
Zadaci za vježbu za 2.MI

1. primjer

Napisati program koji učitava cijele brojeve. Učitavanje treba prekinuti kada se učitava broj 0 (taj se broj ne uzima u obzir). Potrebno je ispisati broj učitanih peteroznamenkastih, četveroznamenkastih, troznamenkastih, dvoznamenkastih i jednoznamenkastih brojeva.

```
#include <stdio.h>

int main()
{
    int brojacZn[5] = {0};
    int broj;

    while (1)
    {
        scanf ("%d", &broj);

        if (broj == 0)
            break;

        if (broj < 0)
            broj = -broj;
```

1. primjer (nastavak)

```
if ( broj / 10000 >= 10)
    continue;
if ( broj / 10000 >= 1)
    brojacZn[0] ++;
else if ( broj / 1000 >= 1)
    brojacZn[1] ++;
else if ( broj / 100 >= 1)
    brojacZn[2] ++;
else if ( broj / 10 >= 1)
    brojacZn[3] ++;
else
    brojacZn[4] ++;
} /* kraj while petlje */

printf ("Peteroznamenkastih ima : %d\n", brojacZn[0]);
printf ("Cetveroznamenkastih ima : %d\n", brojacZn[1]);
printf ("Troznamenkastih ima : %d\n", brojacZn[2]);
printf ("Dvoznamenkastih ima : %d\n", brojacZn[3]);
printf ("Jednoznamenkastih ima : %d\n", brojacZn[4]);

return 0;
}
```

1. primjer (nastavak) - poboljšano rješenje

```
#include <stdio.h>
#include <math.h>
int main(){
    int brojacZn[5] = {0};
    int broj, brZnam, i;

    while (1) {
        scanf ("%d", &broj);
        if (broj == 0)
            break;
        if (broj < 0)
            broj = -broj;
        if (broj < 100000) {
            brZnam=0;
            while(broj){
                broj/=10;
                ++brZnam;
            }
            brojacZn[ brZnam - 1 ]++; /* ili: brojacZn[ (int) log10(broj) ]++; */
        }
    }
    for (i=4; i>=0; --i){
        printf ("Broj %d-znamenkastih je: %d\n", i+1, brojacZn[i]);
    }
    return 0;
}
```

2. primjer

Napisati program koja učitava dva cijela broja i ispisuje uniju znamenki učitanih brojeva. Pretpostaviti da brojevi nemaju više od 9 znamenki. Znamenke treba ispisati poredane po veličini uzlazno. Svaku znamenku treba ispisati samo jedanput.

Npr. za brojeve 2482 i 335228, treba ispisati 2 3 4 5 8

```
#include <stdio.h>
#define MAX 100
int main(){
    int prviBroj, drugiBroj, i;
    int brojacZnamenki[10] = {0};
    printf("Ucitajte prvi broj: ");
    scanf("%d", &prviBroj);
    printf("Ucitajte drugi broj: ");
    scanf("%d", &drugiBroj);
```

2. primjer (nastavak)

```
    int i = 0;
    /*
    Uzima se znamenka s "kraja" broja uz pomoć modulo operacije
    te povećava vrijednost elementa polja koji je na poziciji
    vrijednosti uzete znamenke
    */
    do{
        brojacZnamenki[prviBroj%10]++;
        prviBroj/=10; /* Zadnja se znamenka miče */
    } while(prviBroj>0);

    do{
        brojacZnamenki[drugiBroj%10]++;
        drugiBroj/=10;
    } while(drugiBroj>0);

    for(i=0; i<10; i++){
        if(brojacZnamenki[i]) printf("%d ", i);
    }
}
```

3. primjer

Napisati program koja učitava dva znakovna niza i ispisuje slova koja se nalaze i u jednom i u drugom nizu. Pretpostaviti da nizovi sadrže samo velika slova engleske abecede.

Svako zajedničko slovo treba ispisati samo jedanput.

Primjer: za nizove AABCD A i DAAD treba ispisati AD

```
#include <stdio.h>
#define MAX 100
int main() {
    char niz1[MAX], niz2[MAX];
    int i, j;
    int brojac['Z'-'A'+1] = {0};

    /* Buduci da u zadatku pise da se unose samo velika slova (a
    razmak NIJE veliko slovo) mozemo koristiti %s.*/

    printf("\nUnesite prvi niz:");
    scanf("%s", niz1);
    printf("\nUnesite drugi niz:");
    scanf("%s", niz2);
```

3. primjer (nastavak)

```
/* Moramo usporedjivati "svaki sa svakim". */
for (i=0; niz1[i] != 0; ++i){
    for (j=0; niz2[j] != 0; ++j){
        if (niz1[i] == niz2[j]){
/* Povecavamo brojac koji odgovara zajednickom slovu */
            brojac[niz1[i]-'A']++;
        }
    }
}

printf("\nZajednicki znakovi su:");
for (i=0; i < 'Z'-'A'+1; ++i){
    if (brojac[i] > 0){
        printf("%c", 'A' + i);
    }
}
return 0;
}
```

3. primjer (nastavak)- korištenje pokazivača

```
/* Moramo usporedjivati "svaki sa svakim". */
for (i=0; *(niz1+i) != 0; ++i){
    for (j=0; *(niz2+j) != 0; ++j){
        if (*(niz1+i) == *(niz2+j)){
/* Povecavamo brojac koji odgovara zajednickom slovu */
            brojac[*(niz1+i) - 'A']++;
/*
            (*(brojac + (*(niz1+i) - 'A')))+++;    */
        }
    }
}

printf("\nZajednicki znakovi su:");
for (i=0; i < 'Z'-'A'+1; ++i){
    if (brojac[i] > 0){ /* if (*(brojac+i))>0 */
        printf("%c", 'A' + i);
    }
}
return 0;
}
```

4. primjer

Napisati program koji će učitati cijeli broj n ($1 \leq n \leq 10$), ukoliko unesena vrijednost nije unutar intervala, učitavanje treba ponavljati sve dok se ne unese ispravna vrijednost) i zatim dvodimenzionalno polje znakova dimenzija $N \times N$ popuniti naizmjenice s '#' i '_' kako bi se dobila "šahovnica" (u gornjem lijevom kutu je uvijek '#').

Npr.

#	_	#	_
_	#	_	#
#	_	#	_
_	#	_	#

4. primjer (nastavak)

- gornji lijevi ugao uvijek započinjemo s '#'
- da li se može izračunati kakva formula s obzirom na indekse matrice?

	0	1
0	#	_
1	_	#

- svaki parni redak počinjemo s '#'
- unutar retka alterniramo '#' i '_'
- Dakle:
- ILI
- ako je zbroj indeksa retka i indeksa stupca paran onda '#' inače '_'

	0	1	2
0	#	_	#
1	_	#	_
2	#	_	#

```
za sve retke{
    odredi početni znak u retku
    za sve stupce{
        dodijeli znak;
        alterniraj znak;
    }
}
```

4. primjer (nastavak)

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

int main(){
    char m[MAX][MAX];
    int n, i, j, zn;
    do {
        printf("Unesite n (1<=n<=10):");
        scanf("%d", &n);
    } while (!(n>=1 && n<=10));
    for (i=0; i<n; ++i){
        zn = (i%2 == 0) ? '#' : '_';
        for (j=0; j<n; ++j){
            m[i][j] = zn;
            zn = (zn == '#') ? '_' : '#';
            printf("%c ", m[i][j]);
        }
        printf("\n");
    }
}
```

4. primjer (nastavak)-poboljšano rješenje

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

int main(){
    char m[MAX][MAX];
    int n, i, j;
    do {
        printf("Unesite n (1<=n<=10):");
        scanf("%d", &n);
    } while (!(n>=1 && n<=10));

    for (i=0; i<n; ++i){
        for (j=0; j<n; ++j){
            m[i][j] = ((i+j)%2 == 0) ? '#' : '_';
            printf("%c ", m[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```

5. primjer

Napisati program koji učitava dimenzije cjelobrojne matrice (broj redaka ≤ 10 ; broj stupaca ≤ 10), te članove matrice, a zatim ispisuje matricu, te broj redaka matrice koji imaju sljedeće svojstvo: svaki element retka (osim prvoga) je za jedan veći od prethodnog elementa u retku.

```
#include <stdio.h>
#define MAXRED 10
#define MAXSTUP 10

int main () {
    int mat[MAXRED][MAXSTUP];
    int brRed, brStup;

    int brojDobrihRedaka = 0;
    int dobarRedak;
    int i, j;
    do {
        /* učitati dimenzije matrice */
        scanf("%d %d", &brRed, &brStup);
        while (brRed<1 || brRed>10 || brStup<1 || brStup>10)
            /* while (!(brRed>=1 && brRed<=10 && brStup>=1 && brStup<=10)) */
    }
```

5. primjer (nastavak)

```
/* učitati elemente matrice */
for (i = 0; i < brRed; i++) {
    for (j = 0; j < brStup; j++) {
        scanf("%d", &mat[i][j]);
    }
}
/* provjeriti za svaki redak ispunjava li zadani uvjet */
for (i = 0; i < brRed; i++) {
    dobarRedak = 1;
    /* je li j. clan za 1 veci od (j-1). clana*/
    for (j = 1; j < brStup && dobarRedak == 1; j++) {
        if (mat[i][j] != mat[i][j - 1] + 1) {
            dobarRedak = 0;
        }
    }
    /*ako su svi clanovi retka ispunili uvjet, uvecaj brojac za 1*/
    if (dobarRedak == 1) {
        ++ brojDobrihRedaka;
    }
    dobarRedak = 1;
}
```

5. primjer (nastavak)

```
/*ispis matrice */
for (i = 0; i < brRed; i++) {
    for (j = 0; j < brStup; j++) {
        printf("%d ", mat[i][j]);
    }
    printf("\n");
}

printf("Broj redaka: %d", brojDobrihRedaka);
return 0;
}
```

6. primjer

Napisati program koji učitava cijeli broj koji nema više od 9 znamenki. Ako broj ima 5 znamenki, treba ispisati umnožak znamenki. Ako broj ima 4 znamenke, treba ih redom oduzeti (prva znamenka-druga znamenka- treća znamenka-četvrta znamenka). Ako broj ima 3 znamenke, treba ispisati broj bez središnje znamenke. Ako broj ima 2 znamenke, treba ispisati broj koji čine obrnute znamenke učitano broj. Ako broj ima 1 znamenku, treba ispisati kvadrat te znamenke. Ako broj ima više od šest znamenki, potrebno je ispisati prve dvije znamenke učitano broj.

Napomena: zadatak riješiti bez korištenja if naredbe.

```
#include <stdio.h>

int main()
{
    int broj, znamenke[9]={0}, i=0, brZnam=0, produkt=1;
    scanf("%d", &broj);
    /*na mjestu i u polju znamenke je znamenka na mjestu i u učitano broj*/
    do{
        znamenke[i]=broj%10;
        broj=broj/10;
        i++;
        brZnam++;
    } while (broj!=0);
```

6. primjer (nastavak)

```
switch (brZnam)
{
    case 5:
        for (i=0;i<5; i++) produkt*=znamenke[i];
        printf("%d", produkt);
        break;

    case 4:
        broj = znamenke[3]-znamenke[2]-znamenke[1]-znamenke[0];
        printf ("%d", broj);
        break;

    case 3:
        broj = znamenke[2]*10 + znamenke[0];
        printf ("%d", broj);
        break;
```

6. primjer (nastavak)

```
case 2:
    broj = znamenke[0]*10 + znamenke[1];
    printf ("%d", broj);
    break;

case 1:
    printf ("%d",znamenke[0]*znamenke[0]);
    break;

default:
    printf ("%d %d", znamenke[brZnam-1], znamenke[brZnam -2]);
}

return 0;

}
```

7. primjer

Napisati program koji će sve elemente zadanog cjelobrojnog polja p od N elemenata posmaknuti za određeni broj mjesta posmak ($\text{posmak} < N$). Ukoliko je posmak pozitivan, elementi polja posmiču se udesno, a ukoliko je posmak negativan, elementi polja posmiču se ulijevo. Ispražnjeni elementi pune se nulama.

Primjer posmaka za $\text{posmak} = 2$ i polje [10 40 50 60 12]

Polje prije posmaka: [10 40 50 60 12]

Polje nakon posmaka: [0 0 10 40 50]

```
#include <stdio.h>
#define N 100

int main()
{
    int p[N], posmak, brojelem, i;

    do{
        printf("Unesite broj elemenata polja: ");
        scanf("%d", &brojelem);
    } while (brojelem<1 || brojelem>N);
```

7. primjer (nastavak)

```
printf("Unesite clanove polja: ");
for (i=0; i<brojelem; i++ ){
    scanf("%d",&p[i]);
}

do{
    printf("Unesite posmak: ");
    scanf("%d", &posmak);
}while (posmak>N || posmak<-N );

/* ako je posmak udesno */
if (posmak > 0) {
    for (i = N - 1 - posmak; i >= 0; i --)
        p[i + posmak] = p[i];
    for (i = posmak - 1; i >= 0; i --)
        p[i] = 0;
}
```

7. primjer (nastavak)

```
/* ako je posmak ulijevo */
else if (posmak < 0) {
    posmak = -posmak;
    for (i = 0; i <= N - 1 - posmak; i++)
        p[i] = p[i + posmak];
    for (i = N - posmak; i < N; i++)
        p[i] = 0;
}

printf("\n");
for (i=0; i<brojelem; i++){
    printf("%d ", p[i]);
}
}
```

7. primjer (nastavak)- rješenje s pokazivačima

```
int main()
{
    int p[N]={0}, posmak, brojelem, i;
    do{
        printf("Unesite broj elemenata polja: ");
        scanf("%d", &brojelem);
    } while (brojelem<1 || brojelem>N);

    printf("Unesite clanove polja: ");
    for (i=0; i<brojelem; i++ ){
        scanf("%d", p+i);
    }

    do{
        printf("Unesite posmak: ");
        scanf("%d", &posmak);
    } while (posmak>=N);
}
```

7. primjer (nastavak)- rješenje s pokazivačima

```
/* ako je posmak udesno */
if (posmak > 0) {
    for (i = N - 1 - posmak; i >= 0; i --)
        *(p+i+posmak) = *(p+i);
    for (i = posmak - 1; i >= 0; i --)
        *(p+i) = 0;
} /* ako je posmak ulijevo */
else if (posmak < 0) {
    posmak = -posmak;
    for (i = 0; i <= N - 1 - posmak; i++)
        *(p+i) = *(p+i+posmak);
    for (i = N - posmak; i < N; i++)
        *(p+i) = 0;
}
printf("\n");
for (i=0; i<brojelem; i++) {
    printf("%d ", *(p+i));
}
}
```