB[j])
            {
                jednaki = 0;
                break;
            }
        if (jednaki == 1)
            brojac++;
    }

    printf ("Niz %s se u prvom nizu pojavio %d puta.\n", nizB, brojac);
    return 0;
}
```

6. zadatak

Napišite program koji učitava 10 cijelih brojeva i potom cijeli broj n. Program treba cirkularno pomaknuti cijelo polje ulijevo za broj mjesta koliko je bio uneseni cijeli broj n.

Primjer:

Ulaz: 0 1 2 3 4 5 6 7 8 9 **2**

Izlaz: 2 3 4 5 6 7 8 9 0 1

Rješenje:

```
#include <stdio.h>
#define DULJINA 10

int main ()
{
    int polje [DULJINA], i, x, pom, j;

    printf ("Unesite polje!\n");
    for (i=0;i<DULJINA;i++)
        scanf ("%d", &polje[i]);

    printf ("Unesite pomak x!\n");
    scanf ("%d", &x);

    x = x%10;
    for (i=0;i<x;i++)
    {
        pom = polje[0];
        for (j = 0; j < DULJINA-1; j++)
            polje[j] = polje [j+1];
        polje[DULJINA-1] = pom;
    }

    for (i=0;i<10;i++)
        printf ("%d ", polje[i]);
    printf ("\n");
    return 0;
}
```

7. zadatak

Pomoću naredbe gets unesite polje proizvoljne duljine. Može se pretpostaviti da neće biti učitano više od 80 znakova. Provjerite jeli učitani niz znakova palindrom!

Primjer:

Ulaz: ABBA

Izlaz: Ucitani niz znakova je palindrom.

Rješenje:

```
#define MAX 80

#include <stdio.h>
#include <string.h>
int main ()
{
    char polje [MAX];
    int i, d, palindrom = 1;
    gets (polje);
    d = strlen (polje);
    for (i=0;i<d/2;i++)
        if (polje[i] != polje[d-1-i])
        {
            palindrom = 0;
            break;
        }
    if (palindrom)
        printf ("Ucitani niz znakova je palindrom\n");
    else
        printf ("Ucitani niz znakova nije palindrom\n");

    return 0;
}
```


8. zadatak

Napišite program koji će uz pomoć funkcije `scanf` sa formatom `%s` učitati polje znakova. Program treba kopirati učitani niz u drugo polje i tamo na kraj polja za svako pojavljivanje samoglasnika staviti veliko slovo X (tako da se niz proširi). U originalnom nizu je potrebno izbaciti sva pojavljivanja samoglasnika. Dodatno, prilikom rada sa poljem znakova, obavezno je koristiti pokazivače. Može se pretpostaviti da će biti učitano najviše 50 znakova i da će sva slova koja će biti učitana biti velika slova (ali se mogu pojaviti i ostalo znakovi).

Primjer:

Ulaz: ABCDE

Izlaz: BCD ABCDEXX

Rješenje:

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

#define MAX 50

int main ()
{
    char poljeA [MAX+1], poljeB[2*MAX+1];
    int i,j, duljinaA, duljinaB;
    char *pa, *pb;

    pa = &poljeA[0];
    pb = &poljeB[0];

    scanf ("%s", pa);

    duljinaA = strlen (pa);

    for (i=0;i<=duljinaA;i++)
        *(pb+i) = *(pa+i);

    for (i=0;i<duljinaA;i++)
        if ((*pa+i) == 'A') || (*pa+i) == 'E' || (*pa+i) == 'I' || (*pa+i) == 'O' || (*pa+i)
        == 'U'))
        {
            for (j=i;j<duljinaA;j++)
                *(pa+j) = *(pa+j+1);
            i--;
            duljinaA--;
        }

    duljinaB = strlen (pb);
    j=duljinaB;

    for(i=0;i<duljinaB;i++)
    {
        if ((*pb+i) == 'A') || (*pb+i) == 'E' || (*pb+i) == 'I' || (*pb+i) == 'O' || (*pb+i)
        == 'U'))
        {
            *(pb+j) = 'X';
            j++;
            *(pb+j) = '\0';
        }
    }

    printf ("%s\n", pa);
    printf ("%s\n", pb);

    return 0;
```

9. zadatak

Unesite kvadratnu matricu. Veličina matrice je proizvoljna, ali može se pretpostaviti da neće biti veća od 8. Potrebno je provjeriti ispravnost učitanih dimenzija!

Potom je potrebno obaviti slijedeći niz operacija: 1. Zamijeniti sadržaj prvog i zadnjeg retka matrice. 2. Zamijeniti sadržaj prvog i zadnjeg stupca matrice, 3. Zamijeniti sadržaj glavne i sporedne dijagonale. Nakon obavljenih transformacija matricu ispisati na ekran.

Primjer:

Ulaz: 3 1 2 3 4 5 6 7 8 9

Izlaz:

7 8 9

6 5 4

1 2 3

Rješenje:

```
#include <stdio.h>
#define MAX 8

int main ()
{
    int matrica [MAX][MAX], m,n,i,j, pom;

    scanf ("%d", &m);
    while (m<1 || m> MAX)
    {
        printf ("Unijeli ste neispravne vrijednost! Unesite ponovo!\n");
        scanf ("%d", &m);
    }

    for (i=0;i<m;i++)
        for (j=0;j<m;j++)
            scanf ("%d", &matrica[i][j]);

    for (i=0;i<m;i++)
    {
        for (j=0;j<m;j++)
            printf ("%5d", matrica[i][j]);
        printf ("\n");
    }

    for (i=0;i<m;i++)
    {
        pom = matrica[0][i];
        matrica[0][i] = matrica[m-1][i];
        matrica[m-1][i] = pom;
    }

    for (i=0;i<m;i++)
    {
        pom = matrica[i][0];
        matrica[i][0] = matrica[i][m-1];
        matrica[i][m-1] = pom;
    }
}
```

```

for (i=0;i<m;i++)
{
    pom = matrica[i][i];
    matrica[i][i] = matrica[i][m-1-i];
    matrica[i][m-1-i] = pom;
}

printf ("\n");
for (i=0;i<m;i++)
{
    for (j=0;j<m;j++)
        printf ("%5d", matrica[i][j]);
    printf ("\n");
}

return 0;
}

```

10. zadatak

Učitati matricu proizvoljnih dimenzija (maksimalno 30x30) i ispisati indeks retka koji ima najveću sumu! Pritom je nužno koristiti pokazivače za baratanje matricom. Također, koristite simboličke konstante. Nije potrebno provjeriti ispravnost unesenih vrijednosti već pretpostaviti da će iste biti unesene. Ukoliko dva retka imaju jednaku sumu ispisati indeks onog retka koji ima veću vrijednost indeksa.

Primjer:

Ulaz:

Izlaz:

Rješenje:

```

#include <stdio.h>
#define MAX 30

int main ()
{
    int matrica[30][30],m,n,i,j, najSuma = 0, suma=0, najRedak = 0;
    printf ("Unesite broj redaka i stupaca\n");
    scanf ("%d %d", &m, &n);

    for (i=0;i<m;i++)
        for (j=0;j<n;j++)
            scanf ("%d", &matrica[i][j]);

    for (i=0;i<n;i++)
        najSuma += matrica[0][i];

    for (i=1;i<m;i++)
    {
        suma = 0;
        for (j=0;j<n;j++)
            suma += matrica[i][j];
        if (suma >= najSuma)
        {
            najSuma = suma;
            najRedak = i;
        }
    }
    printf ("Indeks retka sa najvecom sumom je %d.", najRedak);
    return 0;
}

```

11. zadatak

Napišite program koji učitava kvadratnu matricu proizvoljnih dimenzija. Vrijednosti dimenzije zadati preko tipkovnice, a može se pretpostaviti da matrica neće imati više od 8 redaka odnosno 8 stupaca. Također, minimalan broj redaka i stupaca je jedan. U slučaju da su unesene vrijednosti izvan ovog intervala, korisniku dojaviti upozorenje i ponuditi ponovni unos vrijednosti.

Potom za unesenu matricu ispitati je li pseudodeterminanta matrice jednaka nula. Pritom je dovoljno ispitati ima li matrica dva jednaka stupca ili dva jednaka retka. Ukoliko takvi postoje, pseudodeterminatna te matrice je nula.

Primjer:

Ulaz:

Izlaz:

Rješenje:

```
#include <stdio.h>
#define MAX 8

int main ()
{
    int matrica [MAX][MAX], m,n,i,j,k, pom, jeNula;

    scanf ("%d", &m);
    while (m<1 || m> MAX)
    {
        printf ("Unijeli ste neispravne vrijednost! Unesite ponovo!\n");
        scanf ("%d", &m);
    }

    for (i=0;i<m;i++)
        for (j=0;j<m;j++)
            scanf ("%d", &matrica[i][j]);

    for (i=0;i<m;i++)
    {
        for (j=i+1;j<m;j++)
        {
            /*po retcima*/
            jeNula = 1;
            for (k=0;k<m;k++)
                if (matrica[i][k] != matrica[j][k])
                {
                    jeNula = 0;
                    break;
                }
            if (jeNula == 1)
                break;

            /*po stupcima*/
            jeNula = 1;
            for (k=0;k<m;k++)
                if (matrica[k][i] != matrica[k][j])
                {
                    jeNula = 0;
                    break;
                }
        }
    }
}
```

```

        if (jeNula == 1)
            break;
    }
    if (jeNula == 1)
        break;
}

if (jeNula == 1)
    printf ("Pseudodeterminanta matrice iznosi nula!\n");
return 0;
}

```

12. zadatak

Napišite funkciju koja prima dva cijela broja. Pritom je prvi navedeni broj donja granica, a drugi navedeni broj gornja granica. Funkcija na ekran ispisuje sve parne brojeve smještene između donje i gornje granice, granice isključene.

Primjer:

Ulaz: 3 9

Izlaz: 4 6 8

Rješenje:

```

#include <stdio.h>

void ispisiParne (int donja, int gornja)
{
    int i;
    if (donja % 2 == 1)
        donja +=1;
    else
        donja += 2;
    for (i=donja; i<gornja; i=i+2)
        printf ("%d ", i);
    return;
}

int main ()
{
    int donja, gornja;
    scanf ("%d %d", &donja, &gornja);
    ispisiParne (donja, gornja);
    printf ("\n");
    return 0;
}

```

13. zadatak

Napišite funkciju koja kao parametar prima cijeli broj. Funkcija treba provjeriti jeli broj Fibonaccijev. Ukoliko je broj fibonaccijev preko imena funkcije vraća vrijednost 1, a 0 inače.

Primjer:

Ulaz: 13

Izlaz: Uneseni broj je fibonaccijev.

Rješenje:

```
#include <stdio.h>

int jeliFibonacci (int broj)
{
    int prvi = 1, drugi = 1, trenutniFib;
    if (broj == 1)
        return 1;

    while (1)
    {
        trenutniFib = prvi + drugi;
        prvi = drugi;
        drugi = trenutniFib;
        if (broj == trenutniFib)
            return 1;
        if (broj < trenutniFib)
            return 0;
    }
}

int main ()
{
    int broj;
    scanf ("%d", &broj);
    if (jeliFibonacci(broj))
        printf ("Uneseni broj je fibonaccijev.");
    else
        printf ("Uneseni broj nije fibonaccijev.");
    return 0;
}
```

14. zadatak

Napišite funkciju koja prima parametre m i n kao cijeli broj. Funkcija vraća m povrh n kao cijeli broj. Pritom napišite i vlastitu funkciju za izračun $n!$ (n faktoriijela) i koristite je za izračun gdje je to potrebno. Također potrebno je napisati i glavni program sa primjerom poziva funkcije. Dodatno, možete pretpostaviti da će biti unesene ispravne vrijednosti m i n .

Primjer:

Ulaz: 5 3

Izlaz: 10

Rješenje:

```
#include <stdio.h>

int faktoriijela (int broj);

int mpovrh (int m, int n);

int main ()
{
    int m, n;
    scanf ("%d %d", &m, &n);
    printf ("%d povrh %d iznosi %d\n", m,n, mpovrh(m,n));
    return 0;
}

int faktoriijela (int broj)
{
    int umnozak = 1;
    while (broj > 0)
    {
        umnozak *=broj;
        broj--;
    }
    return umnozak;
}

int mpovrh (int m, int n)
{
    return faktoriijela(m)/((faktoriijela(n)*faktoriijela(m-n)));
}
```

15. zadatak

Napišite funkciju koja prima dva cijela broja kao parametre i putem imena funkcije vraća pokazivač na memorijsku lokaciju gdje je smješten veći od ta dva broja. Također napisati glavni program koji demonstrira rad funkcije.

Primjer:

Ulaz: 1 2

Izlaz: Adresa: <neka proizvoljna adresa> Vrijednost: 2

Rješenje:

```
#include <stdio.h>

int* vratiVeci (int *a, int *b)
{
    if (*a > *b)
        return a;
    else if (*b > *a)
        return b;
    else
        return NULL;
}

int main ()
{
    int a, b;
    int *p;
    scanf ("%d %d", &a, &b);
    p = vratiVeci(&a,&b);
    if (p == NULL)
        printf ("Brojevi su jednaki!\n");
    else
        printf ("Adresa: %p Vrijednost: %d\n", p, *p);
    return 0;
}
```


16. zadatak

Napisati funkciju koja prima pozitivan cijeli broj i putem pokazivača u pozivajući program vraća aritmetičku sredinu i broj znamenaka te vrijednost 1 preko imena funkcije. Ukoliko je bio upisan negativan broj, funkcija vraća 0 preko imena funkcije. Napisati i glavni program koji demonstrira upotrebu funkcije.

Primjer:

Ulaz: 1234

Izlaz: Aritmeticka sredina znamenaka broja 1234 iznosi 2.500000 a broj znamenki 4

Rješenje:

```
#include <stdio.h>

int funkcija (int broj, float* arSredina, int* brojZnamenaka)
{
    int suma = 0;

    if (broj < 0)
        return 0;

    *brojZnamenaka = 0;
    while (broj > 0)
    {
        suma += broj%10;
        broj = broj/10;
        (*brojZnamenaka) ++;
    }

    *arSredina = (float)suma / *brojZnamenaka;

    return 1;
}

int main ()
{
    int broj, brojac;
    float sredina;

    scanf ("%d", &broj);
    if (funkcija(broj, &sredina, &brojac))
        printf ("Aritmeticka sredina znamenaka broja %d iznosi %f a broj
znamenki %d\n", broj, sredina, brojac);
    else
        printf ("Unijeli ste negativan broj!\n");

    return 0;
}
```

Dodatak

Znakovne nizove možemo učitavati na tri načina:

1. Pomoću funkcije `scanf` i formata `%c`.

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

int main ()
{
    char polje[50+1];
    int i=0;
    while (scanf ("%c", &polje[i]))
    {
        if (polje[i] == 'X')
            break;
        i++;
    }
    polje[i] = '\0';
    puts (polje);
    return 0;
}
```

Iznad navedeni način je vrlo dobar ukoliko je potrebno učitavati znakove dok se ne unese neki znak, konkretno u ovom slučaju veliko slovo X. U ovom slučaju se čak i pritisak na tipku enter učitava u polje (kao `'\n'`), a jedini znak koji će prekinuti unos kada se na njega naiđe je X. Dodatno može se postaviti i uvjet koji će prekinuti izvođenje petlje ukoliko se učitava previše znakova (više od 50). Također, sami moramo u ovome slučaju na odgovarajuće mjesto postaviti `'\0'`. Ovdje to činimo tako da se zadnji učitani znak – slovo X ne stavi u niz (preko njega stavimo null karakter).

Postoji i drugi način dohvaćanja znak po znak, pomoću funkcije `getchar ()`, ali o njoj više drugom prilikom.

2. Pomoću funkcije `scanf` i formata `%s`.

```
scanf ("%s", polje);
```

Ovakav način unosa učitava do prvog razmaka. Dakle ukoliko trebate unijeti rečenicu, ovakav način unosa i nije odveć pogodan, jer će primjerice „Medo bere jagode.“ učitati samo do Medo. Potom će postaviti null karakter na kraj niza.

3. Pomoću funkcije `gets`.

```
scanf ("%s", polje);
```

Funkcija `gets` je pogodnija za učitavanje rečenica jer ona čeka prvi pritisak na tipku enter. Dakle učitava i razmake. Pritom sama stavlja null-karakter.

Također, znakovne nizove možemo ispisivati na jedan od ova tri načina:

1. Pomoću funkcije `print` i formata `%c`.

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

int main ()
{
    char polje[50+1];
    int i = 0;
    gets(polje);

    while (polje[i] != '\0')
    {
        printf ("%c", polje[i]);
        i++;
    }

    return 0;
}
```

2. Pomoću funkcije `printf` i formata `%s`.

```
printf ("%s", polje);
```

Vrijedi naglasiti da funkcija `printf` sa formatom `%s` ispisuje sve iz polja do prvog null-karaktera. Dakle ukoliko se u polje nalaze razmaci i oni će biti ispisani (za razliku od `scanf`-a sa ovim formatom koji ne učitava razmake).

3. Pomoću funkcije `puts`.

```
puts (polje);
```

Funkcija `puts` ispisuje na ekran polje znakova do null-karaktera.

Bitno je primjetiti da je u sva tri načina nužno da se na kraju znakovnog niza nalazi null-karakter `'\0'`. U suprotnom može nam se desiti da se uz naš znakovni niz ispiše gomila smeća.

Ukoliko želimo izbrojati unesene znakove to možemo učiniti na dva načina:

1. Pomoću funkcije `strlen`. U tom slučaju nužno je uključiti biblioteku `string.h`

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

int main ()
{
    char polje[50+1];
    int duljina;
    gets(polje);
    duljina = strlen (polje);
    printf ("%d\n", duljina);
    return 0;
}
```

2. Pomoću odsječka koda koji sami napišemo:

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

int main ()
{
    char polje[50+1];
    int duljina = 0;
    gets (polje);
    while (polje[duljina] != '\0')
        duljina++;
    printf ("%d\n", duljina);
    return 0;
}
```

Pritom napomenimo da je česta greška da se ovdje upotrijebi operator `sizeof`. Isti vraća koliko smo memorije zauzeli sa deklaracijom polja znakova, ali ne i koliko je zaista znakova uneseno putem tipkovnice.

Dodatno, vrijedi ista opaska kao i za ispis znakovnih nizova – naše polje mora na ispravnom mjestu imati zaključni null karakter.

Ovdje prikazani načini učitavanja, ispisivanja i brojanja učitanih znakova nisu jedini koje možemo upotrijebiti, ali su najčešći.