

## Zadaci za vježbu za 2. međuispit iz Programiranja i programskog inženjerstva

### Zadatak 1.

Napisati program koji izvršava kompresiju ulaznog niza. Program učitava niz prirodnih brojeva (učitavanje se završava kada se učita broj -1 koji se zanemaruje) koji predstavlja niz kojega treba komprimirati. Maksimalna veličina ulaznog niza je 100 brojeva. Niz se komprimira na sljedeći način: slijed ponavljanja istog broja unutar niza se zamjenjuje parom brojeva od kojih je prvi sam broj, a drugi predstavlja broj uzastopnih ponavljanja tog broja. Npr. za ulazni niz 1 1 2 5 5 5 5 1 1 1 -1, izlazni (komprimirani niz) je slijedeći : 1 2 2 1 5 4 1 3. Učitani niz se komprimira te kao takav zapisuje u novo polje. Potrebno je ispisati novodobiveno polje.

### Zadatak 2.

Napisati program koji učitava niz cijelih brojeva (maksimalne veličine 50) i pohranjuje ga u jednodimenzionalno polje. Učitavanje se završava kada se učita broj -1 koji se zanemaruje. Učitani niz treba prepisati u novo polje, ali s obrnutim redoslijedom elemenata. Program treba ispisati tako dobiveno polje. Npr. ako je ulazni niz 2 -3 5 7, izlazni niz je 7 5 -3 2.

### Zadatak 3.

Učitati parametre kružnice ( $r$  i koordinate središta), te zatim učitavati točke koordinatnog sustava (realni brojevi). Za učitane točke provjeriti nalaze li se na kružnici. Nakon što se učita točka  $(0, 0)$ , ispisati koliki je omjer pogodaka kružnice ((koliki je broj točaka na kružnici)/(ukupni broj točaka)).

### Zadatak 4.

Učitati parametre elipse (koordinate dvije točke fokusa i realna vrijednost duljine male poluosi), te zatim učitavati točke koordinatnog sustava (realni brojevi). Za učitane točke provjeriti nalaze li se na elipsi. Nakon što se učita točka  $(0, 0)$ , ispisati koliki je omjer pogodaka elipse ((koliki je broj točaka na elipsi)/(ukupni broj točaka)).

### Zadatak 5.

Napisati program koja učitava dva cijela broja i ispisuje najmanji zajednički višekratnik tih dvaju brojeva.

### Zadatak 6.

Napisati program koji učitava dva cijela broja i ispisuje njihov najveći zajednički djelitelj korištenjem Euklidovog algoritma.

Euklidov algoritam ( $a, b$ ):

```
dok (b>0)
    p=a;
    a=b;
    b=p mod b;
NZD=a;
```

### Zadatak 7.

Napisati program koji učitava dva cijela broja i ispisuje uniju znamenki učitanih brojeva. 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.

**Zadatak 8.**

Učitava se niz znakova čije su vrijednosti iz skupa  $\{0,1\}$  dok se ne pročita znak koji nije element skupa. Tako učitani niz predstavlja binarni ključ za kodiranje niza. Nakon toga se učitava niz znakova (čije su vrijednosti isto iz skupa  $\{0,1\}$ ) kojega treba kodirati izvođenjem operacije XOR nad bitovima koji se nalaze na istoj poziciji unutar tog niza i kodne riječi. Kodna riječ se dobiva na slijedeći način:

- ako je ključ dulji od učitano g niza koji ima duljinu od  $n$  bitova, kodnu riječ predstavlja prvih  $n$  bitova ključa
- ako je ključ jednake duljine kao i učitani niz, kodna riječ je identična ključu
- ako je ključ duljine  $n$ , a učitani niz duljine  $m$  bitova, i vrijedi  $m > n$ , kodna riječ se dobiva potrebnim brojem ponavljanja ključa, t.d. se dobije kodna riječ duljine  $m$  (npr. ključ je 110, a učitani niz 1101010010011 - kodna riječ je 1101101101101)

**Zadatak 9.**

Napisati program koji učitava prirodni broj  $n$  koji predstavlja rang kvadratne matrice, a zatim i cjelobrojne elemente matrice. Program treba izvršiti transponiranje učitane matrice te ispisati tako dobivenu transponiranu matricu.

**Zadatak 10.**

Napisati program koji učitava  $n$  (učitava se kao prirodni broj s tipkovnice) grupa od tri znaka; svaki znak može biti 'S', 'P', 'O' ili '-'. ('S' predstavlja subjekt, 'O' predstavlja objekt, 'P' predstavlja predikat, a '-' predstavlja nedefiniranu vrijednost). Svaka grupa znakova predstavlja rečenicu. Pretpostavimo da je rečenica ispravna ako sadrži barem subjekt i predikat ili predikat i objekt. Redoslijed predikat/subjekt/objekt u rečenici nije bitan. Potrebno je ispisati sve redne brojeve (indekse koji počinju od jedinice) neispravnih rečenica.

**Zadatak 11.**

Napisati program koji će učitati dva niza znakova i ispisati slova koja se nalaze i u jednom i u drugom nizu. Pretpostaviti da nizovi sadrže samo velika slova. Svako zajedničko slovo treba ispisati samo jedanput.

**Zadatak 12.**

Napisati program koji će učitati cijele brojeve  $m$  i  $n$  ( $1 \leq m \leq 20$ ,  $1 \leq n \leq 10$ , *ukoliko unesena vrijednost nije unutar intervala, učitavanje treba ponavljati sve dok se ne unese ispravna vrijednost*) i elemente matrice dimenzija  $m \times n$ , a zatim ispisati indekse onih redaka koji imaju sljedeće svojstvo: svaki element retka (osim prvoga) je za jedan veći od prethodnog elementa u retku.

**Zadatak 13.**

Napisati glavni program u kojem je potrebno učitati rang matrice  $r$ ,  $1 \leq r \leq 20$ . (*ukoliko unesena vrijednost nije unutar intervala, učitavanje treba ponavljati sve dok se ne unese ispravna vrijednost*) i vrijednosti članova dvodimenzionalnog realnog polja. Nakon toga u glavnom programu treba učitati dva cijela broja  $i$  i  $j$  koji predstavljaju indekse stupaca, te ako su indeksi  $i$  i  $j$  valjani, međusobno zamijeniti elemente stupaca  $i$  i  $j$ . Ukoliko bilo koji od unesenih indeksa nije valjan, učitavanje treba ponavljati sve dok se ne unesu ispravne vrijednosti. Indeksi redaka i stupaca počinju od 0. Naposljetku program treba ispisati matricu i ispisati sumu elemenata na glavnoj dijagonali.

**Zadatak 14.**

Napisati program koji učitava cijele brojeve. Učitavanje treba prekinuti kada se učita broj 0. Potrebno je ispisati broj učitanih peteroznamenkastih, četveroznamenkastih, troznamenkastih, dvoznamenkastih i jednoznamenkastih brojeva.

### Zadatak 15.

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 elemente matrice dimenzija  $n \times n$  te ispisati je li učitana matrica gornje trokutaste matrice, donje trokutaste matrice, jedinična matrica ili nijedno od navedenog.

### Zadatak 16.

Napisati program koji učitava niz od 20 znakova i znak  $c$ . Ulazni niz promijeniti na način da se ispred svakog pojavljivanja samoglasnika ubaci znak  $c$ . Nakon promjene, niz treba ispisati.

### Zadatak 17.

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 broja. Ako broj ima 1 znamenku, treba ispisati kvadrat te znamenke. Ako broj ima više od šest znamenki, potrebno je ispisati prve dvije znamenke učitano broja.

*Napomena: zadatak riješiti bez korištenja if naredbe.*

### Zadatak 18.

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 elemente matrice dimenzija  $n \times n$  te ispisati je li matrica centralno simetrična s obzirom na središnji element. Ako jest, program treba ispisati 1, ako nije, ispisati 0, a ako je broj redaka paran broj (pa nema središnjeg elementa), ispisati -1.

Primjer centralno simetrične matrice :

1	2	3	4	5
6	7	8	9	0
-1	-2	-3	-2	-1
0	9	8	7	6
5	4	3	2	1

### Zadatak 19.

Napisati program koji učitava s tipkovnice cijele brojeve koji predstavljaju temperature zraka izmjerene tijekom godine (broj mjerenja nije unaprijed poznat). Pretpostaviti da su temperature u intervalu  $[-100, 100]$ . Učitavanje završiti kada se unese broj izvan tog intervala.

Ukoliko nije unesena barem jedna valjana temperatura prekinuti program, inače ispisati prosječnu temperaturu te najčešću temperaturu (ukoliko ih ima više, ispisati višu temperaturu).

### Zadatak 20.

Napisati program u kojem se definira cjelobrojna matrica od najviše 10 redaka i 20 stupaca. Pri definiranju maksimalnih dimenzija matrice obavezno koristiti simboličke konstante.

S tipkovnice učitavati stvarne dimenzije matrice (broj redaka  $m$  i broj stupaca  $n$ ) dok god ne budu ispravno zadane. Učitati  $m \times n$  elemenata matrice po recima. Vrijednost svakog elementa matrice zamijeniti sumom elemenata koji se u originalnoj matrici nalaze u stupcu ispod tog elementa. Izmijenjenu matricu ne treba ispisati.

## Zadaci s funkcijama

### Zadatak 1.

Napisati **funkciju** `vratiSume` koja za zadani četvero-znamenkasti broj u pozivajući program vraća **dva cijela broja**: cijeli broj koji predstavlja sumu prve dvije znamenke i cijeli broj koji predstavlja sumu zadnje

dvije znamenke zadanog broja. Npr. za zadani broj 3725 funkcija vraća brojeve 10 i 7, a za zadani broj -3007 vraća brojeve 3 i 7.

### Zadatak 2.

Napisati **funkciju** `prebroji` koja za zadani nenegativni cijeli broj u pozivajući program vraća **dva cijela broja**: broj parnih znamenki i broj neparnih znamenki. Napisati glavni program koji učitava broj, poziva funkciju, te ispisuje spomenute brojeve.

Npr.

za učitani broj: 12345 treba ispisati:

```
U broju 12345 ima:  
parnih znamenki:2  
neparnih znamenki:3
```

### Zadatak 3.

Napisati funkciju koja za dani realan broj (float) vraća preko argumenata funkcije njegov cijeli i decimalni dio. Napisati i glavni program koji za učitani broj ispisuje njegov cijeli i decimalni dio.

Primjer

Za učitani broj 23.567

Program ispisuje cijeli dio 23 i decimalni 0.567.

### Zadatak 4.

Napisati prototip funkciju `provjeri` koja od korisnika učitava tri cijela broja i provjerava da li se znamenke drugog i trećeg broja nalaze u skupu znamenki prvog broja. Ukoliko se nalaze funkcija vraća 1, a inače 0.

Primjer: Za ulaz 123, 2, 3 funkcija vraća 1

za ulaz 12, 2, 3 funkcija vraća 0

### Zadatak 5.

Napisati **funkciju** `trokut` koja na temelju koordinata vrhova A, B i C u pozivajući program vraća izračunatu površinu i opseg trokuta. Površina trokuta računa se Heronovom formulom:

$P = \sqrt{s(s-a)(s-b)(s-c)}$ , gdje je  $s$  poluopseg trokuta, a  $a$ ,  $b$  i  $c$  duljine stranica trokuta. Također napisati

**funkciju** `cetverokut` koja na temelju koordinata vrhova A, B, C i D u pozivajući program vraća izračunatu površinu četverokuta. Za izračunavanje površine četverokuta potrebno koristiti funkciju `trokut` (podijeliti četverokut na dva trokuta). Napisati odsječak glavnog programa u kojem se poziva funkcija `cetverokut` te ispisuje površina.

### Zadatak 6.

Napisati **funkciju** `vratiSume` koja za zadani prirodni broj veći od 10 u pozivajući program vraća **dva cijela broja**: cijeli broj koji predstavlja sumu prve i zadnje znamenke i cijeli broj koji predstavlja sumu svih znamenki zadanog broja. Npr. za zadani broj 1528 funkcija vraća brojeve 9 i 16.

### Zadatak 7.

Napisati **glavni program** u kojem se unose dva cijela broja, **prvi** i **drugi**. Napisati **funkciju** `zamijeniZnamenke` koja prvu znamenku prvog broja stavlja na mjesto prve znamenke drugog broja, a zadnju znamenku prvog broja na mjesto zadnje znamenke drugog broja, i obratno (prvu znamenku drugog broja stavlja na mjesto prve znamenke prvog broja, a zadnju znamenku drugog broja na mjesto zadnje znamenke prvog broja). Ako je barem jedan od unesenih brojeva jednoznamenkast, funkcija vraća 1 i ne mijenja vrijednosti brojeva, inače funkcija vraća 0. U **glavnom programu ispisati** nove vrijednosti brojeva **prvi** i **drugi** i odgovarajuću poruku.

### Zadatak 8.

Napisati funkciju `kolinearne` koja preko parametara prima koordinate tri točke (koordinate su cjelobrojne). Funkcija treba ispitati da li su točke kolinerne (sve tri leže na istom pravcu). U slučaju da su kolinerne funkcija treba vratiti 1, a 0 inače.

#### **Zadatak 9.**

Napisati funkciju `parametri` koja prima cjelobrojne koordinate dviju točaka  $T1(x1, y1)$  i  $T2(x2, y2)$  te vraća parametre  $a$  i  $b$  pravca koji prolazi kroz te dvije točke. Parametar  $a$  je koeficijent smjera, a parametar  $b$  je odsječak na osi  $y$  u eksplicitnoj jednadžbi pravca  $y=ax+b$ . Pretpostavka koju ne treba provjeravati je da točke  $T1$  i  $T2$  nisu identične. Napisati glavni program koji učitava koordinate točaka, poziva funkciju i ispisuje koeficijente  $a$  i  $b$ .

#### **Zadatak 10.**

Napisati funkciju `kutRadUStupMinSek` koja zadani kut u radijanima pretvara u stupnjeve, minute i sekunde. Napisati glavni program u kojem se učitava kut u radijanima, poziva funkciju i ispisuje kut u stupnjevima, minutama i sekundama.

## Rješenja

### 1. zadatak

```
#include<stdio.h>

int main() {
    int niz[100], kniz[200];
    int i,n,uzastopni=0, br=0, br2=0;

    do{
        scanf("%d", &n);
        if (n != -1)
            niz[br++] = n;
    }
    while(n!=-1 && br<100);

    /* i će predstavljati poziciju broja za kojeg
    gledam koliko se puta pojavljuje */
    i = 0;
    while (i < br){
        n = niz[i];
        uzastopni=0;
        while(i + uzastopni < br && niz[i + uzastopni] == n)
            uzastopni++;

        kniz[br2++] = n;
        kniz[br2++] = uzastopni;
        i += uzastopni;
    }

    /*ispis*/
    for(i=0 ; i<br2 ; i++)
        printf("%d ", kniz[i]);

    return 0;
}
```

## 2. zadatak

```
#include<stdio.h>
#define MAXDIM 50

int main(){
    int polje[MAXDIM]={0}, polje_obrnuti[MAXDIM]={0}, br=0, elem, i;

    do{
        scanf("%d", &elem);
        if (elem != -1)
            polje[br++] = elem;
    } while(elem!=-1 && elem<50);

    for (i=0; i<br; i++){
        polje_obrnuti[i]=polje[br-1-i];
    }

    for (i=0; i<br; i++){
        printf("%d ", polje_obrnuti[i]);
    }

    return 0;
}
```

## 3. zadatak

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

int main()
{
    float r, x0, y0, x, y;
    int pogodjenihTocaka = 0;
    int brojTocaka = 0;
    printf("Unesite radijus, te koordinate središta kružnice: ");
    scanf("%f %f %f", &r, &x0, &y0);

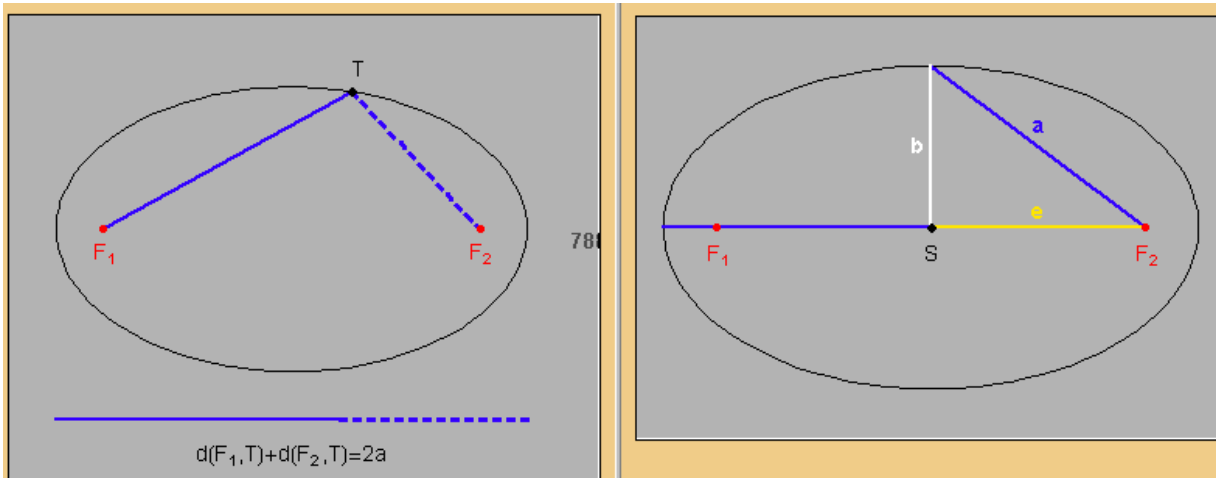
    do
    {
        printf("Unesite točku: ");
        scanf("%f %f", &x, &y);
        if (pow(x-x0, 2.) + pow(y-y0, 2.) == pow(r,2.)) pogodjenihTocaka++;
        brojTocaka++;
    } while((x!=0) || (y!=0));

    printf("Omjer broja točaka na kružnici i ukupnog broja točaka: %f\n",
    ((float)pogodjenihTocaka)/brojTocaka);

    return 0;
}
```

## 4. zadatak

**Definicija:** Elipsa je skup točaka u ravnini za koje je zbroj udaljenosti od dviju čvrstih točaka te ravnine (žarišta ili fokusa) konstantan.



```
#include<stdio.h>
#include<math.h>
int main()
{
    float xf1, yf1, xf2, yf2, dvaA, b, e, x, y;
    int pogodjenihTocaka = 0;
    int brojTocaka = 0;
    /*unos prvog i drugog fokusa te duljine male poluosi */
    printf("Unesite koordinate prvog fokusa elipse : ");
    scanf("%f %f", &xf1, &yf1);

    printf("\nUnesite koordinate drugog fokusa elipse : ");
    scanf("%f %f", &xf2, &yf2);

    printf("\nUnesite duljinu male poluosi elipse : ");
    scanf("%f", &b);
    /* Elipsa je skup točaka u ravnini za koje je zbroj udaljenosti od dviju čvrstih točaka te
    ravnine
    (žarišta ili fokusa) konstantan. Odnosno d(F1,T) + d(F2,T) =2a
    F1 je prvi fokus, F2 drugi fokus, a je velika poluos elipse
    a2=b2 + e2 gdje je b mala poluos a e udaljenost od središta elipse do F1 odnosno F2*/
    /*računanje udaljenosti između 2 fokusa tj. e pomoću kojeg će se izračunati duljina velike
    poluosi tj. a */
    e = sqrt(pow(xf2-xf1, 2.) + pow(yf2-yf1, 2.))/2;
    dvaA = sqrt(pow(b, 2.) + pow(e, 2.)) * 2;

    do
    {
        printf("Unesite točku: ");
        scanf("%f %f", &x, &y);
        if (sqrt(pow(x-xf1, 2.) + pow(y-yf1, 2.)) +
            sqrt(pow(x-xf2, 2.) + pow(y-yf2, 2.)) == dvaA ) pogodjenihTocaka++;
        brojTocaka++;
    } while((x!=0) || (y!=0));

    printf("Omjer broja tocaka na elipsi i ukupnog broja tocaka: %f\n",
    ((float)pogodjenihTocaka)/brojTocaka);
    return 0;
}
```

## 5. zadatak

```
#include<stdio.h>

int main()
{
    int x, y, p;
    printf("Unesite a i b\n");
    scanf("%d %d", &x, &y);

    if (x > y)
        p = x;
    else
        p = y;
    while (p%x || p%y)
        p++;
    printf("%d", p);

    return 0;
}
```

## 6. zadatak

```
#include <stdio.h>

int main(){
    int x, y, p;
    printf("Unesite a i b\n");
    scanf("%d %d", &x, &y);

    if (x > 0 && y > 0)
        while(y > 0){
            p = x;
            x = y;
            y = p % y;
        }
    printf ("%d", x);
    return 0;
}
```

## 7. zadatak

```
#include <stdio.h>

void UniijaZnamenkiUzlazno(int prviBroj, int drugiBroj){
    /*
    Polje koje će služiti za sumaciju znamenki oba broja
    */
    int brojacZnamenki[10] = {0};
    int i = 0;
    /*
    do-while petlja kako bi se uzelo u obzir i ako je
    ulazni broj nula - taj se slučaj gleda kao jedna
    znamenka 0
    */
    do{
        /*
        Uzima se znamenka s "kraja" broj uz pomoć modulo operacije
        te povećava vrijednost elementa polja koji je na poziciji
        vrijednosti uzete znamenke
        */
        brojacZnamenki[prviBroj%10]++;
        /*
        Zadnja se znamenka miče
        */
        prviBroj/=10;
    } while(prviBroj>0);
    do{
        brojacZnamenki[drugiBroj%10]++;
        drugiBroj/=10;
    } while(drugiBroj>0);
    /*
    Prolazi se kroz polje i ispisuju samo oni indeksi polja
    čije vrijednosti nisu nula (te su se znamenke javile)
    */
    for(i=0; i<10; i++){
        if(brojacZnamenki[i]) printf("%d ", i);
    }
}

int main(){
    int prviBroj, drugiBroj;
    printf("Ucitajte prvi broj: ");
    scanf("%d", &prviBroj);
    printf("Ucitajte drugi broj: ");
    scanf("%d", &drugiBroj);
    UniijaZnamenkiUzlazno(prviBroj, drugiBroj);
    return 0;
}
```

## 8. zadatak

```
#include <stdio.h>

int main (void)
{
    char kljuc[100]; int duljinak;
    char tekst[100]; int duljinat;
    char znak; int i;

    printf("Unesite kljuc: ");
    for (i=0; i<100; i++)
    {
        scanf("%c", &znak);
        if (znak=='0' || znak=='1'){
            kljuc[i]=znak;
        }else{
            break;
        }
    }
    duljinak=i;

    printf("Unesite tekst: ");
    for (i=0; i<100; i++)
    {
        scanf("%c", &znak);
        if (znak=='0' || znak=='1'){
            tekst[i]=znak;
        }else{
            break;
        }
    }
    duljinat=i;

    printf("\nKljuc: ");
    for (i=0; i<duljinak; i++)
    {
        printf("%c", kljuc[i]);
    }
    printf("\nTekst: ");
    for (i=0; i<duljinat; i++)
    {
        printf("%c", tekst[i]);
    }

    /*kodiranje*/
    for (i=0; i<duljinat; i++)
    {
        if (kljuc[i%duljinak]=='1')
        {
            if (tekst[i]=='0') tekst[i]='1';
            else tekst[i]='0';
        }
    }

    /* ispisi kodirani tekst */
    printf("\nKodirani\ntekst: ");
    for (i=0; i<duljinat; i++) printf("%c", tekst[i]);

    return 0;
}
```

## 9. zadatak

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

int main()
{
    int rang,i,j,tmp;
    int m[MAXRANG][MAXRANG];

    do
    {
        printf("Unesi rang matrice: ");
        scanf("%d",&rang);
    }while(rang>MAXRANG || rang<0);

    /*unos*/
    for(i=0;i<rang;i++)
    {
        for(j=0;j<rang;j++)
        {
            printf("m[%d][%d] = ",i,j);
            scanf("%d",&m[i][j]);
        }
    }

    /*transponiranje*/
    for(i=0;i<rang;i++)
    {
        for(j=0;j<i;j++)
        {
            tmp=m[i][j];
            m[i][j]=m[j][i];
            m[j][i]=tmp;
        }
    }

    /*ispis*/
    for(i=0;i<rang;i++)
    {
        for(j=0;j<rang;j++)
        {
            printf("%4d ",m[i][j]);
        }
        printf("\n");
    }
}
```

## 10. zadatak

```
#include <stdio.h>
#define MAXREC 15

int main()
{
    char pr[MAXREC][3];
    int n;
    int i,j;
    int imaP,imaS,imaO;

    do
    {
        printf("Unesi broj recenica: ");
        scanf("%d",&n);
    }
    while(n>MAXREC || n<0);

    for(i=0;i<n;i++)
    {
        printf("%d. ",i+1);
        scanf(" %c%c%c",&pr[i][0],&pr[i][1],&pr[i][2]);
    }

    printf("Neispravne recenice: \n");

    for(i=0;i<n;i++)
    {
        imaP=imaS=imaO=0;
        for(j=0;j<=2;j++)
        {
            if(pr[i][j]=='S') imaS=1;
            if(pr[i][j]=='P') imaP=1;
            if(pr[i][j]=='O') imaO=1;
        }

        if(!(imaP && (imaS || imaO)))
        {
            /*printf(" %c%c%c ",pr[i][0],pr[i][1],pr[i][2]);*/
            printf("%d.\n",i+1);
        }
    }
}
```

## 11. zadatak

```
#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);

    /* Moramo usporedjivati "svaki sa svakim".
       Prva petlja ide po svim znakovima prvog niza, a druga po svim
       znakovima drugog niza.
       Sjetimo se, kraj znakovnog niza je oznacen specijalnim
       znakom koji ima ASCII kod 0.
    */
    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;
    /*
       Pitanje za vjezbu:
       - Kako bi ispisali zajednicke znakove redom kako su se
       pojavljivali u 1.nizu, a ne abecedno?
       Npr.   EEAABB   EBA
       -> EAB
    */
}
```

## 12. zadatak

```
#include <stdio.h>
#define MAXRED 20
#define MAXSTUP 10

int main () {
    int mat[MAXRED][MAXSTUP];
    int m, n;

    int dobarRedak;
    int i, j;

    /* ucitati dimenzije matrice */
    do {
        printf("\nUpiši dimenzije matrice (m n): ");
        scanf("%d %d", &m, &n);
    } while (m<1 || m>MAXRED || n<1 || n>MAXSTUP);

    /* ucitati elemente matrice */
    for (i = 0; i < m; i++) {
        for (j = 0; j < n; j++) {
            printf("\nUpiši element matrice (%d, %d)", i+1, j+1);
            scanf("%d", &mat[i][j]);
        }
    }

    /* provjeriti za svaki redak da li ispunjava zadani uvjet */
    for (i = 0; i < m; i++) {
        dobarRedak = 1;
        /* za svaki redak: da li je j-ti clan za 1 veci od (j-1) - clana*/
        for (j = 1; j < n && dobarRedak == 1; j++) {
            if (mat[i][j] != mat[i][j - 1] + 1) {
                dobarRedak = 0;
                break;
            }
        }
        /* ako su svi clanovi retka ispunili uvjet, onda uvecaj brojac za 1
*/
        if (dobarRedak == 1) {
            printf("\n%d. redak je dobar!", i+1);
        }
    }

    return 0;
}
```

### 13. zadatak

```
#include <stdio.h>
#define MAX 20

int main(){
    int r;
    int i,j;
    int istup, jstup;
    float polje[MAX][MAX];
    float pom;

    /* Učitavanje ranga matrice */
    do {
        printf("\nUpišite rang matrice u intervalu [1,20]: ");
        scanf("%d", &r);
    } while(r<1 || r>MAX);

    /* Učitavanje elemenata matrice */
    for (i=0; i<r; i++) {
        for(j=0; j<r; j++) {
            printf("\nUpišite element polja (%d, %d)", i, j);
            scanf("%f", &polje[i][j]);
        }
    }

    /* ispis matrice */
    printf("\n\nPolje prije zamjene:\n");
    for (i=0; i<r; i++) {
        for(j=0; j<r; j++) {
            printf("%6.2f", polje[i][j]);
        }
        printf("\n");
    }

    /* Učitavanje indeksa stupaca koje treba zamijeniti */
    do{
        printf("\n\nUpišite indekse stupaca koje treba zamijeniti: ");
        scanf("%d %d", &istup, &jstup);
    }while(istup<0 || istup>=r || jstup<0 || jstup>=r);

    /* Zamjena elemenata stupaca */
    for(i=0; i<r; i++) {
        pom = polje[i][istup];
        polje[i][istup] = polje[i][jstup];
        polje[i][jstup] = pom;
    }

    /* ispis matrice nakon zamjene stupaca */
    printf("\n\nPolje poslije zamjene:\n");
    for (i=0; i<r; i++) {
        for(j=0; j<r; j++) {
            printf("%6.2f", polje[i][j]);
        }
        printf("\n");
    }

    return 0;
}
```

## 14. zadatak

```
#include <stdio.h>
#include <math.h>
int main(){
    int br[5] = {0};
    int broj, i;
    int brZnam;
    while (1) {
        scanf ("%d", &broj);
        if (broj == 0)
            break;

        if (broj < 0)
            broj = -broj;
        if (broj < 100000) {
            brZnam=0;
            while(broj){
                broj/=10;
                ++brZnam;
            }
            br[ brZnam - 1 ]++;
            /* ili: br[ (int) log10(broj) ]++; */
        }
    }

    for (i=4; i>=0; --i){
        printf ("Broj %d-znamenastih je: %d\n", i+1, br[i]);
    }
    return 0;
}
```

## 15. zadatak

```
#include <stdio.h>

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

    printf("\nUnesite matricu %dx%d\n", n, n);

    gt = dt = jed = 1;

    for (i=0; i<n; ++i){
        for (j=0; j<n; ++j){
            scanf("%d", &m[i][j]);
            /* Ima jedinice na dijagonali a ostalo nule:*/
            if (i == j && m[i][j] != 1) {
                jed = 0;
            } else if (i!=j && m[i][j] != 0) {
                jed = 0;
            }
            /* GT matrica mora imati sve elemente ispod gl. dij. jednake 0*/
            if (i>j && m[i][j] != 0) {
                gt = 0;
            }
            /* DT matrica mora imati sve elemente iznad gl. dij. jednake 0*/
            if (i<j && m[i][j] != 0) {
                dt = 0;
            }
        }
        printf("\n");
    }
    /* Nisu if-else if zato jer je moguće da matrica ima istodobno
    više navedenih svojstava
    Stovise, ako je matrica jedinica onda je i gt i dt. */
    if (jed){
        printf("\nMatrica je jedinica.");
    }
    if (gt){
        printf("\nMatrica je gornja trokutasta.");
    }
    if (dt){
        printf("\nMatrica je donja trokutasta.");
    }
    if (jed+gt+dt == 0){
        printf("\nMatrica nije niti jedinica niti GT niti DT.");
    }
}
```

## 16. zadatak

```
#include <stdio.h>

int main()
{
    /*treba uzeti dovoljno veliki niz (ekstremni slučaj da su u nizu samo samoglasnici*/
    char c,niz[40];

    int i,j,duljina=20;
    printf("Upisite niz od 20 znakova, a zatim (bez razmaka) unesite i znak koji
treba ubaciti ispred svakog samoglasnika\n");

    for(i=0; i<duljina ; i++)
        scanf("%c", &niz[i]);

    scanf("%c", &c);

    for(i=0; i<duljina ; i++){
        switch(niz[i]){
            case 'a':case 'e': case 'i': case 'o': case 'u':
                /* pomak svih znakova udesno */
                for(j=duljina ; j>i ; j--)
                    niz[j] = niz[j-1];
                niz[i] = c;

                duljina++;
                i++;/* da ne bi došli u beskonačno pomicanje, jer je sad
samoglasnik pomaknut jedno mjestu udesno */
                break;
        }
    }

    for(i=0; i<duljina ; i++)
        printf("%c", niz[i]);

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

## 17. zadatak

```
#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 učitanoj broju*/
    do{
        znamenke[i]=broj%10;
        broj=broj/10;
        i++;
        brZnam++;
    }while (broj!=0);

    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;
        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;
}
```

## 18. zadatak

```
#include <stdio.h>
#define maxstup 10
int main()
{
    int mat[maxstup][maxstup];
    int i,j,n,rez;
    do{
        printf("Unesite rang matrice: ");
        scanf("%d", &n);
    }
    while(n<1 || n>10);
    for(i=0; i<n ; i++){
        for(j=0 ; j<n ; j++){
            printf("mat[%d][%d] = ", i, j);
            scanf("%d", &mat[i][j]);
        }
    }

    rez = 1; /* pretpostavimo da je matrica centralno simetricna */
    if (n % 2 == 0)
        rez = -1;
    else{
        for(i = 0; i < n ; i++){
            for(j = 0 ; j <= i ; j++){
                if (mat[i][j] != mat[n-1-i][n-1-j]){
                    rez = 0;
                    break;
                }
            }
        }
    }
    printf("%d\n", rez);
    return 0;
}
```

## 19. zadatak

```
#include <stdio.h>

int main(){
    int br[100-(-100)+1]={0};
    int maxi, sum, temp, i, brMjerenja = 0;

    while(1){
        scanf("%d", &temp);
        if(temp>=-100 && temp<=100){
            br[temp-(-100)]++;
            brMjerenja++;
        }
        else
            break;
    }
    if (brMjerenja == 0)
        return 0;
    maxi = sum = 0;

    for(i=0;i<201;i++){
        if (br[i] >= br[maxi])
            maxi = i;
        sum += br[i] * (i-100);
    }
    printf("%f %d", (float)sum/brMjerenja, maxi-100);
    return 0;
}
```

## 20. zadatak

```
#include <stdio.h>
#define MAXRED 10
#define MAXSTUP 20
int main () {
    int i, j, k, suma, m, n;
    int mat[MAXRED][MAXSTUP];
    /* učitavanje velicine matrice */
    do {
        printf("Upisi broj redaka i broj stupaca:");
        scanf("%d %d", &m, &n);
    } while (m < 1 || m > MAXRED || n < 1 || n > MAXSTUP);
    /* učitavanje elemenata matrice */
    for (i = 0; i < m; i++)
        for (j = 0; j < n; j++)
            scanf("%d", &mat[i][j]);
    /* izmjena vrijednosti elemenata matrice */
    for (i = 0; i < m; i++)
        for (j = 0; j < n; j++) {
            suma = 0;
            for (k = i + 1; k < m; k++)
                suma += mat[k][j];
            mat[i][j] = suma;
        }

    for (i = 0; i < m; i++){
        for (j = 0; j < n; j++)
            printf("%3d", mat[i][j]);
        printf("\n");
    }
    return 0;
}
```

## RJEŠENJA ZADATAKA S FUNKCIJAMA

### Zadatak 1.

```
#include <stdio.h>
void vratiSume(int broj, int *prviDio, int *drugiDio) {
    if (broj < 0) broj = -broj;
    *prviDio = (broj/1000) + (broj - (broj/1000)*1000)/100;
    *drugiDio = (broj%10) + (broj%100 - broj%10)/10;
    return;
}
int main () {
    int n, prvi, drugi;
    printf("Upisite 4-znamenkasti broj: ");
    scanf("%d", &n);
    if (n >= 1000 && n <= 9999 || n >= -9999 && n <= -1000) {
        vratiSume(n, &prvi, &drugi);
        printf("Trazeni rastav broja %d jest %d i %d\n", n, prvi, drugi);
    }
    else
        printf("Broj nije ispravan\n");
    return 0;
}
```

## Zadatak 2.

```
void prebroji(unsigned int br, int *brP, int *brN){
    *brP = *brN = 0;
    while(br) {
        if (br%2) { // isto kao i br%10%2
            (*brN)++;
        } else {
            (*brP)++;
        }
        br /= 10;
    }
}

int main(){
    unsigned int br, brP, brN;
    scanf("%d", &br);
    prebroji(br, &brP, &brN);
    printf("U nizu %d ima:\nparnih znamenki:%d\nneparnih znamenki%d", br,
brP, brN);
    return 0;
}
```

## Zadatak 3.

```
void funkcija(float broj, int *cijeli, float *decimalni)
{
    *cijeli = (int) broj;
    *decimalni = broj - *cijeli;
}

int main()
{
    int c;
    float b, d;

    printf("Unesi broj:");
    scanf("%f", &b);
    funkcija(b, &c, &d);
    printf("Cijeli dio: %d, decimalni dio: %f", c, d);

    return 0;
}
```

#### Zadatak 4.

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

#define MAX_DULJINA 100

//za svaku znamenku unutarnjih broja (to je drugi ili treći broj)
//provjerava se je li sadržana u skupu znamenki prvog broja
int provjeri_znamenke(int broj_vanjski, int broj_unutarnji){
    int znamenka_analiza, znamenka_vanjskog_broja;
    int broj_vanjski_kopija;
    int pronadjena_znamenka;
    //prolaz kroz sve znamenke unutarnjeg broja
    while(broj_unutarnji > 0){
        znamenka_analiza = broj_unutarnji % 10;
        pronadjena_znamenka = 0;
        broj_vanjski_kopija = broj_vanjski;
        //prolaz kroz sve znamenke vanjskog broja
        while(broj_vanjski_kopija > 0){
            znamenka_vanjskog_broja = broj_vanjski_kopija % 10;
            if(znamenka_vanjskog_broja == znamenka_analiza)
            {
                pronadjena_znamenka = 1;
                break;
            }
            broj_vanjski_kopija /= 10;
        }
        broj_unutarnji /= 10;
        //ukoliko neka znamenka unutarnjeg broja nije pronađena
        //odmah se vraća 0 jer dalje nema smisla provjeravati
        //u suprotnom se moraju provjeriti i ostale znamenke unutarnjeg
broja
        if(!pronadjena_znamenka){
            return 0;
        }
    }
    //tek ako postoje sve znamenke unutarnjeg broja (u vanjskom broju)
    //smije se vratiti 1
    return 1;
}

//Učitavanje brojeva i poziv funkcije za provjeru znamenki
int provjeri(){
    int prvi_broj, drugi_broj, treci_broj;
    scanf("%d", &prvi_broj);
    scanf("%d", &drugi_broj);
    scanf("%d", &treci_broj);
    if(provjeri_znamenke(prvi_broj, drugi_broj) &&
provjeri_znamenke(prvi_broj, treci_broj)){
        printf("Provjera uspjela");
        return 1;
    }
    else{
        printf("Provjera NIJE uspjela");
        return 0;
    }
}

int main()
{
```

```
    provjeri();
    return 0;
}
```

### Zadatak 5.

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

void trokut(int x1, int y1, int x2, int y2, int x3, int y3, float *opseg,
float *povrsina){

    float a,b,c,s;

    a=sqrt((float) (x1-x2)*(x1-x2)+(y1-y2)*(y1-y2));
    b=sqrt((float) (x1-x3)*(x1-x3)+(y1-y3)*(y1-y3));
    c=sqrt((float) (x2-x3)*(x2-x3)+(y2-y3)*(y2-y3));

    *opseg=a+b+c;
    s=(a+b+c)/2;
    *povrsina=sqrt(s*(s-a)*(s-b)*(s-c));
}

float cetverokut(int x1,int y1,int x2, int y2,int x3,int y3,int x4,int y4){

    float opseg1,opseg2,povrsina1,povrsina2;

    trokut(x1,y1,x2,y2,x3,y3,&opseg1,&povrsina1);
    trokut(x1,y1,x3,y3,x4,y4,&opseg2,&povrsina2);

    return povrsina1+povrsina2;
}

int main(){

    int x1,y1,x2,y2,x3,y3,x4,y4;
    float povrsina;
    ...
    povrsina=cetverokut(x1,y1,x2,y2,x3,y3,x4,y4);

    printf("Povrsina=%f",povrsina);

    return 0;
}
```

### Zadatak 6.

```
#include <stdio.h>

void vratiSume(int broj, int *prvasuma, int *drugasuma) {
    int i;

    *prvasuma=broj%10;
    *drugasuma=0;

    while (broj>0){
        i=broj%10;
        *drugasuma+=i;
        broj/=10;
    }
}
```

```

        *prvasuma+=i;

        return;
    }

int main () {
    int n, prvasuma, drugasuma;
    printf("Upisite prirodni broj: ");
    scanf("%d", &n);
    if (n > 10) {
        vratiSume(n, &prvasuma, &drugasuma);
        printf("Trazene sume broja %d su %d i %d\n", n, prvasuma,
drugasuma);
    }
    else
        printf("Broj nije ispravan\n");
    return 0;
}

```

### Zadatak 7.

```

#include <stdio.h>
int zamijeniZnamenke(int *prvi, int *drugi) {
    int pom1, pom2, br1, br2, z1 = 1, z2 = 1;
    if(((*prvi) < 0 || (*prvi) > 9) && ((*drugi) < 0 || (*drugi) > 9))
    {
        //zamjena prvih znamenki
        br1 = *prvi; br2 = *drugi;
        while(br1 != 0 || br2 != 0)
        {
            if (br1 != 0)
            {
                pom1 = br1 % 10;
                br1 /= 10;
                if (br1 != 0) z1 *= 10;
            }
            if (br2 != 0)
            {
                pom2 = br2 % 10;
                br2 /= 10;
                if (br2 != 0) z2 *= 10;
            }
        }
        *prvi = *prvi - pom1 * z1 + pom2 * z1;
        *drugi = *drugi - pom2 * z2 + pom1 * z2;

        //zamjena zadnjih znamenki
        pom1 = (*prvi)%10;
        pom2 = (*drugi)%10;
        *prvi = (*prvi) - pom1 + pom2;
        *drugi = (*drugi) - pom2 + pom1;

        return 0;
    }
    else
    {
        return 1;
    }
}

int main () {
    int prvi, drugi, rez;

```

```

scanf("%d %d", &prvi, &drugi);
rez = zamijeniZnamenke(&prvi, &drugi);
if (rez == 0) printf("Zamijenjeni brojevi su: %d %d", prvi, drugi);
else printf("Unesen je jednoznamenkasti broj.");
}

```

### Zadatak 8

```

int kolinearne(int x1, int y1, int x2, int y2, int x3, int y3)
{
    double x2mx1, a, b;
    if(x1 == x2)
    {
        if(x1==x3)
            return 1;
        else
            return 0;
    }
    else
    {
        a=((double)y2-y1)/((double)(x2-x1));
        b=-x1*a+y1;
        if(y3==(a*x3+b))
            return 1;
        else
            return 0;
    }
}

```

### Zadatak 9

```

void parametri(int x1, int y1, int x2, int y2, float * a, float * b){
    *a=(float)(y1-y2)/(x1-x2);
    *b=y1-(*a)*x1;
}

int main(){
    int x1, x2, y1, y2;
    float a,b;
    printf("Učitajte kordinate točaka T1 i T2: ");
    scanf("%d %d %d %d", &x1, &y1, &x2, &y2);
    parametri(x1, y1, x2, y2, &a, &b);
    printf("\nParametar a=%f, b=%f", a , b);
}

```

### Zadatak 10

```

#include <stdio.h>
#define pi 3.141592f

void kutRadUStupMinSek(float kutRadF, int* stupF, int* minF, int* sekF)
{
    float tmp = kutRadF/pi * 180;
    *stupF = tmp;
    tmp = (tmp - (int)tmp)*60;
    *minF = tmp;
    *sekF = (int)((tmp - *minF) * 60);
}

int main()
{
    float kutRad;
}

```

```
int stup, min, sek;
printf("Kut u radijanima: ");
scanf("%f", &kutRad);

kutRadUStupMinSek(kutRad, &stup, &min, &sek);
printf("Kut u radijanima %f = %d stupnjeva, %d minuta i %d sekundi",
      kutRad, stup, min, sek);
return 0;
}
```