

* * * C * C * * O * M * G * C * C * G * M * O * * C * C * * *
* * C * C * * R G A M N * * + * * N M A G R * * C * C * *
* C * * P * R * I * * * + * * * I * R * P * * C *

Autor: Murtic Nirmel
www.etf.ba

PROGRAMIRANJE U PROGRAMSKIM JEZICIMA „C“ & „C++“

ZBIRKA ZADATAKA

Verzija: **FiVe dOts**

11.04.2006

AUTOR: Murtic Nirmel (FiVe dOts)
Email: matrix.bih@gmail.com

Email: matrix.bih@gmail.com ; MSN: nirmel_m@hotmail.com

* * * C * C * * O * M * G * C * C * G * M * O * * C * C * * *
* * C C * * R G A M N * * + * * N M A G R * * C C * *
* C * * P * R * I * * * + * * * I * R * P * * C *

Autor: Murtic Nirmel
www.etf.ba

... Sadržaj ...

Napomena	3
Uvod	4
Zadaci iz C-a	5
Zadaci iz Cpp-a	62

***C*C*O*M*G*C*C*G*M*O**C**
*C*C*RGAMN**++**NMAGR**C**
*C**P*R*I**+**I*R*P**C*

Autor: Murtic Nirmel
www.etf.ba

... Napomena ...

Cilj skripte nije omogućavanje studentima da iste zadatke prepisu, već da na osnovu ovih rješenja napišu neka svoja.

Skriptu možete preuzeti sa web stranice:

<http://www.etf.ba>

Skripta će biti nadopunjavana, neće stati na ovome! A obnovljenu verziju skripte ćete moći preuzeti sa stranice već navedene.

Rješenja su provjerena i rade u **Dev Cpp-u**, ukoliko ipak neki od programa ne radi, prvo **provjerite dobro** da niste neki dio zaboravili kopirati. Neki znakovi nakon što se prebace u pdf nisu isti, ako ipak postoji greška, javite mi na mail ili msn, da ispravim.

Sretno sa programiranjem...

... do rješenja se može doći na 1000... načina... **vama treba samo jedno rješenje**, bilo koje... **a želi se najbolje, najbrže i najkrace...** jedno je **želja...** drugo su **mogućnosti...**

... ako vam baš i neide najbolje... zapamtite... **nijedan program nije vrijedan vaših živaca...**

Za štetu nastalu korištenjem programa, autor skripte ne odgovara :)

* * * C * C * * O * M * G * C * C * G * M * O * * C * C * * *
* * C * C * * R G A M N * * + * * N M A G R * * C * C * *
* C * * P * R * I * * * + * * * I * R * P * * C *

Autor: Murtic Nirmel
www.etf.ba

... Uvod ...

Cilj ove skripte je da prikaže primjenu programskog jezika C & C++ na zadacima. Podrazumjeva se da ste prije „proucavanja“ ove skripte savladali sintaksu C-a. Dva glavna „problema“ koji se javljaju kroz programiranje su:

- **sintaksa programskog jezika;**
- **logika programa (daleko veci problem).**

Savladati sintaksu i nije neki problem, mnogo veci problem je nauciti razmisljati programerski. Jedno ide uz drugo, nemoze se programirati ako neznamo sintaksu, a ne vrijedi nam ni sintaksa ako nemozemo „skontati“ kako da dodjemo do rjesenja.

Medjutim, iz dana u dan „izlaze“ novi programski jezici, nove sintakse, i sintaksa nekog starijeg programskog jezika nam nemoze mnogo pomoci. Znacni mnogo vise se treba osvrnuti da se savlada ovaj drugi problem, jer ako znamo rijesiti program u jednom programskom jeziku, to mozemo uraditi u svakom (uz kratko proucavanje sintakse).

Stoga ovdje mozete naci samo rjesene i objasnjene zadatke, kao i neke metode koje su prakticne za primjenjivanje u programima, a za sintaksu, pronadjite neku drugu litereturu ;).

* * * C * C * * O * M * G * C * C * G * M * O * * C * C * * *
* * C C * * R G A M N * * + * * N M A G R * * C C * *
* C * * P * R * I * * * + * * * I * R * P * * C *

Autor: Murtic Nirmel
www.etf.ba

... Zadaci iz C-a ...

```
***C*C*O*M*G*C*C*G*M*O*C*C**
**C*C**R*G*A*M*N**++**N*M*A*G*R**C*C**
*C**P*R*I**++**I*R*P**C*
```

Autor: Murtic Nirmel
www.etf.ba

1. Napraviti program koji učitava koordinate dvije tačke u trodimenzionalnom prostoru:

Tačke t_1 i t_2 su definirane kao trojke:

$$\begin{aligned} - t_1 &= (x_1, y_1, z_1) , \\ - t_2 &= (x_2, y_2, z_2) . \end{aligned}$$

Kao izlaz program daje udaljenost između tih tački.

NAPOMENA: Za drugi korijen koristiti funkciju **sqrt()** koja je definirana u datoteci math.h. Na primjer, naredba **a=sqrt(9)**; će izračunati drugi korijen iz 9 i pridružiti izračunatu vrijednost varijabli a.

RJEŠENJE:

```
// Uključujemo datoteke u kojima se nalaze funkcije koje ćemo koristiti.
# include <stdio.h>
# include <math.h>
// Definisavanje glavne funkcije main()
int main () {
// Definisavanje potrebnih varijabli
float x1,y1,z1,x2,y2,z2,d;
// Unos koordinata x1,y1,z1 (printf - ispis poruke; scanf - unos varijabli)
printf("Unesi tačku a(x1,y1,z1): ");
scanf("%f,%f,%f", &x1, &y1, &z1);
// Unos koordinata x2,y2,z2
printf("Unesi tačku b(x2,y2,z2): ");
scanf("%f,%f,%f", &x2, &y2, &z2);
// Racunanje udaljenosti po formuli (sqrt - funkcija koja racuna drugi korjen)
d=sqrt((x2-x1)*(x2-x1)+(y2-y1)*(y2-y1)+(z2-z1)*(z2-z1));
// Ispis udaljenosti d
printf("Udaljenost između tački a i b je: %f", d);
// Kraj programa (getch() - stavljamo da bi zadržali poruku u kojoj je
// rezultat. U suprotnom nam se neće prikazati poruka na ekranu, iako je
// program uspješno izvršen
getch();
// Return 0; - (umjesto 0 možemo staviti bilo koji cio broj, jednostavno
// nekako je potrebno završiti funkciju main() pa se obično stavlja return 0;
return 0;
}
```

```
*** C * C * * O * M * G * C * C * G * M * O * * C * C * * *
* * C * C * * R * G * A * M * N * * * + * * * N * M * A * G * R * * * C * C * *
* * C * * P * R * I * * * + * * * I * R * P * * * C * *
```

Autor: Murtic Nirmel
www.etf.ba

2. Napravite C-program koji učitava potrošnju električne energije u KWh (velika tarifa(VT) i mala tarifa(MT)) i pripadajuće cijene po KWh (cijena za malu tarifu i cijena za veliku tarifu).

Kao izlaz program daje:

- ukupnu potrošnju u KWh;
- novčani iznos racuna;
- udio velike tarife i male tarife u ukupnoj potrošnji.

Na primjer, izlaz iz programa treba biti sljedećeg formata:

Ukupna potrošnja je 234.25 KWh.

Udio velike tarife u ukupnoj potrošnji je 25 %.

Udio male tarife u ukupnoj potrošnji je 75%.

Iznos racuna je 25.45 KM.

RJEŠENJE:

```
# include <stdio.h>
int main () {
// Definisavanje potrebnih varijabli
float VT,MT,CV,CM,UKUPNO,RACUN,UDIOM,UDIOV;
// Unos male tarife, velike tarife, cijene male tarife i cijene velike tarife
printf("\n Unesi potrošnju po velikoj tarifi (KWh): ");
scanf("%f", &VT);
printf("\n Unesi potrošnju po maloj tarifi (KWh): ");
scanf("%f", &MT);
printf("\n Cijena za veliku tarifu po jednom Kwh: ");
scanf("%f", &CV);
printf("\n Cijena za malu tarifu po jednom Kwh: ");
scanf("%f", &CM);
// Racunanje ukupne potrošnje
UKUPNO=VT+MT;
// Racunanje cijene ukupne potrošnje (cijena po maloj tarifi + cijena po
// velikoj tarifi)
RACUN=VT*CV+MT*CM;
// Izracunava udio velike i male tarife u ukupnoj potrošnji (procenat)
UDIOM=(MT/UKUPNO)*100;
UDIOV=(VT/UKUPNO)*100;
// Ispis rezultata
printf ("\n Ukupna potrošnja električne energije je: %.f KWh", UKUPNO);
printf ("\n Ukupna cijena električne energije je: %.f KM", RACUN);
printf ("\n Udio velike tarife u potrošnji je: %.f procenata", UDIOV);
printf ("\n Udio male tarife u potrošnji je: %.f procenata", UDIOM);
// Kraj programa
getch();
return 0;
}
```

```
* * * C * C * * O * M * G * C * C * G * M * O * * C * C * * *
* * C * C * * R G A M N * * * + * * N M A G R * * C C * *
* C * * P * R * I * * * + * * * I * R * P * * C *
```

Autor: Murtic Nirmel
www.etf.ba

3. Napisati program koji ce izracunati presjek skupova S1 i S2. Skupovi S1 i S2 definirani su intervalima realnih brojeva:

S1=[A, B]
S2=[C, D].

Na primjer, za slijedeci ulaz: **A=3, B=9, C=7, D=11**
program kao izlaz treba dati:

Rezultantni interval je **REZ=[7, 9].**

RJEŠENJE:

```
#include <stdio.h>
main() {
// Definisanje varijabli
float A,B,C,D,MAX,MIN;
// Unos skupova
printf("\n Unesi prvi skup X[A,B] A>B: ");
scanf("%f,%f", &A,&B);
printf("\n Unesi drugi skup Y[C,D] C>D: ");
scanf("%f,%f", &C,&D);

/* Logika zadatka je sljedeca: Prvo moramo pretpostaviti da je prvi broj
intervala veci ili jednak drugom, A>=B i C>=D, jer ako ovo ne vrijedi interval
nije matematicki ispravan (taj uslov nismo ispitivali da nebi komplikovali
zadatak). Presjek skupa ce biti neki interval [MIN,MAX].
Uslov da zadatak ima RJEŠENJE je da je B < C kao i D < A sto je i logicko jer
u suprotnom presjeka nema!
Broj MIN je ustvari jedan od brojeva A ili C (manji broj od ova 2 broja),
a MAX je manji broj od brojeva B i D. Ako je A=C ili B=D uzima se vrijednost
pod ELSE ali tada nam je svakako svejedno koja ce se uzeti vrijednost kao MIN
ili MAX jer je tada A=C odnosno B=D */

if (A>C) MAX=A; else MAX=C;
if (B<D) MIN=B; else MIN=D;
if ((B<C)|| (A>D)) printf("\n Nema presjeka!"); else {
printf ("\n Presjek je: [%.f,%.f]", MAX, MIN); }
// Kraj programa
getch();
return 0;
}
```

```
***C**O**M**G**C**G**M**O**C**C**
**C**R**G**A**M**N**++**N**M**A**G**R**C**C**
**C**P**R**I**++**I**R**P**C**
```

Autor: Murtic Nirmel
www.etf.ba

4. Napisati program koji učitava koeficijente **a**, **b** i **c** realnog tipa. Ti koeficijenti definiraju funkciju **f(x)**:

$$f(x) = ax^2 + bx + c$$

Program zatim provjerava da li su ti koeficijenti u rasponu -10 do 10. Ako nisu, program ispisuje poruku:

Koeficijenti **a**, **b** i **c** nisu u zadanom rasponu.

Ako koeficijenti **a**, **b** i **c** jesu u zadanom rasponu, program kao izlaz daje vrijednost prve derivacije u tacki **x**. Tacka **x** se takodjer unosi sa standardnog ulaza.

Na primjer, ako je ulaz **a=1**, **b=2**, **c=1**, **x=1** program kao izlaz daje: **Prva derivacija u tacki x=1 je 4.**

RJEŠENJE:

```
# include <stdio.h>
int main () {
// Definisanje varijabli
float a,b,c,x,izvod;
// Unos koeficijenata a,b,c
printf ("Unesi koef. (a,b,c) funkcije f(x)=ax2+bx+c (-10<(a,b,c)<10):");
scanf ("%f,%f,%f", &a,&b,&c);
// Unos vrijednosti x
printf ("Unesi vrijednost x");
scanf ("%f", &x);
// Ispit uslova
if (a>-10&&a<10&&b>-10&&b<10&&c>-10&&c<10){
// Izracunavanje izvoda po odredjenoj matematickoj formuli
izvod=2*a*x+b;
// Ispis izvoda
printf("Prva derivacija u tacki x=%.f je %.f", x, izvod);
// Ispis odgovarajuce poruke ukoliko uslov nije zadovoljen
} else printf ("Koeficijenti nisu iz odgovarajuceg intervala!");
// Kraj programa
getch();
return 0;
}
```

```
***C*C*O*M*G*C*C*G*M*O**C**  
*C**R*G*A*M*N**++**N*M*A*G*R**C**  
*C**P*R*I**+**I*R*P**C*
```

Autor: Murtic Nirmel
www.etf.ba

5. Napisati program koji ucitava tri cijela broja iz intervala od 50 do 200. Ako svi uneseni brojevi nisu iz tog intervala, program treba ispisati poruku **"Svi brojevi nisu iz intervala 50 do 200"** i završiti program. Ako su brojevi unutar tog intervala program treba ispisati ucitane brojeve po velicini od najmanjeg prema najvećem.

RJEŠENJE:

```
#include <stdio.h>  
main() {  
    // Definisavanje varijabli  
    int x[3], i, j, p;  
    // Unos brojeva  
    printf("\n Unesi tri broja u rasponu od 50 do 200 (U formatu: a, b, c): ");  
    scanf("%d, %d, %d", &x[0], &x[1], &x[2]);  
    // Provjera jesu li brojevi iz odgovarajućeg intervala  
    if (x[0]>50&&x[1]>50&&x[2]>50&&x[0]<200&&x[1]<200&&x[2]<200) {  
        /* Sortiranje brojeva  
        Potrebne su nam dvije for petlje koje ce „listati“ brojeve koji se nalaze u  
        nekom nizu. Ovaj tip sortiranja se uglavnom svodi na provjeru da li je prvi  
        broj manji od drugog ili trećeg, i ako jeste, onda mijenja mjesto sa tim  
        brojem. Medjutim nakon ovoga nije završeno sortiranje. Ako npr. Unesemo  
        brojeve 3, 2, 1, nakon prvog prolaska imamo 2, 3, 1 nakon drugog 2, 1, 3 ali  
        jos uvijek nije završeno sortiranje, i zbog toga i postoji prva for petlja  
        koja ce to ponoviti jos jednom i dobivamo 1, 2, 3 i niz je sortiran, i sada ga  
        samo jos treba ispisati na ekran. */  
        for (i=0;i<2;i++)  
            for (j=0;j<2;j++)  
                if (x[j]>x[i]) {  
                    p=x[i];  
                    x[i]=x[j];  
                    x[j]=p;  
                }  
        // Ispis sortiranih brojeva  
        printf("\n Sortirani brojevi: %d, %d, %d ", x[0], x[1], x[2]);  
        // Ako brojevi nisu iz intervala, ispis odgovarajuće poruke  
        } else { printf("\n Svi brojevi nisu iz intervala 50 do 200!!!");  
    }  
    // Kraj programa  
    getch();  
    return 0;  
}
```

```
***C**C**O**M**G**C**C**G**M**O**C**C**  
**C**C**R**G**A**M**N**++**N**M**A**G**R**C**C**  
*C**P**R**I**++**I**R**P**C*
```

Autor: Murtic Nirmel
www.etf.ba

6. Napisati program koji ce ispisati brojeve djeljive sa 5, 7 ili 11, a manje ili jednake od nekog ucitanog broja n sa tastature. Brojeve treba ispisati od najveceg prema najmanjem. Obavezno koristiti **for** petlju. Na primjer, ako je korisnik za n unio broj 20 onda program treba ispisati:

```
20  
15  
14  
11  
10  
7  
5
```

RJEŠENJE:

```
#include <stdio.h>  
int main()  
{  
    // Definisane varijabli  
    int n, i;  
    // Unos nekog prirodnog broja  
    printf("Unesite broj: ");  
    scanf("%d", &n);  
    /* For petlja postavlja brojac „i“ na vrijednost unesenog broja, i provjerava se da li je taj broj djeljiv sa 5, 7 ili 11, i ako jeste printa ga na ekran i umanjuje brojac za jedan, ako broj nije djeljiv sa ovim brojevima, onda se samo brojac umanja na kraju for petlje bez printanja broja na ekranu. Isti se postupak ponavlja i za broj manji za jedan, kao i broj manji za dva... sve tako do broja 1 (mogli smo staviti i 4 jer brojevi ispod 5 svakako nisu djeljivi sa ovim brojevima! */  
    for(i=n;i>0;i--)  
    {  
        if ((i % 5 == 0)|| (i % 7 == 0)|| (i % 11 == 0)) printf("\n Broj: %d", i);  
    }  
    // Kraj programa  
    getch();  
    return 0;  
}
```

```
***C*C*O*M*G*C*C*G*M*O*C*C***
**C**R*G*A*M*N**++**N*M*A*G*R**C*C**
*C**P*R*I**++**I*R*P**C*
```

Autor: Murtic Nirmel
www.etf.ba

7. Napisati program koji ce izracunavati sumu prema slijedećoj formuli: $suma=1-1/2+1/3-1/4+\dots+1/n$

Broj **n** se unosi sa tastature(standardnog ulaza). Program treba ispisati vrijednost sume na standardnom izlazu (monitoru) koristeći slijedeću poruku:

Koristeći **<n>** clanova suma je **<suma>**.

(Na mjestu **<n>** se treba nalaziti konkretno uneseni broj n, a na mjestu **<suma>** se treba nalaziti konkretno izracunata suma.)

Zadatak riješite koristeći **while** petlju.

RJEŠENJE:

```
#include <stdio.h>
int main()
{
// Definisavanje varijabli
float i=1,n,suma=0;
// Unos prirodnog broja
printf ("\n Unesi broj n:");
scanf("%f", &n);
/* Racunanje sume
While petlja vrti podprogram u okviru while petlje sve dok je brojac „i“ koji
je na pocetku 1, manji od unesenog broja n. Ako je uneseni broj 1 ne ulazi se
u petlju i na izlazu se ispisuje Suma je:1. Ako je broj veci od 1 onda se
racuna suma prema matematickoj formuli. Grupisali smo formulu datu u zadatku
tako da trazimo sume po dva broja (odnosno razlike), i zatim te razlike
brojeva sabiramo, ali moramo paziti jer moramo brojac „i“ uvecati za 2 jer smo
u formuli ukljucili odmah 2 broja.
Ovaj nacin ima i dobru i losu stranu. Dobra strana je jer nemoramo uvoditi
nikakvu pomocnu varijablu i citav kod za racunanje sume se sastoji od dva
izraza, a losa strana je kad unesemo neparne brojeve sto necemo dobiti tacnu
sumu i tako kad unesemo broj 3 program ce izracunati 1-1/2+1/3=0.5 sto nije
tacno, a to smo dobili jer program nemoze grupisati clanove po 2 jer ih imamo
tri i zadnji ostaje sam i ne racuna se. Ali ovaj mali nedostatak mozemo
zanemariti jer se u ovakvim zadacima obicno unose veliki brojevi da se sve
svodi na preciznost nalazenja broja 0.69 pa nam je svejedno da li je zadnji
clan uracunat ili ne, ali eto ako zelimo ipak 100% precizno uraditi, zadatak
moze se uraditi i bez grupisanja sa mijenjanjem predznaka sume, ali to uradite
sami :) */
while ((i<n)&&(n>1)) {
suma=suma+(1/i-1/(i+1));
i=i+2;
}
// Ispis sume
if (n>1) printf ("\n Suma je: %f", suma); else if (n=1) printf ("Suma je: 1");
// Kraj programa
getch();
return 0;
}
```

```
***C*C*O*M*G*C*C*G*M*O*C*C**
**C*C**R*G*A*M*N**++**N*M*A*G*R**C*C**
*C**P*R*I**++**I*R*P**C*
```

Autor: Murtic Nirmel
www.etf.ba

8. Napraviti program koji sa standardnog ulaza ucitava cijeli broj m koji predstavlja redni broj mjeseca u godini, te ucitava i broj g koji predstavlja godinu, a na monitoru ispisuje naziv tog mjeseca i broj dana u tom mjesecu. Pretpostaviti da je prestupna svaka cetvrta godina, iako je pravilo o racunanju prestupnih godina nešto složenije. Na primjer, ako su ulazne vrijednosti $m=2$, $g=2000$ program treba ispisati:

februar, 2000. g., broj dana:29

Ili ako su ulazne vrijednosti $m=5$, $g=2005$ program treba ispisati:
maj, 2005. g., broj dana: 31

RJEŠENJE:

```
#include <stdio.h>
main() {
/* Deklarisanje varijabli
Definisanje dvodimenzionalne matrice 12 sa 9 (12 mjeseci, a broj 9 je ustvari
broj slova od kojih se sastoje mjeseci, najvise ima septembar (9 slova). */
int m,g,d,i;
char mj[12][9]={{ "Januar"}, {"Februar"}, {"Mart"}, {"April"},
                 {"Maj"}, {"Juni"}, {"Juli"}, {"August"}, {"Septembar"},
                 {"Oktobar"}, {"Novembar"}, {"Decembar"}};
// U programu se uvodi i naredba „goto“ koja „skace“ sa linije na liniju
// programa.
pocetak:
// Unos mjeseca i godine
printf ("\n Unesi mjesec i godinu (m,g):"); scanf ("%d,%d", &m,&g);
// Ako se unese 1,3,5,7,8,10 ili 12 mjesec onda ima d=31 dan
if ((m==1)|| (m==3)|| (m==5)|| (m==7)|| (m==8)|| (m==10)|| (m==12)) d=31; else
// Ako se ipak unese 4,6,9 ili 11 mjesec onda ima d=30 dana
if ((m==4)|| (m==6)|| (m==9)|| (m==11)) d=30; else
/* Ili ako se ipak unese mjesec 2 onda se provjerava jos i prijestupna godina
(svaka godina djeljiva sa 4), i u zavisnosti da li je prijestupna ili ne d=29
odnosno d=29 dana. Ako broj nije od 1 do 12 vraca se na pocetak programa*/
if ((m==2)) if ((g%4==0)) d=29; else d=28; else goto pocetak;
// Ispis mjeseca koji se nalazi zapisan u mj[] (m-1 jer nam je januar pod
brojem 0)
printf("%s , %d. g., broj dana: %d", mj[m-1], g, d);
// Kraj programa
getch();
return 0;
}
```

```
***C*C*O*M*G*C*C*G*M*O*C*C***
**C*C**R*G*A*M*N**++**N*M*A*G*R**C*C**
*C**P*R*I**++**I*R*P**C*
```

Autor: Murtic Nirmel
www.etf.ba

9. Napisati program koji učitava realnu matricu dimenzija 10x10 te pronalazi najmanji element na glavnoj dijagonali.

RJEŠENJE:

```
#include <stdio.h>
main (){
// Deklarisanje varijabli
float A[10][10], MIN;
int i,j;
/* Unos matrice (koristimo 2 for petlje pri unosu (i su redovi a j su kolone).
Brojevi se unose po redovima.
*/
printf("\n Unesi matricu A[10][10]: \n");
for (i=0;i<10;i++)
for (j=0;j<10;j++)
{
printf(" [%d,%d]:", i+1, j+1);
scanf("%f", &A[i][j]);
}
/* Trazenje najmanjeg elementa na dijagonali!
Postupak je sljedeci. Elementi na dijagonali su tipa [i,j] gdje su i,j
jednaki. Tako dijagonalni elementi matrice 3x3 su [1,1], [2,2], [3,3]. Posto
nam trebaju samo elementi kada su (i,j) jednaki koristimo jednu for petlju. Na
pocetku stavimo da je najmanji element prvi element na poziciji [1,1], u C-u
je pozicija (0,0) ustvari pozicija (1,1) u realnom svijetu. I ulazimo u for
petlju. Ispituju se redom brojevi kod kojih su brojeci „i“ i „j“ jednaki. Ako
je neki od brojeva koji se ispituju manji od MIN onda se taj broj pridruzi
varijabli MIN. */
MIN=A[0][0];
for (i=1,j=1;i<10;i++,j++) if (A[i][j]<MIN) MIN=A[i][j];
// Ispis najmanjeg broja na dijagonali
printf("\nNajmanji element na glavnoj dijagonali je: %.f", MIN);
// Kraj programa
getch();
return 0;
}
```

```
***C**C**O**M**G**C**C**G**M**O**C**C**
**C**C**R**G**A**M**N**++**N**M**A**G**R**C**C**
**C**P**R**I**++**I**R**P**C**
```

Autor: Murtic Nirmel
www.etf.ba

10. Napišite funkciju *faktor* () koja izracunava $n!$ pri cemu je prototip funkcije *faktor* (): **double faktor (int n)**

Zatim, napisati glavni program koji ucitava neki prirodni broj **n, $1 < n < 10$** ; a zatim izracunava $n!$ pozivom funkcije **faktor()** i rezultat ispisuje na standardni izlaz.

RJEŠENJE:

```
#include <stdio.h>
main (){
// Deklarisanje varijabli
int N;
double F;
// Deklarisanje prototipa funkcije
double faktor (int n);
//Unos prirodnog broja
printf("\nUnesi prirodan broj N: ");
scanf("%d", &N);
/* Pozivanje funkcije faktor() i pridruzivanje vrijednosti koju funkcija
izracuna varijabli F */
F=faktor(N);
// Ispis faktorijele
printf("Faktorijela od %d je: %g", N, F);
// Kraj programa
getch();
return 0;
}

/* Funkcija koja racuna vrijednost faktorijele. Funkcija prima prirodan broj N
a vraca realan broj (double) kao rezultat. */
double faktor (int n) {
int i;
double fak=1;
/* Racunanje faktorijele se vrsi tako da se broj od kojeg trazimo faktorijelu
pomnozi sa svakim cijelim brojem manjim od sebe a koji je veci od 1. Na
pocetku je fak=1 jer ako bi bilo 0 dobili bi uzastopno mnozenje sa 0 i nebi
dobili ispravan rezultat */
for (i=n;i>1;i--) fak=fak*i;
// Vracanje izracunate faktorijele
return fak;
}
```

```
***C*C*O*M*G*C*C*G*M*O*C*C**
* * C C * * R G A M N * * + * * N M A G R * * C C * *
* C * * P * R * I * * * + * * * I * R * P * * C *
```

Autor: Murtic Nirmel
www.etf.ba

11. Napisati program koji učitava broj elemenata (n) nekog cjelobrojnog polja (Polje) a zatim i elemente tog polja (maksimalno 100). Program nakon toga pronalazi maksimalni element unesenog polja i ispisuje ga na standardni izlaz. Na primjer, ako su ulazni podaci: **n=4, Polje=(2,5,20,4)** program ce ispisati vrijednost broja 20.

RJEŠENJE:

```
# include <stdio.h>
main (){
// Definisavanje varijabli
int n, i, niz[100], max;
// Unos broja elemenata
printf("\nUnesi broj elemenata: ");
scanf("%d", &n);
// Unos niza od N elemenata (pomocu for petlje)
for (i=0;i<n;i++)
{
printf ("\nUnesi element broj %d: ", i+1);
scanf("%d", &niz[i]);
}
// Postavljanje vrijednosti max na vrijednost prvog clana niza
max=niz[0];
/* For petlja vrti brojeve od 2 broja (prvi je ukljucen u liniju iznad) i
provjerava se da li je i-ti broj veci od max i ako jeste varijabli max se
pridruzuje vrijednost tog elementa*/
for (i=1;i<n;i++)
if (max<niz[i]) max=niz[i];
// Ispis najveceg elementa niza
printf("Najveci element niza je: %d", max);
// Kraj programa
getch();
return 0;
}
```

```
* * * C * C * * O * M * G * C * C * G * M * O * * C * C * * *
* * C C * * R G A M N * * + * * N M A G R * * C C * *
* C * * P * R * I * * * + * * * I * R * P * * C *
```

Autor: Murtic Nirmel
www.etf.ba

12. Napisati program koji učitava 5 realnih elemenata dva polja A i B a zatim izračunava elemente trećeg polja C (koji također ima 5 elemenata) pri čemu se i -ti element polja C dobiva prema slijedećoj formuli:

$$C[i]=2*A[i]+B[i], i=1,2,\dots,n$$

Na kraju program na standardni izlaz ispisuje elemente sva tri polja: A, B i C.

RJEŠENJE:

```
# include <stdio.h>
main (){
// Definisane varijabli
float A[5], B[5], C[5];
int i;
/* Unos elemenata prvog polja (pomocu for petlje, i+1 zbog realnosti, da nam
ne ispisuje se na ekranu "Unesi element 0") */
for (i=0;i<5;i++){
printf("\nUnesi %d element prvog polja: ",i+1);
scanf("%f", &A[i]);
}
// Unos elemenata drugog polja
for (i=0;i<5;i++){
printf("\nUnesi %d element drugog polja: ",i+1);
scanf("%f", &B[i]);
}
// Racunanje clanova matrice C po formuli C=2*A+B (pomocu jedne for petlje).
for (i=0;i<5;i++) C[i]=2*A[i]+B[i];
// Ispis elemenata polja A
printf("\nElementi prvog polja A: ");
for (i=0;i<5;i++) printf("%.f ", A[i]);
// Ispis elemenata polja B
printf("\nElementi drugog polja B: ");
for (i=0;i<5;i++) printf("%.f ", B[i]);
// Ispis elemenata polja C
printf("\nElementi treceg polja C: ");
for (i=0;i<5;i++) printf("%.f ", C[i]);
// Kraj programa
getch();
return 0;
}
```

```
***C*C*O*M*G*C*C*G*M*O*C*C***
**C*C**R*G*A*M*N**++**N*M*A*G*R**C*C**
* C * * P * R * I * * * + * * * I * R * P * * C *
```

Autor: Murtic Nirmel
www.etf.ba

13. Napisati program koji učitava broj elemenata nekog polja (maksimalno 100) a zatim i elemente tog polja. Program zatim pronalazi najmanji element i na standardni izlaz ispisuje sve elemente koji su manji od dvostruke vrijednosti pronađenog najmanjeg elementa.

RJEŠENJE:

```
# include <stdio.h>
main (){
// Definisane varijabli
int n, i, niz[100], min;
// Unos broja elemenata
printf("\nUnesi broj elemenata: ");
scanf("%d", &n);
// Unos niza od N elemenata (pomocu for petlje)
for (i=0;i<n;i++){
printf ("\nUnesi element broj %d: ", i+1);
scanf("%d", &niz[i]);
}
// Postavljanje vrijednosti min na vrijednost prvog clana niza
min=niz[0];
/* For petlja vrti brojeve od drugog broja (prvi je ukljucen u liniju iznad) i
provjerava se da li je i-ti broj manji od min i ako jeste varijabli min se
pridruzuje vrijednost tog elementa. */
for (i=1;i<n;i++)
if (min>niz[i]) min=niz[i];
/* For petlja vrti sve brojeve i za svaki broj se vrši provjera da li je manji
od dvostruke vrijednosti najmanjeg broja, i ako je taj broj zaista
manji od 2*min ispisuje se na ekran, a ako nije ide se na iduci broj. */
for (i=0;i<n;i++)
if (niz[i]<min*2) printf("\n %d", niz[i]);
// Kraj programa
getch();
return 0;
}
```

```
***C**C**O**M**G**C**C**G**M**O**C**C**
**C**C**R**G**A**M**N**+**+**N**M**A**G**R**C**C**
**C**C**P**R**I**+**+**I**R**P**C**
```

Autor: Murtic Nirmel
www.etf.ba

14. Napisati funkciju koja kao argument prima cijeli broj a vraća broj koji se dobije tako da se iz argumenta izbacе cifre 2,3,8 i 9.

Prototip funkcije je: **int izbaci2389(int x)**

Nakon toga, napišite glavni program koji učitava neki cijeli broj, poziva funkciju **izbaci2389** i ispisuje rezultat na standardnom izlazu.

RJEŠENJE:

```
# include <stdio.h>
main (){
// Definisanje prototipa funkcije
int izbaci2389(int x);
// Definisanje varijabli
int N, X;
// unos prirodnog broja
printf("\nUnesi prirodan broj: ");
scanf("%d", &N);
// Poziv funkcije i pridruzivanje rezultata varijabli X
X=izbaci2389(N);
// Ispis prepravljenog broja
printf("\nPrepravljeni broj: %d", X);
// Kraj programa
getch();
return 0;
}

// Funkcija za izbacivanje cifri 2,3,8,9 iz prirodnog broja
int izbaci2389(int x)
{
int t=1, y=0, ost;
/* While petlja se izvrsava dok je x>0. Uzima se ostatak dijeljenja broja sa
10 (npr. od broja 129, ostatak je 9), zatim se varijabli x odbije zadnja cifra
dijeljenjem sa 10 (do odbijanja dolazi jer radimo sa int brojevima pa nema
decimalnog zarez a (npr 129/10=12). Zatim provjerimo dali je dobijeni ostatak
ost jednak 2,3,8 ili 9 i ako jest vracamo se na pocetak petlje i vrsimo isti
postupak za broj koji smo dobili odbijanjem zadnje cifre (i sve tako dok ne
odbijemo sve cifre, tj. dok ne dobijemo broj 0). Ako varijabla ost ipak je
razlicita od 2,4,8,9 onda se varijabli y doda vrijednost ost pomnozena sa t. U
programu varijabla t nam ustvari broji koja je decimala u pitanju. Kad je t=1
radimo sa jedinicama, kad je t=10 radimo sa deseticama... Tako npr. broj 1234
mozemo rastaviti kao 4*1+3*10+2*100+1*1000 gdje su nam brojevi 1,10,100,1000
ustvari nase t. */
while (x>0){
ost=x%10;
x=x/10;
if ((ost==2)|| (ost==3)|| (ost==8)|| (ost==9)) continue;
y=y+ost*t;
t=t*10;
}
return y;
}
```

```
***C*C*O*M*G*C*C*G*M*O*C*C***
**C**R**G**A**M**N**++**N**M**A**G**R**C**C**
* C * * P * R * I * * * + * * * I * R * P * * C *
```

Autor: Murtic Nirmel
www.etf.ba

15. Napisati funkciju koja kao argument uzima potrošnju u KWh za veliku i malu tarifu te na standardnom izlazu ispisuje racun za utrošenu elektricnu energiju.

Pretpostaviti da je cijena za veliku tarifu po utrošenom KWh 0.10 KM, a cijena za malu tarifu po utrošenom KWh je 0.05 KWh.

Prototip funkcije je: **void racun(float vt, float mt)** gdje :

* **vt** oznacava potrošnju za veliku tarifu

* **mt** oznacava potrošnju za malu tarifu

Nakon toga, napišite glavni program koji ucitava potrošnju za veliku tarifu i malu tarifu, te na standardnom izlazu ispisuje racun za tu potrošnju.

RJEŠENJE:

```
# include <stdio.h>
main (){
// Definisanje prototipa funkcije
void racun(float vt, float mt);
// Definisanje varijabli
float vt, mt;
// Unos potrosnje po velikoj i maloj tarifi
printf("Unesi potrosnju za veliku tarifu: ");
scanf("%f", &vt);
printf("Unesi potrosnju za malu tarifu: ");
scanf("%f", &mt);
// Poziv void funkcije (funkcija ne vraca rezultat), sa argumentima vt i mt
racun(vt, mt);
// Kraj programa
getch();
return 0;
}
// Funkcija koja kreira racun za utrosak energije na osnovu argumenata vt i mt
void racun(float vt, float mt)
{
float cijena;
// Racunanje ukupne cijene na osnovu argumenata vt i mt i zadatih cijena
cijena=vt*0.10+mt*0.05;
// Ispis rezultata
printf("\n-----RACUN ZA UTROSENU EL. ENERGIJU----- ");
printf("\nPotrosnja VT: %f KWh", vt);
printf("\nPotrosnja MT: %f KWh", mt);
printf("\n***** ");
printf("\nUkupno: %f KM", cijena);
printf("\n***** ");
}
```

```
***C*C*O*M*G*C*C*G*M*O**C*C**
**C*C**R*G*A*M*N**++**N*M*A*G*R**C*C**
*C**P*R*I**+**I*R*P**C*
```

Autor: Murtic Nirmel
www.etf.ba

16. Napisati funkciju koja kao argument uzima tri cijela broja **a**, **p** i **c**. Funkcija vraća cijeli broj koji se dobije tako da se u broju **a** iza **p**-te cifre s desne strane umetne cifra **c**.

Ako je **p** veći od ukupnog broja cifri izlaz iz funkcije treba biti jednak ulazu. Ako se broj **c** sastoji od više cifri onda se umeće samo zadnja cifra iz tog broja.

Prototip funkcije je: **int umetni(int a, int p, int c)**

Nakon toga, napišite program koji učitava neki pozitivan cijeli broj **a**, poziciju umetanja cifre **p** i broj koji definira cifru koja se umeće **c**, te poziva funkciju **umetni()** i ispisuje na standardnom izlazu rezultirajući broj.

RJEŠENJE:

```
# include <stdio.h>
main (){
// Definisane prototipa funkcije
int umetni(int a, int p, int c);
// Definisane varijabli
int a, p, c, rez;
// Unos brojeva a,p,c
printf("Unesi brojeve a, p, c: ");
scanf("%d,%d,%d", &a,&p,&c);
// Poziv funkcije umetni() i pridruzanje rezultata varijabli rez
rez=umetni(a,p,c);
// Ispis rezultata
printf("Rezultat: %d", rez);
// Kraj programa
getch();
return 0;
}

// Funkcija koja umeće broj c na p-tu poziciju u broju a s desna
int umetni(int a, int p, int c)
{
int t=1, ost, x=0,br=0;
// Broju c pridruzuje se samo njegova zadnja cifra
c=c%10;
/* Racuna se ostatak dijeljenja sa 10 (ost), broju a se "otkida" zadnja cifra.
Ukoliko su brojevi p i br (brojac) identicni na to mjesto treba ubaciti broj
c. t nam broji decimale (vidi 14. zadatak).
Varijabli x se dodaje ostatak ost pomnozen sa t.
X je trazeni broj koji funkcija vraća u glavni program. */
while (a>0){
ost=a%10;
a=a/10;
if (p==br) {x=x+c*t; t=t*10; p=1000;}
x=x+ost*t;
t=t*10;
br++;
}
return x;
}
```

```
***C*C*O*M*G*C*C*G*M*O*C*C**
**C*C**R*G*A*M*N**++**N*M*A*G*R**C*C**
*C**P*R*I**++**I*R*P*C*
```

Autor: Murtic Nirmel
www.etf.ba

17. Napraviti program koji ce na monitoru ispisati prvih N cijelih brojeva i njihove kvadrate u formatu:

```
1 na kvadrat je 1
2 na kvadrat je 4
3 na kvadrat je 9
.....
```

Broj N se unosi sa tastature. Koristiti **for** petlju.

RJEŠENJE:

```
#include <stdio.h>

int main() {
// Definisanje varijabli
    int n,i;

/* Unos broja n sa tastature */

    printf("\nUnesi broj n: ");
    scanf("%d", &n);

/* Ispis prvih n brojeva i njihovih kvadrata, krenuvsi od broja 1, i znaci
krece od 1 i ide sve do broja za jedan manji od unesenog */

    for ( i = 1 ; i < n ; i++ )
        printf("\n %d na kvadrat je %d \n", i, i*i);

// Kraj programa
    getch();
    return 0;
}
```

```

***C*C*O*M*G*C*C*G*M*O*C*C**
* * C C * * R G A M N * * + * * N M A G R * * C C * *
* C * * P * R * I * * * + * * * I * R * P * * C *

```

Autor: Murtic Nirmel
www.etf.ba

18. Napraviti program koji izracunava rješenja sistema linearnih jednačbi:

$$\begin{aligned}
 a_{11}x_1 + a_{12}x_2 &= b_1 \\
 a_{21}x_1 + a_{22}x_2 &= b_2 \\
 (\text{nepoznate su } x_1 \text{ i } x_2)
 \end{aligned}$$

Koeficijenti a_{11} , a_{12} , b_1 , a_{21} , a_{22} , b_2 se unose sa tastature. Ako sistem jednačbi nema rješenje program ispisuje poruku «RJEŠENJE ne postoji».

RJEŠENJE:

```

#include <stdio.h>
int main() {
// Definisane varijabli
float a11,a21,a12,a22,x1,x2,b1,b2;
/* Unos podataka (U dijelu %fx1%fx2=%f izmedju %fx1 i %fx2 nema plusa, u
suprotnom bi morali pri unosu operacije - (minus) upisivati +-) */
printf("\n Unesite 1 linearnu jednacinu a11x1+a12x2=b1 (x1, x2 nepoznate): ");
scanf("%fx1%fx2=%f", &a11, &a12, &b1);
printf("\n Unesite 2 linearnu jednacinu a21x1+a22x2=b2 (x1, x2 nepoznate): ");
scanf("%fx1%fx2=%f", &a21, &a22, &b2);
/* Izracunavanje x2 metodom supstitucije */
x2=(b2 * a11 - a21 * b1) / (a22 * a11-a21 * a12);
/* Izracunavanje x1 metodom uvrstavanja x2 u prvu jednacinu */
x1=(b1 - a12 * x2) / a11;
/* Provjerava se da li je sistem odredjen */
/* Ukoliko je sistem odredjen ispisuju se rjesenja, u suprotnom se ispisuje
poruka: RJEŠENJE ne postoji */
if ((a11*a22)==(a12*a21)) printf("\nRJEŠENJE ne postoji!!!");
else printf("\nRezultat je: x1=%.2f i x2=%.2f", x1, x2);
// Kraj programa
getch();
return 0;
}

```

```
***C**C**O**M**G**C**C**G**M**O**C**C**  
**C**C**R**G**A**M**N**+**+**N**M**A**G**R**C**C**  
*C**P**R**I**+**+**I**R**P**C*
```

Autor: Murtic Nirmel
www.etf.ba

19. Napisati program koji ucitava 4 cijela broja, a kao izlaz daje dva najveca broja. Na primjer, ako korisnik kao ulaz unese slijedece cijele brojeve:

23 5 45 33

rezultat izvodjenja programa treba biti:

Najveci broj je 45.

Drugi broj po velicini je 33.

Napomena: Tekstualne poruke vašeg programa trebaju biti iste kao gore navedene.

RJEŠENJE:

```
#include <stdio.h>  
  
int main() {  
    // Definisanje varijabli  
    int i,j,p,niz[4];  
    /* Unos brojeva niza (4 broja), pomocu for petlje koja ide od 1 do 4. */  
    printf("\n Unesite 4 cijela broja!");  
    for(i=1;i<=4;i++){  
        printf("\n %d broj: ", i);  
        scanf("%d",&niz[i]);  
    }  
  
    /* Proces sortiranja  
    Koristena je metoda bubblesort (spomenuta u jednom od zadataka ranije), to je  
    metoda „preljevanja“, provjerava se da li je j-ti clan manji od j+1-vog i ako  
    jeste mijenjaju se mjesta.  
    Npr. Unesemo 1,2,3,4. Nakon prvog prolaska imamo 2,1,3,4. Nakon drugog  
    prolaska 2,3,1,4. Nakon treceg prolaska imamo 2,3,4,1. I onda prva for petlja  
    proces ponavlja jos jednom. Pa imamo 3,2,4,1 zatim i 3,4,2,1 pa onda ostaje  
    isto jer je 2 vec vece od 1. Ali niz nije sortiran! Prva for petlja jos jednom  
    pokrece postupak i imamo 4,3,2,1. Proci ce se onako bezveze jos 2 puta kroz  
    petlju „j“ i zatim se izlazi iz petlje, i ispisuju se rezultati*/  
  
    for(i=1;i<=3;i++)  
        for(j=1;j<=3;j++){  
            if (niz[j+1]>niz[j]) {  
                p=niz[j];  
                niz[j]=niz[j+1];  
                niz[j+1]=p;  
            }  
        }  
  
    /*Ispis rjesenja zadatka*/  
  
    printf("\n Najveci broj je %d", niz[1]);  
    printf("\n Drugi broj po velicini je %d", niz[2]);  
  
    // Kraj programa  
    getch();  
    return 0;  
}
```

```
***C*C*O*M*G*C*C*G*M*O**C*C**
**C*C**R*G*A*M*N**++**N*M*A*G*R**C*C**
*C**P*R*I**++**I*R*P**C*
```

Autor: Murtic Nirmel
www.etf.ba

20. Napisati program koji ce sa tastature ucitati prirodan broj, te ispitati da li je broj prost. Ako se unese broj koji nije prirodan ($n \leq 0$), program ispisuje poruku „**Broj nije prirodan**“ i završava. Ako je broj prost program ispisuje poruku „**Uneseni broj je prost**“. Ako broj n nije prost program ispisuje poruku „**Uneseni broj je slozen**“. Ako je unesen broj $n=1$ program ispisuje poruku: „**Broj 1 nije niti prost niti slozen**“.

Napomena: Pod prostim brojem podrazumijevamo prirodan broj koji se ne može napisati kao umnožak dvaju prirodnih brojeva manjih od tog broja. Broj 1 nije niti prost niti složen.

RJEŠENJE:

```
#include <stdio.h>

main () {

// Definisavanje varijabli
int i,br, ispit;
// Unos jednog prirodnog broja
printf ("\n Unesi prirodan broj: ");
scanf ("%d", &br);
// Provjera da li je broj prirodan, i ako nije ispisuje
// se poruka da broj nije prirodan
if (br<=0) printf("\n Broj nije prirodan"); else
// Provjera da li je uneseni broj, broj 1, i ako jeste,
// ispisuje se odgovarajuca poruka
if (br==1) printf("\n Broj 1 nije ni prost ni slozen"); else
/* Ako je unesen prirodan broj veci od 1, provjerava se da li je broj prost
ili slozen, varijabla "ispit" kao rezultat. Djelimo broj „br“ sa svakim brojem
manjim od polovine unesenog broja. Na pocetku je „ispit=0“ sto nam govori da
je broj neparan. Ulazi se u petlju i pocinje dijeljenje sa brojevima 2,3,4...
Ako se broj „br“ uspije podijeliti sa nekim od brojeva varijabla ispit ce
automatski poprimiti vrijednost 1. U zavisnosti da li je broj „br“ uspio se
podijeliti sa nekim brojem bez ostatka, zavisiti ce i cinjenica da li je broj
prost ili slozen. */
{
ispit=0; i=2;
while (i<br/2) {
if ((br % i == 0)) ispit=1;
i++;
}
// Ispis odgovarajuce poruke na izlazu (PROST ILI SLOZEN)
if (ispit) printf("\n Broj je slozen"); else printf("\n Broj je prost");}
// Kraj programa
getch();
return 0;
}
```

```
***C**C**O**M**G**C**C**G**M**O**C**C**
**C**C**R**G**A**M**N**++**N**M**A**G**R**C**C**
**C**C**P**R**I**++**I**R**P**C**
```

Autor: Murtic Nirmel
www.etf.ba

21. Napraviti program koji izracunava $n!$ (n faktorijela). Broj n se unosi sa tastature i treba biti u rasponu **[1,20]**. Ako broj n nije u datom rasponu, program ispisuje poruku „Uneseni broj nije u datom rasponu“ i završava se program.

Napomena: $n!$ je definirano kao $n! = n \cdot (n-1) \cdot (n-2) \cdot \dots \cdot 3 \cdot 2 \cdot 1$.

RJEŠENJE:

```
#include <stdio.h>

main () {

// Definisavanje varijabli
int i;
float n, f;

// Unos prirodnog broja
printf("\n Unesi broj N u rasponu od [1,20]: ");
scanf("%f", &n);

/* Racunanje faktorijele, ukoliko je broj u rasponu [1,20]. Faktorijela se
racuna tako sto se uneseni broj pomnozi sa svakim brojem manjim od sebe, i sve
se to pridruzuje varijabli „f“. */
f=1;
if ((n>0) && (n<21)) {
    for (i=0;i<n;i++) f=f*(n-i);

// Ispis rezultata
printf("\n Faktorijela od broja %.0f je: %0.f", n, f);

// Ispis poruke ukoliko broj n nije u odgovarajućem rasponu
} else printf ("\n Uneseni broj nije u datom rasponu!");

// Kraj programa
getch();
return 0;
}
```

```
***C* C* *O* M* G* C* C* G* M* O* *C* C* * *
* * C C * * R G A M N * * + * * N M A G R * * C C * *
* C * * P * R * I * * * + * * * I * R * P * * C *
```

Autor: Murtic Nirmel
www.etf.ba

22. Napisati program koji ce izracunati sumu kvadrata prvih n prirodnih brojeva. Broj n se unosi sa tastature i treba biti u intervalu **[2,100]**. Ako broj n nije u tom intervalu program ispisuje poruku „Broj nije u zadanom intervalu“ i završava. Na primjer, ako se unese $n=3$, program ce izracunati sumu: **suma=1²+2²+3²=14**.

RJEŠENJE:

```
#include <stdio.h>
main () {
// Definisanje varijabli
    int i,broj,suma;
// Unosenje prirodnog broja sa tastature u rasponu od [2,100]
    printf("\n ULAZ: Broj \"N\" u rasponu od [2,100]: ");
    scanf("%d", &broj);

/* Ukoliko se unese broj u tacnom intervalu [1,100] racunanje sume kvadrata.
Imamo varijablu suma (na pocetku 0), koja se krece sabirati redom sa:
1*1 + 2*2 + 3*3 + 4*4 ... I sve do unesenog broja.*/
    suma=0;
    if ((broj>1) && (broj<101)) {
        for (i=1;i<=broj;i++) suma=suma+(i*i);
    }

// Ispis sume kvadrata prvih n brojeva
    printf("\n Suma kvadrata prvih %d brojeva: %d", broj, suma);

// Ispis odgovarajuce poruke ukoliko broj n nije u odgovarajucem rasponu
    } else printf ("\n Uneseni broj nije u datom rasponu!");

// Kraj programa
    getch();
    return 0;
}
```

```
***C*C*O*M*G*C*C*G*M*O*C*C***
**C*C**R*G*A*M*N**++**N*M*A*G*R**C*C**
*C**P*R*I**++**I*R*P**C*
```

Autor: Murtic Nirmel
www.etf.ba

23. Napraviti program koji ce na ekranu ispisivati broj PI prema formuli $PI=1-1/3+1/5-1/7\dots 1/i$, a vrijednost **eps** unosimo sa tastature i mora biti u intervalu **0.1** do **0.0001**. Suma se racuna sve dok je **1/i** vece od **eps**.

RJEŠENJE:

```
#include <stdio.h>

main () {

// Definisane varijabli
float eps,pi,n;

// Unos broja koji oznacava preciznost
printf("\n Unesi broj eps u rasponu od [0.0001,0.1]: ");
scanf("\n %f", &eps);

/* Provjera da li je broj „eps“ iz odredjenog intervala, i ako jeste racunanje
broja „pi“. Radi lakseg sastavljanja formule (jer imamo - i + naizmjenicno)
grupisacemo clanove po 2. i onda sve svodimo na operaciju sabiranja razlika 2
broja. Na kraju se brojac uvecava za 4 jer smo odmah po dva clana u formuli
koristili */
n=1; pi=0;
if ((eps>0.0001) && (eps<0.1)) {
while (1/n>eps) {
pi=pi+(4/n-4/(n+2));
n=n+4;
}

// Ispis broja PI
printf ("PI je: %f", pi);

// Ispis poruke ukoliko broj eps nije u odgovarajucem rasponu
} else printf ("\n Broj nije u trazenom rasponu!");
// Kraj programa
getch();
return 0;
}
```

```
***C* C* *O* M* G* C* C* G* M* O* *C* C* * *
* * C* C* * R* G* A* M* N* * * + * * N* M* A* G* R* * * C* C* * *
* C* * * P* * R* * I* * * * + * * * * I* * R* * P* * * C* *
```

Autor: Murtic Nirmel
www.etf.ba

24. Napisati C program koji ce učitati dimenzije (**RA x KA**) realne matrice **A** (maksimalno 10 x 10) , dimenzije (**RB x KB**) realne matrice **B** (maksimalno 10 x 10), te učitati elemente tih matrica. Program zatim treba izračunati matricu **C** koja je proizvod matrica **A** i **B**. Na kraju program treba ispisati sve tri matrice: **A**, **B** i **C**.

* **RA**, **RB** oznacavaju broj redova matrica **A** i **B**, respektivno;

* **KA**, **KB** oznacavaju broj kolona matrica **A** i **B**, respektivno.

Prilikom unosa dimenzija matrica program treba provjeriti da li su unesene dimenzije u dozvoljenim granicama, tj. da li je **RA<=10**, **RB<=10**, **KA<=10**, **KB<=10**, te isto tako program treba provjeriti da li se matrice mogu množiti, tj. da li je ispunjeno: **KA=RB**

RJEŠENJE:

```
#include <stdio.h>
main(){
float A[10][10], B[10][10], C[10][10];
int RA, RB, KA, KB, i, j, k;
// Unos dimenzija prve i druge matrice
printf("\nUnos dim. matrica A[RA,KA], B[RB,KB], u obliku \"RA,KA,KB\": ");
scanf("%d,%d,%d,%d", &RA, &KA, &RB, &KB);

// Ispit uslova: da li su dimenzije u intervalu od 1 do 10.
if ((RA<=10)&&(RA>0)&&(RB<=10)&&(RB>0)&&(KA<=10)&&(KA>0)&&(KB<=10)&&(KB>0)) {
// Ispit uslova: da li je broj kolona prve matrice i broj redova druge isti.
if (KA==RB){

/* Unos elemenata prve matrice A (pomocu 2 for petlje, objasnjeno u jednom od
prethodnih zadataka. */
printf("\nUnesi elemente matrice A[%d,%d]: \n", RA, KA);
for (i=0;i<RA;i++){
for (j=0;j<KA;j++){
printf("Unesi element [%d,%d]: ", i+1, j+1);
scanf("%f", &A[i][j]); }

// Unos elemenata druge matrice B
printf("\nUnesi elemente matrice B[%d,%d]: \n", RB, KB);
for (i=0;i<RB;i++){
for (j=0;j<KB;j++){
printf("Unesi element [%d,%d]: ", i+1, j+1);
scanf("%f", &B[i][j]); }

/* Racunanje matrice C po formuli C=A*B. Dvije for petlje identicne for
petljama pri unosu matrice ili ispisu, koje „setaju“ po redovima i kolonama
matrica A,B,C. Prema matematickom postupku proizvod matrica A i B je:
C[1][1]=A[0][0] * B[0][0] + A[0][1] * B[1][0] + ... + A[0][n] * B[n][0] sto je
primjenjeno na formulu koja se nalazi unutar for petlje sa argumentom „k“.
Argument „k“ se krece od 1 do broja kolona prve matrice, i predstavlja broj
sabitaka u formuli iznad. C[i][j] = 0; je uvedeno jer vrijednost C[i][j] nije
definisana, a moramo je koristiti u formuli unutar for petlje po „k“. */
```

```
*** C * C * * O * M * G * C * C * G * M * O * * C * C * * *
* * C * C * * R * G * A * M * N * * * + * * * N * M * A * G * R * * * C * C * *
* C * * P * R * I * * * + * * * I * R * P * * C *
```

Autor: Murtic Nirmel
www.etf.ba

```
for (i=0;i<RA;i++)
for (j=0;j<KB;j++) {
C[i][j]=0;
for (k=0;k<KA;k++) C[i][j]=C[i][j]+A[i][k]*B[k][j]; }
// Ispis matrice A
printf("\nMatrica A[%d,%d]: \n", RA, KA);
for (i=0;i<RA;i++){
printf("\n");
for (j=0;j<KA;j++) printf("%4.f ", A[i][j]); }
// Ispis matrice B
printf("\nMatrica B[%d,%d]: \n", RB, KB);
for (i=0;i<RB;i++){
printf("\n");
for (j=0;j<KB;j++) printf("%4.f ", B[i][j]); }
// Ispis matrice C
printf("\nMatrica C[%d,%d]: \n", RA, KB);
for (i=0;i<RA;i++){
printf("\n");
for (j=0;j<KB;j++) printf("%4.f ", C[i][j]); }
// Ispis poruka ukoliko podaci nisu regularni
} else printf("\n Matrice se nemogu pomnoziti!");
} else printf("\n Matrice nisu u odgovarajucem intervalu!");
// Kraj programa
getch();
return 0;
}
```

```
***C*C*O*M*G*C*C*G*M*O**C*C**
**C*C**R*G*A*M*N**++**N*M*A*G*R**C*C**
*C**P*R*I**++**I*R*P**C*
```

Autor: Murtic Nirmel
www.etf.ba

25. Napisati C program koji sa standardnog ulaza učitava elemente realne matrice dimenzija 5 x 5, te izračunava sumu elemenata svake kolone i sumu elemenata svakog reda.

Za unos i pohranjivanje matrice koristite dvodimenzionalno polje, dok za pohranjivanje suma po redovima i kolonama trebate koristiti jednodimenzionalno polje.

Program takodjer treba pronaci:

*** u kojoj koloni je najmanja suma,**

*** u kojem redu je najveća suma,**

te ispisati na standardni izlaz unesenu matricu, redne brojeve i sume pronadjene kolone, odnosno, reda.

RJEŠENJE:

```
#include <stdio.h>
main(){

// Definisavanje varijabli
float MATRIX[5][5], SUMA[5], sumaR, sumaK;
int i,j, Kolona, Red;

// Unos elemenata matrice
printf("\nUnesi elemente matrice MATRIX[5,5]: \n");
for (i=0;i<5;i++)
for (j=0;j<5;j++){
printf("[%d,%d]: ", i+1, j+1);
scanf("%f", &MATRIX[i][j]); }

/* Postavljanje vrijednosti elemenata niza SUMA na 0 (jer vrijednosti niza
SUMA[] nisu definisane. */
for (i=0;i<5;i++) SUMA[i]=0;

/* Racunanje suma po redovima. Sabiramo sve brojeve prvog reda, znaci „i“ se
ne mijenja, a „j“ se uvecava za 1 sve do 5. I kada se sabere tih 5 brojeva, to
je suma prvog reda i zapisuje se u SUMA[0], zatim se isto ponavlja za ostale
redove, tako sto se samo „i“ uvecava za 1 i prelazi time na iduci red. */
for (i=0;i<5;i++)
for (j=0;j<5;j++)
SUMA[i]=SUMA[i]+MATRIX[i][j];

/* Nalazenje najveće od izracunatih suma. Postavlja se da je prva suma
najveća, i zatim se provjerava da li od preostalih ima neka koja je veća, i
ako je veća, onda se ta suma postavlja da je najveća. Vrijednost „i+1“ nam
predstavlja broj reda u kojem je ta suma (na pocetku stavili smo da je „i=1“
jer ispred stoji da je „sumaR=SUMA[0]“. */
sumaR=SUMA[0]; Red=1;
for (i=1;i<5;i++) if (sumaR<SUMA[i]) {sumaR=SUMA[i]; Red=i+1;}

// Postavljanje vrijednosti elemenata niza SUMA na 0
for (i=0;i<5;i++) SUMA[i]=0;
```

```
***C*C*O*M*G*C*C*G*M*O*C*C**
* C C * * R G A M N * * + * * N M A G R * * C C * *
* C * * P * R * I * * * + * * * I * R * P * * C *
```

Autor: Murtic Nirmel
www.etf.ba

```
/* Racunanje suma po kolonama... ista procedura, samo smo zamijenili mjesta
brojaca „i, j“ unutar „MATRIX[i][j]“. */
for (i=0;i<5;i++)
for (j=0;j<5;j++)
SUMA[i]=SUMA[i]+MATRIX[j][i];

// Pronalazenje najmanje od izracunatih suma
sumaK=SUMA[0]; Kolona=1;
for (i=1;i<5;i++) if (sumaK>SUMA[i]) {sumaK=SUMA[i]; Kolona=i+1;}

// Ispis matrice MATRIX
printf("\nMatrica MATRIX[5,5]: \n");
for (i=0;i<5;i++){
printf("\n");
for (j=0;j<5;j++) printf("%.f ", MATRIX[i][j]); }

// Ispis rjesenja
printf("\nU %d koloni je najmanja suma koja iznosi %.f", Kolona, sumaK);
printf("\nU %d redu je najveca suma koja iznosi %.f", Red, sumaR);

// Kraj programa
getch();
return 0;
}
```

```
***C*C**O*M*G*C*C*G*M*O**C*C**
**C**R*G*A*M*N**++**N*M*A*G*R**C*C**
*C**P*R*I**++**I*R*P**C*
```

Autor: Murtic Nirmel
www.etf.ba

26. Program ucitava jednu rijec **a** i znak **c** koji se treba prebrojati u toj rijeci. Nakon toga program poziva funkcije **duzina()** i **izbroji()** i **samogl()**, te ispisuje dužinu unesene rijeci **a** i broj pojavljivanja znaka **c** u toj rijeci, kao i broj samoglasnika u rijeci.

RJEŠENJE:

```
#include <stdio.h>
// Funkcije „duzina“, „izbroji“ i „samogl“
// Funkcija duzina prima pokazivac na prvo slovo nekog stringa, a zatim u
while petlji provjerava se koliko znakova ima rijec, kada se dodje do nultog
znaka, izlazi se iz petlje, rezultat je varijabla „d“ koja se vraca iz
funkcije.
int duzina(char *string)
{
int d=0;
while (*string++!='\0') d++;
return d;
}

/* Slicno prvoj funkciji, samo funkcija pored pokazivaca na prvi znak stringa
prima i znak koji se broji. I umjesto da broji se broj clanova, umjesto d++
ide uslov (ako je slovo na koje pokazuje pokazivac nase slovo), „b“ se uveca
za 1. Kada „dodje“ while petlja do nultog znaka, izlazi se iz petlje a
varijabla „b“ se vraca kao rezultat.
NAPOMENA: U gornjoj funkciji stoji u uslovu *string++ sto prebacava pokazivac na
iduci clan nakon sto while petlja napravi jedan „krug“, dok u drugoj funkciji
stoji u uslovu samo *string ali je poslije if petlje uvecan pokazivac za „1“,
sto nam daje opet isti efekat. */
int izbroji(char *string, char c)
{
int b=0;
while (*string!='\0')
{
if (*string==c) b++;
string++;
}
return b;
}

/* Identicno funkciji izbroji() samo umjesto provjere „*string==c“ ide uslov
da li je „*string“ neki od samoglasnika, a ne znak koji unosimo. Funkcija
prima samo pokazivac bez znaka. */
int samogl (char *string)
{
int b=0;
while(*string!='\0')
{
if (*string=='a' || *string=='A' || *string=='E' || *string=='e' || *string=='I' ||
*string=='i' || *string=='O' || *string=='o' || *string=='U' || *string=='u') b++;
string++;
}
return b;
}
```

```
***C**C**O**M**G**C**C**G**M**O**C**C**  
**C**C**R**G**A**M**N**++**N**M**A**G**R**C**C**  
*C**P**R**I**++**I**R**P**C*
```

Autor: Murtic Nirmel
www.etf.ba

```
int main()  
{  
  // Definisiranje varijabli  
  char a[80],c;  
  // Unos rijeci bez razmaka i jednog slova koje ce se prebrojavati  
  printf("\nUnesite jednu rijec do 20 znakova (bez razmaka:");  
  scanf("%s",a);  
  printf("\nKoji znak treba prebrojati:");  
  scanf("\n%c",&c);  
  // Ispis rjesenja  
  printf("\nUnesena rijec sadrzi %d znakova!",duzina(a));  
  printf("\nBroj znakova %c je: %d",c,izbroji(a,c));  
  printf("\nBroj samoglasnika je: %d", samogl(a));  
  // Kraj programa  
  getch();  
  return 0;  
}
```

```
***C**C**O**M**G**C**C**G**M**O**C**C**  
**C**C**R**G**A**M**N**+**+**N**M**A**G**R**C**C**  
*C**P**R**I**+**+**I**R**P**C*
```

Autor: Murtic Nirmel
www.etf.ba

27. Napisati program koji u realno polje **temp** učitava 10 realnih vrijednosti koje predstavljaju temperature u 10 razlicitih dnevnih termina. Program zatim poziva funkcije **maxtemp()** i **prtemp()** koje izracunavaju maksimalnu dnevnu temperaturu i prosjecnu dnevnu temperaturu i ispisuje na standardnom izlazu te vrijednosti.

RJEŠENJE:

```
#include <stdio.h>  
main ()  
{  
    // Definisavanje varijabli  
    float temp[10], *p;  
    int i;  
    // Definisavanje prototipa funkcija  
    float maxtemp(float *p);  
    float prtemp(float *p);  
    // Unos 10 temperatura u niz temp[];  
    for (i=0;i<10;i++)  
    {  
        printf("\nUnesi temperaturu br.%d: ",i+1);  
        scanf("%f", &temp[i]);  
    }  
    // Postavljanje pokazivaca p na prvi clan niza temp[];  
    p=temp;  
    // Isipis rezultata uz poziv funkcija  
    printf("\nMaksimalna temperatura je: %.1f", maxtemp(p));  
    printf("\nProsjecna temperatura je: %.1f", prtemp(p));  
    // Kraj programa  
    getch();  
    return 0;  
}  
/* Funkcija maxtemp() prima pokazivac na realan niz i pomocu for petlje (vidi  
11. zadatak) pronalazi se najveći broj (temperatura i vraca se iz funkcije */  
float maxtemp(float *p)  
{  
    float max=*p;  
    int i;  
    for (i=0;i<10;i++) { if (max<*p) max=*p; p++; }  
    return max;  
}  
/* Funkcija prtemp() prima pokazivac na realan niz i pomocu for petlje se  
saberu svi clanovi niza i na kraju (poslije for petlje) se podijeli sve sa 10  
(broj temperatura), tj. vraca se vrijednost (pr/10) kao rezultat */  
float prtemp(float *p)  
{  
    float pr=0;  
    int i;  
    for (i=0;i<10;i++) { pr=pr+*p; p++; }  
    return (pr/10);  
}
```

```
***C*C*O*M*G*C*C*G*M*O*C*C**
**C*C*RGAMN**++**NMAGR**C*C**
* C * * P * R * I * * * + * * * I * R * P * * C *
```

Autor: Murtic Nirmel
www.etf.ba

28. Modificirati prethodni program (27) tako da program učitava cijeli broj n a zatim se sa tastature unosi n vrijednosti za temperature u polje **temp**. Pretpostaviti da je $n \leq 10$. U skladu s tim potrebno je modificirati funkcije **maxtemp()** i **prtemp()** tako da imaju slijedeće prototipove:

```
float maxtemp(int n, float *p);
float prtemp(int n, float *p);
```

RJEŠENJE:

```
# include <stdio.h>
main ()
{
// Definisane varijabli
float temp[10], *p;
int i, n;
// Definisane prototipa funkcija
float maxtemp(int n, float *p);
float prtemp(int n, float *p);
// Unos broja n
printf("\nUnesi broj temperatura: ");
scanf("%d", &n);
// Unos n temperatura u niz temp[];
for (i=0;i<n;i++)
{
printf("\nUnesi temperaturu br.%d: ",i+1);
scanf("%f", &temp[i]);
}
// Postavljanje pokazivaca p na prvi clan niza temp[];
p=temp;
// Isipis rezultata uz poziv funkcija
printf("\nMaksimalna temperatura je: %.1f", maxtemp(n, p));
printf("\nProsjecna temperatura je: %.1f", prtemp(n, p));
// Kraj programa
getch();
return 0; }

/* Funkcija prima pokazivac na realan niz, i broj n i pomocu for petlje (vidi
11. zadatak) pronalazi se najveći broj (temperatura i vraca se iz funkcije */
float maxtemp(int n, float *p) {
float max=*p;
int i;
for (i=0;i<n;i++) { if (max<*p) max=*p; p++; }
return max; }

/* Funkcija prima pokazivac na realan niz i broj „n“ i pomocu for petlje se
saberu svi clanovi niza i na kraju (poslije for petlje) se podijeli sve sa
brojem „n“ (broj temperatura), tj. vraca se vrijednost (pr/n) kao rezultat */
float prtemp(int n, float *p){
float pr=0;
int i;
for (i=0;i<n;i++) { pr=pr+*p; p++; }
return (pr/n); }
```

```
***C*C*O*M*G*C*C*G*M*O*C*C**
**C*C**R*G*A*M*N**++**N*M*A*G*R**C*C**
*C**P*R*I**+**I*R*P**C*
```

Autor: Murtic Nirmel
www.etf.ba

29. Napisati funkciju s imenom **adresa()** koja kao argument uzima dva stringa koji predstavljaju **ime** i **prezime** a kao rezultat vraca se novi string koji se dobije tako da se string koji predstavlja **prezime** promijeni u string koji predstavlja e-mail adresu slijedeceg oblika: **[prezime].[ime]@etf.unsa.ba**
Prototip funkcije je: **char *adresa(char *ime, char *prezime)**
Na primjer, ako su argumenti **ime="Marko"**, **prezime="Markovic"** funkcija **adresa()** vraca pokazivac na rezultirajuci string **prezime** koji je promijenjen u novi string oblika:
Markovic.Marko@etf.unsa.ba.

Nakon toga, napisati glavni program koji ucitava ime i prezime neke osobe i na standardnom izlazu ispisuje toj osobi pridruženu **e-mail** adresu koja se dobije pozivom funkcije **adresa()**.

RJEŠENJE:

```
#include <stdio.h>
// Funkcija adresa()
char *adresa(char *ime, char *prezime)
{
    // Definisane varijabli potrebnih za rad u funkciji
    char *p, POMOC[]="@etf.unsa.ba", *POM;
    /* Pointer „p“ pokazuje na pocetak prezimena, a „POM“ pokazuje na pomocni niz
    „POMOC“ u kojem se nalazi string „@etf.unsa.ba.“ */
    p=prezime; POM=POMOC;

    // Generisanje email adrese
    // Ova while petlja pokazivac postavi iza zadnjeg znaka, tj. Na nulti znak.
    while (*p!='\0') p++;
    // Umjesto nultog znaka upise se „.“ i predje se na iduci znak
    *p='.'; p++;
    // Poslije tacke se ubacaju karakteri iz niza na koji pokazuje pokazivac „ime“
    while (*ime!='\0') { *p=*ime; ime++; p++; }
    // Poslije imena se doda string na koji pokazuje pokazivac „POM“
    while (*POM!='\0') { *p=*POM; POM++; p++; }
    // Vracanje pokazivaca iz funkcije
    return p;
}

// Glavni program
int main(){
    // Definisane varijabli
    char *ime, *prezime, *c;
    char U1[100], U2[100];
    // Unos podataka ime i prezime
    printf("\nUnesi ime:");
    scanf("%s", &U1);
    printf("\nUnesi prezime:");
    scanf("%s", &U2);
    // Postavljanje pokazivaca „c“ na pocetak prezimena
    c=U2;
    // Postavljanje pokazivaca ime i prezime na stringove U1 i U2
```

```
*** C * C * * O * M * G * C * C * G * M * O * * C * C * * *
* * C * C * * R * G * A * M * N * * * + * * * N * M * A * G * R * * * C * C * * *
* C * * P * R * I * * * + * * * I * R * P * * C * *
```

Autor: Murtic Nirmel
www.etf.ba

```
ime=U1;
prezime=U2;
// Poziv funkcije „adresa()“ i slanje pokazivaca „ime“ i „prezime“ u funkciju
prezime=adresa(ime, prezime);
// Ispis Email adrese
printf("\nGenerirana korisnikova Email adresa: %s", c);
// Kraj programa
getch();
return 0;
}
```

```
*** C * C * * O * M * G * C * C * G * M * O * * C * C * * *
* * C * C * * R * G * A * M * N * * * + * * * N * M * A * G * R * * * C * C * *
* * C * * P * * R * * I * * * + * * * I * * R * * P * * * C * *
```

Autor: Murtic Nirmel
www.etf.ba

30. Napisati funkciju koja kao argument uzima znakovno polje **jmbg** (jedinostveni matični broj građanina) a kao izlaz vraća strukturnu varijablu koja predstavlja datum rođenja dotične osobe. Prototip funkcije je: **struct datum fdatum(char *jmbg)** gdje **struct datum** predstavlja strukturu definiranu kao:

```
struct datum
{
int dan;
int mjesec;
int godina;
};
```

Na primjer, ako je strukturna varijala **drodj** definirana kao: **struct datum drodj;** te ako je ulaz u funkciju **jmbg="2405978234567"** poziv funkcije **drodj=fdatum(jmbg);** treba vratiti strukturnu varijablu koja predstavlja datum **24.05.1978.**, tj. funkcija vraća slijedeće podatke strukturne varijable tipa datum:

```
drodj.dan=24;
drodj.mjesec=5;
drodj.godina=1978;
```

Nakon toga napišite glavni program koji sa standardnog ulaza učitava **jmbg** kao znakovno polje a nakon poziva funkcije **fdatum()** ispisuje datum rođenja osobe sa dotičnim jedinstvenim matičnim brojem.

RJEŠENJE:

```
# include <stdio.h>

// Definisiranje strukture datum
struct datum {
int dan;
int mjesec;
int godina;
};

/* Funkcija koja vraća datum kreiran od znakovnog polja u koje je upisan
matični broj */

struct datum fdatum(char *jmbg){

// Definisiranje varijable drodj kao strukturne varijable od datum
struct datum drodj;

// Definisiranje pokazivaca koji pokazuje na strukturnu varijablu
struct datum *pd;
int i;
```

```
***C**O**M**G**C**C**G**M**O**C**C**
**C**R**G**A**M**N**++**N**M**A**G**R**C**C**
**C**P**R**I**++**I**R**P**C**
```

Autor: Murtic Nirmel
www.etf.ba

```
// Racunanje varijabli strukturne varijable
// Pretpostavka je da je korisnik ciji maticni broj unesete rodjen lxxx godine
drodj.dan=0; drodj.mjesecc=0; drodj.godina=1000;
// Racunanje jedinice dana
for (i=0;i<10;i++) if (*(jmbg==49+i) drodj.dan=10*(1+i);
// Racunanje desetice dana i sabiranje sa jedinicom
for (i=0;i<10;i++) if (*(jmbg+1)==49+i) drodj.dan+=(1+i);
// Racunanje jedinice mjeseca
for (i=0;i<10;i++) if (*(jmbg+2)==49+i) drodj.mjesecc=10*(1+i);
// Racunanje desetice mjeseca i sabiranje sa jedinicom
for (i=0;i<10;i++) if (*(jmbg+3)==49+i) drodj.mjesecc+=(1+i);
// Racunanje stotice godine i sabiranje sa 1000
for (i=0;i<10;i++) if (*(jmbg+4)==49+i) drodj.godina+=100*(1+i);
// Racunanje desetice godine i sabiranje sa stoticom
for (i=0;i<10;i++) if (*(jmbg+5)==49+i) drodj.godina+=10*(1+i);
// Racunanje jedinice godine i sabiranje sa deseticom
for (i=0;i<10;i++) if (*(jmbg+6)==49+i) drodj.godina+=(1+i);
// Postavljanje pokazivaca na prvi znak strukturne varijable „drodj“
pd=&drodj;
// Funkcija vraca objekat koji se nalazi na adresi na koju pokazuje „pd“
return *pd;
}

// Glavni dio programa
main () {
struct datum drodj;
char jmbg[40], *p;

// Pretpostavka je da je uneseni maticni broj odgovarajuci
printf("Unesi maticni broj: ");
scanf("%s", &jmbg);

// Poziv funkcije fdatum i pridruzivanje rezultata strukturnoj varijabli drodj
drodj=fdatum(jmbg);

// Ispis rjesenja
printf("\nDatum rodjenja: %d.%d.%d", drodj.dan, drodj.mjesecc, drodj.godina);

// Kraj programa
getch();
return 0;
}
```

```
***C**C**O**M**G**C**C**G**M**O**C**C**
**C**C**R**G**A**M**N**+**+**N**M**A**G**R**C**C**
**C**P**R**I**+**+**I**R**P**C
```

Autor: Murtic Nirmel
www.etf.ba

31. Napisati program koji broji rijeci koje imaju **n** znakova u tekstualnoj datoteci cije ime se unosi sa tastature. Na primjer, ako u direktoriju(folderu) **T11** na disku **C:** postoji datoteka **pismo.txt** sa slijedecim sadržajem:

Lijep pozdrav!

Sretnu i uspjesnu Novu godinu zeli Vam kompanija BHKOM.

Isto tako se nadamo da ce se dosadasnja uspjesna saradnja nastaviti i u novoj 2006. godini.

Direktor: IVIC IVICA

program treba nakon unosa imena sa tastature i broja **n**:

Unesite broj znakova: 5

Unesite ime datoteke: C:\T11\pismo.txt

dati slijedeci izlaz:

Datoteka pismo.txt sadrzi 4 rijeci sa 5 znakova.

Napomene:

Rijeci sa pet znakova za gore navedenu datoteku su:

{**Lijep, BHKOM, novoj, IVICA**}.

Prilikom brojanja znakova ne uzimaju se u obzir znakovi interpunkcije: tacka (.), zarez (,), uskliknik (!) i sl.

Prilikom rješavanja zadatka obavezno napravite testnu datoteku **pismo.txt** i pohranite je u direktorij **T11**, te tu datoteku koristite za testiranje vašeg rješenja.

RJEŠENJE:

```
# include <stdio.h>
main() {
// Definisanje varijabli
FILE *datoteka;
char dat[60], znak;
int znakovi, br=0, rez=0;
// Unos putanje do datoteke
printf("Unesi putanju do datoteke: ");
scanf("%s", &dat);
// Ispituje se da li datoteka postoji, ako postoji, ide se dalje u program u
// suprotnom se ispisuje error poruka i izlazi iz programa
if ( (datoteka = fopen (dat, "r")) == NULL) {
printf("Greska kod otvaranja datoteke %s !\n", dat);
getch();
exit(1); }
// Unos broja znakova
printf("Unesi broj znakova: ");
scanf("%d", &znakovi);

/* While petlja vrti se i uzima se znak po znak iz datoteke sve dok ne dodje
se do kraja datoteke i kada stignemo na kraj izlazi se iz programa. Ukoliko je
uneseni karakter razlicit od znakova ? ! . \n , \t i blanko znaka onda se
varijabla br uveca za jedan, u suprotnom se provjeri da li je broj br jednak
broju znakova koji smo unjeli, i ako jeste varijabla rez se uveca za jedan, i
onda se br postavi na 0. Kada stignemo do kraja datoteke izlazi se iz
petlje.*/
```

```
*** C * C * * O * M * G * C * C * G * M * O * * C * C * * *
* * C * C * * R * G * A * M * N * * * + * + * * N * M * A * G * R * * * C * C * *
* C * * P * R * I * * * + * * * I * R * P * * C *
```

Autor: Murtic Nirmel
www.etf.ba

```
while((znak=fgetc(datoteka))!=EOF){
if ((znak=='?')||(znak=='!')||(znak==' ')
||(znak=='.')||(znak=='\n')||(znak==',')||(znak=='\t')) {
if (br==znakovi) rez++; br=0;} else br++;
}
/* Ukoliko zadnja rijec se sastoji od trazelog broja znakova, ona nece biti
uracunata jer smo izašli iz petlje, pa je potrebno dodati ovaj uslov iza
petlje da bi smo imali 100% tacan program*/
if (br==znakovi) rez++;
//Zatvaranje datoteke
fclose(datoteka);
// Izlaz iz programa
printf("Datoteka %s sadrzi %d rijeci sa %d znakova.", dat, rez, znakovi);
getch();
return 0;
}
```

```
***C*C*O*M*G*C*C*G*M*O**C**
**C**R*G*A*M*N**++**N*M*A*G*R**C**
**C**P*R*I**+**I*R*P**C*
```

Autor: Murtic Nirmel
www.etf.ba

32. Napravite tekstualnu datoteku **"ORPodaci.txt"** (pomocu programa Notepad ili slicnog) slijedeceg formata:

```
[prezime ] . . . . . [ime] . . . . . [brbod1] . . . . . [brbod2]
```

gdje :

- kolona [prezime] zauzima 15 mjesta;
- kolona [ime] zauzima 10 mjesta;
- kolona [brbod1] zauzima 4 mjesta i predstavlja broj bodova na prvom parcijalnom ispitu
- kolona [brbod2] zauzima 4 mjesta i predstavlja broj bodova na drugom parcijalnom ispitu
- U datoteku **"ORPodaci.txt"** unesite podatke za 15 studenata.
- Spasite datoteku **"ORPodaci.txt"** u direktorij C:\T12.
- **VAŽNO:** Prilikom kreiranja datoteke **"ORPodaci.txt"** budite precizni u zauzimanju mjesta za pojedine kolone.

Zatim, napisati program koji ucitava sadržaj tekstualne datoteke **"ORPodaci.txt"** i na temelju te datoteke kreira novu datoteku pod imenom **"Polozili.txt"** u kojoj se treba nalaziti sortiran (od veceg prema manjem) spisak onih studenata koji su položili oba parcijalna ispita (to su oni studenti koji imaju na oba parcijalna 10 ili više bodova).

Datoteka **"Polozili.txt"** je slijedeceg formata:

```
[prezime ] . . . . . [ime] . . . . . [ukupno]
```

gdje :

- kolona [prezime] zauzima 15 mjesta;
- kolona [ime] zauzima 10 mjesta;
- kolona [ukupno] zauzima 4 mjesta i predstavlja ukupan broj bodova sa oba parcijalna ispita.

RJEŠENJE:

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

/* Definisanje strukture forma koja se sastoji iz varijabli prezime i ime tipa
char i prvi i drugi tipa integer */
struct forma {
char prezime[15];
char ime[10];
int prvi;
int drugi;
};

main(){

/* Definisanje varijabli:
Neo - pokazivac na fajl koji se cita
Trinity - pokazivac na fajl u koji se pise
```

```
*** C * C * * O * M * G * C * C * G * M * O * * C * C * * *
* * C C * * R G A M N * * + * * N M A G R * * C C * *
* C * * P * R * I * * * + * * * I * R * P * * C *
```

Autor: Murtic Nirmel
www.etf.ba

```
podaci[15] - strukturalni niz podaci u koji ce se upisati svi podaci koji se
ucitaju iz fajla, maksimalno 15...
Love - pomocna strukturalna varijabla
i,j,k - potrebni projaci za sortiranje, i jos ponesta :) */
FILE *Neo, *Trinity;
struct forma podaci[15];
struct forma Love;
int i=0, j=0, k=0;

/* Otvaranje datoteke ORPodaci.txt i pridruzivanje pokazivacu Neo, ako ne
postoji datoteka izlazi se iz programa */
if ((Neo=fopen("C:\\T12\\ORPodaci.txt", "r"))==NULL)
{
printf("Greska pri otvaranju datoteke");
exit(1);
}

/* Datoteka polozili.txt se kreira i pridruzuje pokazivacu Trinity */
Trinity = fopen ("c:\\T12\\Polozili.txt", "w");

/* Citanje podataka iz datoteke i upisivanje istih u strukturu podaci.
Varijabla i se vraca kao broj koji oznacava broj studenata koji su polozili
obe parcijale. podaci[i].prvi se automatski sabira sa podaci[i].drugi i ukupna
vrijednost postaje ustvari podaci[i].prvi!*/
while(fscanf(Neo, "%15s%10s%4d%4d", &podaci[i].prezime, &podaci[i].ime, &podaci[i]
.prvi, &podaci[i].drugi) != EOF )
{
if ((podaci[i].prvi>=10)&&(podaci[i].drugi>=10))
podaci[i].prvi+=podaci[i].drugi; else i--;
i++;
}

// Proces sortiranja... objasnjen u jednom od zadataka ranije...
for(j=0; j<i-1; j++)
for(k=j+1; k<i; k++)
if (podaci[k].prvi>podaci[j].prvi)
{
Love=podaci[j];
podaci[j]=podaci[k];
podaci[k]=Love;
}

// Upis podataka u datoteku na koju pokazuje pokazivac Trinity (Polozili.txt)
for (j=0; j<i; j++)
fprintf(Trinity, "%-15s%-10s%-4d\n",
podaci[j].prezime, podaci[j].ime, podaci[j].prvi);

// Zatvaranje datoteka...
fclose(Neo);
fclose(Trinity);

// Kraj programa...
getch();
return 0;
}
```

Email: matrix.bih@gmail.com ; MSN: nirmel_m@hotmail.com

```
* * * C * C * * O * M * G * C * C * G * M * O * * C * C * * *
* * C C * * R G A M N * * + * * N M A G R * * C C * *
* C * * P * R * I * * * + * * * I * R * P * * C *
```

Autor: Murtic Nirmel
www.etf.ba

33. Napisati funkciju *okreni()* koja kao argument uzima neki string i okreće sadržaj tog stringa naopako. Prototip funkcije *okreni()* treba biti:

```
void okreni(char* p)
```

Zatim napravite glavni program koji sa standardnog ulaza učitava neki string, te ga ispisuje naopako.

Primjer:

Unesite string: **Sarajevo**

Naopako: **ovejaraS**

RJEŠENJE:

```
#include <stdio.h>
// Funkcija okreni
void okreni(char *p){
// Definisane pomocnih varijabli
char pom;
int i=0, j=0, br=0;
// Brojanje slova rijeci, pointer „odlazi“ na kraj rijeci.
while (*p!='\0') {br++; p++;}
/* Pointer je „otisao“ na kraj rijeci, a nama treba u nastavku da je na
pocetku, pa cemo ga vratiti na pocetak tako sto ga pomaknemo za isti broj
mjesta nazad, za koji smo ga while petljom pomjerali naprijed! */
p=p-br;
// Glavni dio programa, ALGORITAM.
/* Recimo da smo unjeli rijec LOVE. Rijec koju trebamo dobiti kao rezultat je
EVOL. Znaci potrebno je zamjeniti zadnje slovo sa prvim, i drugo sa
predzadnjim. Brojac „i“ nam predstavlja znak do kojeg smo dosli sprijeda, a
brojac „j“ straga. Na pocetku ce biti i=0, j=3 (jer je br=3, nemojte se
zbuniti sto je 3 a ne 4, tu su ipak cetiri clana 0,1,2,3). Znaci nakon sto se
for petlja jednom izvrši „i“ se uveca za 1 i postaje 1, a j se umanja za 1 i
postane 2. Znaci u prvom slucaju pomocu pomocne varijable „pom“ mijenjamo
znakove na 0 i 3 poziciji, drugi put ce se zamjeniti znakovi na 1 i 2
poziciji. I dobili smo trazenu rijec. Mozda zbunjuje uslov i<br/2; ali je
neophodan, broj permutacija je jednak polovini duzine rijeci, nacrtajte sebi
neki primjer i zakljucite to :). Ukoliko se uzme rijec sa neparnim brojem
slova onda se srednji clan ne pomjera nigdje, ostaje, sto je i logicko. Dok se
clanovi oko njega izmjenjaju simetricno. Funkcija nevraca nikakvu vrijednost,
ali promjene ostaju zapamcene zbog pristupa vrijednostima preko pointera! */
for (i=0, j=br-1; i<br/2; i++, j--){
pom=*(p+i);
*(p+i)=*(p+j);
*(p+j)=pom;
}
}
// Glavni program
main(){
// Definisane varijabli i pointera
char NIZ[100];
char *t;
// Unos neke rijeci i zapisivanje iste u NIZ
printf("Unesi neku rijec: ");
scanf("%s", &NIZ);
```

```
*** C * C * * O * M * G * C * C * G * M * O * * C * C * * *
* * C * C * * R * G * A * M * N * * * + * * * N * M * A * G * R * * * C * C * * *
* C * * P * R * I * * * + * * * I * R * P * * C *
```

Autor: Murtic Nirmel
www.etf.ba

```
// Postavljanje pokazivaca „t“ na „NIZ“
t=NIZ;
// Pozivanje funkcije i slanje pokazivaca „t“ u funkciju
okreni(t);
/* Ispis stringa NIZ preuredjenog funkcijom (printa se „t“, a „t“ nam pokazuje
na „NIZ“). */
printf("%s", t);
// Kraj programa
getch();
return 0;
}
```

```
***C*C*O*M*G*C*C*G*M*O**C*C**
**C**R*G*A*M*N**++**N*M*A*G*R**C*C**
*C**P*R*I**++**I*R*P**C*
```

Autor: Murtic Nirmel
www.etf.ba

34. Napisati funkciju **ubaci5()** koja kao argument uzima neki prirodni broj **n** i dva broja **a** i **b**, a zatim vraća se broj koji bi nastao zamjenom cifri na mjestu **a** i **b** cifrom 5 (gledano s desne strane). Funkcija ne treba ništa ispisivati. Funkcija **ubaci5()** treba imati prototip:

```
int ubaci5(int n,int a,int b);
```

Pretpostaviti da su **a** i **b** valjano unesene vrijednosti koje imaju smisla. Primjeri ulaznih argumenata i vraćenih vrijednosti:

Primjer1: za ulazne argumente **n=123321**, **a=2**, **b=3** funkcija vraća broj **123551**.

Primjer 2: za ulazne argumente **n=23455**, **a=3**, **b=5** funkcija vraća broj **53555**.

RJEŠENJE:

```
#include <stdio.h>
// Funkcija ubaci5();
int ubaci5(int n,int a,int b) {
// Definisane varijabli
int c=0,t=1,o,i;
/* Brojac „i“ kreće od 1(prvi znak straga) i ide dok broj „n“ koji smo primili
u funkciju ne postane 0. Svaki put kada for petlja se izvršava ponovo, uzima
se ostatak dijeljenja sa 10, odnosno uzima se zadnja cifra broja „n“, i
pridružuje varijabli „o“. Zatim se broj „n“ podijeli sa 10, čime mu se odbije
zadnja cifra. Suma „c“ se uvećava za „t*o“ gdje t se mijenja kao
1,10,100,1000. Kada i bude jednako „a“ ili „b“ onda se umjesto „t*o“ doda
„t*5“ i time se na „a-to/b-to“ mjesto doda broj 5 umjesto broja koji je bio na
toj poziciji. Naredba „continue;“ će vratiti na početak for petlje kako se
nebi izvršile i linije ispod, jer bi time dobili ubacivanje brojeva, a ne
zamjenu!!! Savjet: Probajte napisati recimo petocifren broj, i skontajte malo
na papiru prvo kako biste te zamjene uradili, skontajte ulogu varijabli „o“ i
„t“!!! */
for (i=1;n>0;i++){
o=n%10;
n=n/10;
if (i==a) {c=c+t*5; t=t*10; continue;}
if (i==b) {c=c+t*5; t=t*10; continue;}
c=c+t*o;
t=t*10;}
return c;}
main(){
// Definisane varijabli a,b,n
int n, a, b;
// Unos brojeva a,b,n
printf("Unesi prirodan broj n:"); scanf("%d", &n);
printf("Unesi brojeve a,b:"); scanf("%d,%d", &a, &b);
/* Poziv funkcije, slanje n,a,b u funkciju, a rezultat se
pridružuje varijabli n, koja se printa u idućoj liniji. */
n=ubaci5(n,a,b);
printf("%d", n);
// Kraj programa.
getch();
return 0; }
```

```
***C**C**O**M**G**C**C**G**M**O**C**C**
**C**C**R**G**A**M**N**++**N**M**A**G**R**C**C**
**C**P**R**I**++**I**R**P**C
```

Autor: Murtic Nirmel
www.etf.ba

35. Napisati program koji učitava podatke iz ulazne formatirane tekstualne datoteke „**gradovi.txt**“ i kreira izlaznu datoteku „**prosjeci.txt**“ koja sadrži spisak gradova s prosječnom temperaturom za prva tri mjeseca. Datoteka „**gradovi.txt**“ je sljedećeg formata:

```
[grad] [JAN] [FEB] [MAR]
```

gdje je:

[grad] kolona u kojoj su zapisani nazivi gradova i koja zauzima 15 mjesta;

[JAN], [FEB], [MAR] kolona u kojoj su zapisane cjelobrojne prosječne temperature za mjesec januar, februar i mart, respektivno. Ove kolone zauzimaju po 10 mjesta.

Datoteka „**prosjeci.txt**“ je sljedećeg formata:

```
[grad] [PRTEMP]
```

gdje je:

[grad] kolona u kojoj su zapisani nazivi gradova i koja zauzima 15 mjesta;

[PRTEMP] kolona u kojoj su zapisane prosječne temperature koristeći dva decimalna mjesta i maksimalno dva mjesta prije decimalne tačke.

Primjer sadržaja datoteke „**gradovi.txt**“:

```
TRAVNIK 0 1 2
ZENICA 2 3 4
TUZLA 2 5 5
SARAJEVO 0 3 1
```

Primjer sadržaja datoteke „**prosjeci.txt**“:

```
TRAVNIK 1.00
ZENICA 3.00
TUZLA 4.00
SARAJEVO 1.33
```

RJEŠENJE:

```
#include <stdio.h>
main(){
// Definisane varijabli
FILE *Neo, *Trinity;
char grad[15];
float JAN, FEB, MAR, PRO;
/* Provjera da li postoji datoteka gradovi.txt i ako ne postoji ispis greske,
u suprotnom se nastavlja program */
if ((Neo=fopen("C:\\gradovi.txt","r"))==NULL)
{
printf("Greska pri otvaranju datoteke");
return 0;
}
// Otvaranje datoteke prosjeci.txt
Trinity=fopen("C:\\prosjeci.txt","w");
/* Citanje podataka iz datoteke gradovi.txt izracunavanje prosjecne
temperature PRO i printanje u datoteku prosjeci.txt */
```

```
* * * C * C * * O * M * G * C * C * G * M * O * * C * C * * *  
* * C C * * R G A M N * * + * * N M A G R * * C C * *  
* C * * P * R * I * * * + * * * I * R * P * * C *
```

Autor: Murtic Nirmel
www.etf.ba

```
while(fscanf(Neo,"%15s%10f%10f%10f", &grad, &JAN, &FEB, &MAR) != EOF ) {  
PRO=(JAN+FEB+MAR)/3;  
fprintf(Trinity,"%-15s%2.2f\n", grad, PRO);  
}  
// Kraj programa  
close(Neo);  
close(Trinity);  
getch();  
return 0;  
}
```

```
***C**C**O**M**G**C**C**G**M**O**C**C**  
* **C**C**R**G**A**M**N**+**+**N**M**A**G**R**C**C**  
* **C** **P**R**I**+**+**I**R**P**C**
```

Autor: Murtic Nirmel
www.etf.ba

36. Napisati funkciju **izbacip()** koja kao argument uzima neki prirodni broj **n**, a zatim vraća broj koji bi nastao izbacivanjem parnih cifri iz tog broja. Funkcija ne treba ništa ispisivati. Funkcija **izbacip()** treba imati prototip:

```
int izbacip(int n);
```

Na primjer, za argument **123420** funkcija treba vratiti broj **13**.

RJEŠENJE:

```
// Funkcija izbacip() prima neki integer broj  
int izbacip(int n){  
/* Definisane varijabli, „t“ je tezisnica koja se svaki put kada dodajemo  
neku cifru pomnozi sa 10, „o“ je ostatak, „i“ je brojac, a „c“ je broj koji ce  
funkcija vratiti kao rezultat */  
int c=0,t=1,o,i;  
/* „i“ se uvecava za 1 i for petlja se vrti sve dok broj „n“ bude jednak 0 */  
for (i=1;n>0;i++){  
/* Uzima se ostatak dijeljenja sa 10, odnosno zadnja cifra broja */  
o=n%10;  
// „Odbije“ se zadnja cifra broja „n“ dijeljenjem sa 10  
n=n/10;  
/* Ako je broj paran, vraća se na pocetak petlje i time se izbacuju parne  
cifre */  
if (o%2==0) continue;  
// Dodavanje cifre na broj „c“  
c=c+t*o;  
// Mnozenje tezisnice sa 10  
t=t*10;  
}  
// Vracanje varijable c  
return c;  
}
```

```
***C**C**O**M**G**C**C**G**M**O**C**C**  
**C**C**R**G**A**M**N**+**+**N**M**A**G**R**C**C**  
*C**P**R**I**+**+**I**R**P**C*
```

Autor: Murtic Nirmel
www.etf.ba

37. Napisati funkciju **ispis_preko_a()** koja kao argument uzima neko cjelobrojno polje i dva prirodna broja **a** i **d**, a zatim ispisuje sve elemente polja koji su veci od broja **a**. Broj **d** predstavlja broj elemenata polja. Funkcija treba imati slijedeci prototip:

```
void ispis_preko_a(int *polje, int a, int d)
```

Na primjer, za ulazne argumente

```
int polje[]={4,5,7,20,8,10,2,-2}, a=6, d=8
```

trebaju se ispisati brojevi: 7, 20, 8, 10

RJEŠENJE:

```
// Funkcija prima pokazivac na polje, i dva cijela broja „a“, „d“  
void ispis_preko_a(int *polje, int a, int d)  
{  
    // Definisavanje brojac „i“  
    int i;  
    /* For petlja se „vrti“ sve dok brojac i koji krece od 0 ne bude veci od broja  
    „d“ koji je funkcija primila. */  
    for (i=0; i<d; i++){  
        /* Ako je broj „a“ koji je primljen u funkciju manji od broja u polju do kojeg  
        smo dosli on se ispisuje na ekran.*/  
        if (a<polje[i]) printf("%d, ", polje[i]);  
    }  
}
```

```
***C**C**O**M**G**C**C**G**M**O**C**C**
**C**C**R**G**A**M**N**++**N**M**A**G**R**C**C**
**C**C**P**R**I**++**I**R**P**C**
```

Autor: Murtic Nirmel
www.etf.ba

38. Napisati funkciju **zamijeni()** koja ce u jednodimenzionalnom polju **A** velicine **d** zamijeniti **i**-ti element sa minimalnim elementom u polju **A**. Elementi polja **A** su vec definirani u glavnom programu. Funkcija treba imati prototip:

```
void zamijeni(int *A, int i, int d)
```

Na primjer, ako su elementi polja A:

```
int A[]={3, 7, 10, 20, 18, 4,10, 2, 20}, a argumenti i i d imaju slijedece vrijednosti i=3, d=9 funkcija zamijeni() promjeni polje A u polje sa slijedecim elementima: 3, 7, 2, 20, 18, 4, 10, 2, 20 (na trece mjesto se stavlja minimalni element iz polja a to je 2)
```

RJEŠENJE:

```
// U funkciju se prima niz brojeva „A“, broj „i“, broj „d“.
void zamijeni(int *A, int i, int d)
{
// Definisanje brojaca „j“ i varijable „min“
int j,min;
// Postavljanje vrijednosti „min“ na prvi clan niza
min=A[0];
/* Trazenje najmanjeg elementa u nizu. Pretpostavili smo da je to prvi broj, i
sada trebamo ispitati da li postoji neki broj krenuvsi od drugog (j=1 umjesto
j=0 u for petlji) i ako postoji neki manji postavi se da je taj broj „min“. */
for (j=1; j<d; j++) if (min>A[j]) min=A[j];
/* Postavljanje na i-tu poziciju vrijednosti „min“ (i=1 ide jer nam je i-ti
clan ustvari broj za jedan manje, jer nam je prvi clan nulti clan! */
A[i-1]=min;
// Ispis niza
for (j=0; j<d; j++) printf("%d ", A[j]);
}
```

```
* * * C * C * * O * M * G * C * C * G * M * O * * C * C * * *
* * C C * * R G A M N * * + * * N M A G R * * C C * *
* C * * P * R * I * * * + * * * I * R * P * * C *
```

Autor: Murtic Nirmel
www.etf.ba

39. Data je tekstualna datoteka `c:\pismo.txt`.

Napisati funkcije za:

- ispis datoteke na standardni izlaz zadržavajući strukturu po redovima;
- prepis datoteke u novu mijenjajući mala i velika slova;
- ispisati datoteku u obrnutom redosljedu na izlaz (prva postaje zadnja, druga postaje prethodna riječ...)
- ispitati da li je neka riječ palindrom (isto značenje kada se čita i u obrnutom smjeru)

RJEŠENJE:

```
# include <stdio.h>
// Ispis datoteke na standardni izlaz zadržavajući strukturu po redovima...
void ispis(FILE *datoteka){
char znak;
printf("\nSadržaj datoteke je: \n\n");
while(fscanf(datoteka,"%c", &znak)!=EOF){
printf("%c", znak);
}
}

/* Ispitati da li je neka riječ palindrom (isto značenje kada se čita i u
obrnutom smjeru) */
void palindrom(FILE *datoteka){
char znak[100];
char *p, *t;
int i, br=0, ispit=0;
printf("\n\nProvjera palindroma: \n");
while(fscanf(datoteka,"%s", &znak)!=EOF){
p=znak;
t=znak;
br=0;
while(*p!='\0') {p++; br++;} p--;
ispit=1;
for(i=0;i<(br/2);i++) if (*(p-i)!=*(t+i)) {ispit=0; break;}
if (ispit==1) {printf("\nPalindrom je: %s", znak); ispit=0;}
}
}

// Prepis datoteke u novu mijenjajući mala i velika slova
void ispisunovu(FILE *datoteka){
FILE *datoteka1;
char znak;
datoteka1=fopen("c:\\pismo1.txt","w");
while(fscanf(datoteka,"%c", &znak)!=EOF){
if ((znak>64)&&(znak<91)) znak=znak+32; else
if ((znak>96)&&(znak<122)) znak=znak-32;
fprintf(datoteka1, "%c", znak);
}
fclose(datoteka1);
}
```

```
***C*C*O*M*G*C*C*G*M*O*C*C***
**CC**RGAMN**++**NMAGR**CC**
*C**P*R*I***+***I*R*P**C*
```

Autor: Murtic Nirmel
www.etf.ba

```
/* Ispisati datoteku u obrnutom redosljedu na izlaz (prva postaje zadnja,
druga postaje pretposljednja rijec...) */
void ispisobrnuto(FILE *datoteka){
char znak, matrix[1000][1000], pok[1000];
int i=0, j=0, br=0, k=0, niz[100];
printf("\n\nSadrzaj datoteke je (naopako): \n");
while(fscanf(datoteka,"%c", &znak)!=EOF){
if (znak==' ') {br++; j++; i=0; continue;}
if (znak=='\n') {br++; niz[k]=j; j++; k++; i=0; continue;}
matrix[j][i]=znak;
i++;
}
niz[k]=j;
j=0;
for (i=br;i>=0;i--) {
if((i==niz[k-j])&&(k-j!=-1)) { printf("\n"); j++;}
printf("%s ", matrix[i]);
}
}

main() {
// Definisane pokazivaca na datoteku
FILE *datoteka;
// Otvaranje datoteke za citanje
datoteka=fopen("c:\\pismo.txt","r");
// Poziv funkcije za ispis datoteke na ekran...
ispis(datoteka);
fclose(datoteka);
// Otvaranje datoteke za citanje
datoteka=fopen("c:\\pismo.txt","r");
// Poziv funkcije za prepis datoteke u novu mijenjajuci mala i velika slova.
ispisunovu(datoteka);
fclose(datoteka);
// Otvaranje datoteke za citanje
datoteka=fopen("c:\\pismo.txt","r");
/* Poziv funkcije za ispis datoteke u obrnutom redosljedu na izlaz (prva
postaje zadnja, druga postaje pretposljednja rijec...) */
ispisobrnuto(datoteka);
fclose(datoteka);
// Otvaranje datoteke za citanje
datoteka=fopen("c:\\pismo.txt","r");
/* Poziv funkcije za ispitivanje da li je neka rijec palindrom (isto znacenje
kada se cita i u obrnutom smjeru) */
palindrom(datoteka);
fclose(datoteka);
// Kraj programa...
getch();
return 0;
}
```

```
*** C * C * * O * M * G * C * C * G * M * O * * C * C * * *
* * C * C * * R G A M N * * + * * N M A G R * * C * C * *
* C * * P * R * I * * * + * * * I * R * P * * C *
```

Autor: Murtic Nirmel
www.etf.ba

40. Napisati program koji cita više linija teksta sa ulaza i štampa koliko rijeci je dužine 2 3 4 i više slova. Štampu realizovati pomocu funkcije STAM_REZ u datoteku ili na ekranu. Iz datoteke rezultata dati izvještaj o ukupnom broju rijeci i prosjecnoj dužini rijeci.

RJEŠENJE:

```
#include <stdio.h>

// Funkcija koja kreira datoteku sa rezultatima
void STAM_REZ(char *tekst, int br){

// Definisane pokazivaca na fajl
FILE *rezultati;

// Definisane varijabli
int i,b=0,dva=0,tri=0,cetiri=0,vise=0, brojslova=0, prosjek=0;

// Otvaranje datoteke u koju cemo pisati podatke
rezultati=fopen("c:\\rezultati.txt","w");

// For petlja vrti se "br" puta...
for (i=0;i<br;i++){

/* Ukoliko znak do kojeg smo dosli nije „?“ „!“ „.“ „\n“ , „\t“ ili „blanko“
brojac „b“ se uvecava za 1, brojac „b“ nam pokazuje koliko slova je u rijeci
izbrojano. */
if ((tekst[i]=='?')||(tekst[i]=='!')||(tekst[i]==' ' )
|| (tekst[i]=='.')||(tekst[i]=='\n')||(tekst[i]==',')||(tekst[i]=='\t')) {

/* Nakon sto naidjemo na neki od ovih znakova u uslovu ulazimo u petlju... i
imamo... ako je b>1 broj slova ce se sabrati sa brojem slova rijeci koja se
trenutno obradjuje, ako je b=2 onda se varijabla dva koja broji rijeci sa 2
slova uveca za 1. Ako je tri onda se varijabla tri uveca za 1... ako je preko
4 onda se varijabla vise uveca za 1... primjetite da se brojac vraca na 0
izlaskom iz uslova, nebitno da li je bilo b=0 ili 1 ili 2 ili ... */
if (b>1) brojslova=brojslova+b;
if (b==2) {dva++; b=0;} else
if (b==3) {tri++; b=0;} else
if (b==4) {cetiri++; b=0;} else
if (b>4) {vise++; b=0;} else b=0;
} else b++;
}

// Upis podataka u datoteku...
fprintf(rezultati,"Rijeci sa 2 slova: %d\n", dva);
fprintf(rezultati,"Rijeci sa 3 slova: %d\n", tri);
fprintf(rezultati,"Rijeci sa 4 slova: %d\n", cetiri);
fprintf(rezultati,"Rijeci sa preko 4 slova: %d\n", vise);
prosjek=brojslova/(dva+tri+cetiri+vise);
fprintf(rezultati,"PROSJEK: %d\n", prosjek);
```

```
***C**C**O**M**G**C**C**G**M**O**C**C**  
**C**C**R**G**A**M**N**++**N**M**A**G**R**C**C**  
*C**P**R**I**++**I**R**P**C*
```

Autor: Murtic Nirmel
www.etf.ba

```
/* Zatvaranje datoteke i izlaz iz funkcije... „void“ funkcija ne vraća  
nista!!! Izlaskom je sve što se u njoj desavalo zaboravljeno, ali zato imamo  
sve rezultate pohranjene u datoteci iz koje ćemo po povratku u program citati  
podatke.*/  
fclose(rezultati);  
}  
  
// Glavni program  
main(){  
  
// Definisane varijabli  
int br=0, i=0, p=0, dva, tri, cetiri, vise, prosjek;  
char tekst[1000];  
  
// Definisane pokazivaca na fajl  
FILE *izvjestaj;  
  
/* Unos teksta (koristio sam while petlju koja upisuje znak po znak, a ne %s  
koja upisuje cijeli string odjednom, a zašto? pa jednostavno još uvijek neznam  
drugi način da izbjegnem situaciju kad upisete prvi razmak, a program to  
shvati kao kraj stringa i računa samo da je string prva riječ, a ostalo kao da  
niste unjeli. Ovako se učitava sve što unesete sa tastature, što nam i treba. U  
zadatku nije naznačeno šta će biti prekid unosa, pa sam ja stavio takticu :) )*/  
printf("Unesi tekst[i], (Tacka za kraj unosa!):");  
while (tekst[i-1]!='.') { scanf("%c", &tekst[i]); i++; br++;}  
  
/* Poziv funkcije u koju saljemo tekst koji smo unijeli kao i brojac „br“ koji  
je prebrojao broj unesenih karaktera, mogli smo i bez njega, al eto ja nekako  
više volim for petlje od while petlji :) */  
STAM_REZ(tekst, br);  
  
// Otvaranje datoteke za citanje i pridruzivanje pokazivacu izvjestaj.  
izvjestaj=fopen("c:\\rezultati.txt","r");  
  
// Citanje podataka iz datoteke... formati isti kao pri upisu :)  
fscanf(izvjestaj,"Rijeci sa 2 slova: %d\n", &dva);  
fscanf(izvjestaj,"Rijeci sa 3 slova: %d\n", &tri);  
fscanf(izvjestaj,"Rijeci sa 4 slova: %d\n", &cetiri);  
fscanf(izvjestaj,"Rijeci sa preko 4 slova: %d\n", &vise);  
fscanf(izvjestaj,"PROSJEK: %d", &prosjek);  
  
// Ispis izvjestaja na ekranu.  
printf("\n\nIZVJESTAJ: ");  
printf("\n\nUkupno rijeci: %d", dva+tri+cetiri+vise);  
printf("\n\nProsjecna duzina rijeci: %d", prosjek);  
  
// Zatvaranje datoteke i kraj programa  
fclose(izvjestaj);  
getch();  
return 0;  
}
```

```
***C*C*O*M*G*C*C*G*M*O*C*C**
**C**R*G*A*M*N**++**N*M*A*G*R**C*C**
* C * * P * R * I * * * + * * * I * R * P * * C *
```

Autor: Murtic Nirmel
www.etf.ba

41. Program treba unjeti 2 broja i ispisati NZD... najveći zajednički djelilac...

RJEŠENJE:

```
#include <stdio.h>
main() {
// Definicija varijabli
int i,a,b,min,nzd=1;
// Unos dva prirodna broja
printf("Unesi dva broja: ");
scanf("%d,%d", &a, &b);
// Pridružuje se varijabli min manji od unesena 2 broja
if (a<=b) min=a; else min=b;
// I se stavlja da je jednako min, i umanjuje se sve do 1... prvi broj
// sa kojim uspiju podijeliti se oba broja je i NZD... izlazi se iz petlje kad
// se oba broja uspiju bodijeliti...
for (i=min;i>0;i--){
if (((a%i)==0)&&(b%i)==0) {nzd=i; break;}
}
// Ispis NZD-a
printf("\nNajveći zajednički sadržilac je: %d" , nzd);
// Kraj programa
getch();
return 0;
}
```

```
* * * C * C * * O * M * G * C * C * G * M * O * * C * C * * *
* * C C * * R G A M N * * + * * N M A G R * * C C * *
* C * * P * R * I * * * + * * * I * R * P * * C *
```

Autor: Murtic Nirmel
www.etf.ba

42. Ispis sljedeceg znaka na ekranu...

```
+++++--+++++
++++---++++
+++-----+++
++-----++
+-----+
++-----++
+++-----+++
++++---++++
+++++--+++++
```

```
#include <stdio.h>
main() {
int k,i,j,a=11; // a predstavlja dimenzije znaka
for (k=0;k<a;k++) printf("+");
printf("-");
for (k=0;k<a;k++) printf("+");
printf("\n");
for (i=1;i<a;i++){
for (k=0;k<a-i;k++) printf("+");
for (k=a;k>a-i;k--) printf("-");
printf("-");
for (k=a;k>a-i;k--) printf("-");
for (k=0;k<a-i;k++) printf("+");
printf("\n");
}
for (i=2;i<a;i++){
for (k=a;k>a-i;k--) printf("+");
for (k=0;k<a-i;k++) printf("-");
printf("-");
for (k=0;k<a-i;k++) printf("-");
for (k=a;k>a-i;k--) printf("+");
printf("\n");
}
for (k=0;k<a;k++) printf("+");
printf("-");
for (k=0;k<a;k++) printf("+");
printf("\n");
// Svaki pokusaj pojasnjenja ovog zadatka je suvisan... ovo cete ili skontat
ili necete :) jednostavna igra sa for petljama... pokusajte uraditi preko
funkcija :)
// Kraj programa
getch();
return 0;
}
```

```
* * * C * C * * O * M * G * C * C * G * M * O * * C * C * * *
* * C C * * R G A M N * * + * * N M A G R * * C C * *
* C * * P * R * I * * * + * * * I * R * P * * C *
```

Autor: Murtic Nirmel
www.etf.ba

43. Unesite broj članova niza cijelih brojeva, a zatim i članove niza a zatim ih sortirajte od najvećeg ka najmanjem i ispisite sortirano na ekran.

RJEŠENJE:

```
# include <stdio.h>
main(){
// Definisane varijabli
int n,i,j,niz[100],pom;
// Unos broja članova niza
printf("\n Unesite broj članova: "); scanf("%d", &n);
/* Unos niza */
for (i=0;i<n;i++){
    printf("\n Unesi član %d :", i+1);
    scanf("%d", &niz[i]); }
/* Znači imamo neki niz, recimo od 5 članova: [1,3,4,2,5] i trebamo ga
sortirati od većeg ka manjem broju... na kraju treba izgledati [5,4,3,2,1].
Pri korištenju ove metode, koristimo dvije for petlje, od kojih je jedna (po
„i“ brojacu, tu samo onako radi reda) :) dok druga petlja obavlja većinu
posla. Možete primjetiti da u uslovu stoji „i<(n-1)“ kao i „j<(n-1)“. Ako
pogledamo jedan niz od 5 članova i ako bi mijenjali brojeve tako da mijanjamo,
prvi i drugi član, zatim drugi i treći, pa 3 i 4, pa 4 i 5. Znači imamo 4
izmjene iako imamo 5 članova, i zbog toga u uslovu ide „n-1“ umjesto „n“!
Znači program radi tako da se vanjska for petlja izvrši „n-1“ puta a
unutrasnja se izvrši „(n-1)*(n-1)“ puta... u našem slučaju od 5 članova, znači
16 puta. Možda je ova metoda spora dosta jer se okreće 16 puta, što bi se
moglo smanjiti, ali zbog jednostavnosti izgleda, prvo sam uzeo ovu metodu.
Dalje imamo uslov u unutrasnjoj petlji „niz[j]<niz[j+1]“ koji bi u prijevodu
značio, ako je j-ti član manji od idućeg (j+1) onda im mijenja mjesta...
Graficki desavanja pri sortiranju naseg niza izgledaju ovako (uspravno
citajte).
1 - 3 3 3 3   4 4 4 4   4 4 4 4   5 5 5 5
3 - 1 4 4 4   3 3 3 3   3 5 5 5   4 4 4 4
4 - 4 1 2 2   2 2 5 5   5 3 3 3   3 3 3 3
2 - 2 2 1 5   5 5 2 2   2 2 2 2   2 2 2 2
5 - 5 5 5 1   1 1 1 1   1 1 1 1   1 1 1 1
I na kraju smo dobili sortirani niz... znači 16 puta se vrtela unutrasnja
petlja po 4 puta je kretala od prvog do četvrtog člana i vrsila izmjene po
uslovu u if petlji, i na kraju rezultat mora biti sortirani niz! Da smo htjeli
obrnuto da sortiramo od najmanjeg ka najvećem samo bi se promijenio znak < u
uslovu u if petlji u znak > ... :) */
for (i=0;i<(n-1);i++)
for (j=0;j<(n-1);j++)
if (niz[j]<niz[j+1]) {
    pom=niz[j+1];
    niz[j+1]=niz[j];
    niz[j]=pom;
}
// Ispis niza i kraj programa...
printf("\n Ispis niza: ");
for (i=0;i<n;i++) printf("[%d]", niz[i]);
getch();
return 0;
}
```

```
***C*C*O*M*G*C*C*G*M*O*C*C***
* * C C * * R G A M N * * + * * N M A G R * * C C * *
* C * * P * R * I * * + * * I * R * P * * C *
```

Autor: Murtic Nirmel
www.etf.ba

44. Zadatak br. 43. (prethodni) na drugi nacin...

RJEŠENJE:

```
# include <stdio.h>
main(){
// Definisane varijabli
int n,i,j,niz[100],pom;
// Unos broja clanova niza
printf("\n Unesite broj clanova: "); scanf("%d", &n);
/* Unos niza */
for (i=0;i<n;i++){
    printf("\n Unesi clan %d :", i+1);
    scanf("%d", &niz[i]); }

/* U prethodnom primjeru sam uradio zadatak metodom bubblesort, a ovaj sam
uradio koristeći metodu NALAZENJA NAJVECEG U NIZU/PODNIZU i stavljanje na
"vrh". Znaci samo se ove for petlje razlikuju u odnosu na prethodni zadatak :)
Imamo vanjsku petlju koja je identicna onoj iz prethodnog zadatka, i ona vrti
unutrasnju petlju n-1 puta. Sad, ako pogledate prethodni primjer primjetiti
cete na grafickom prikazu da nekoliko puta su se vrtile for petlje ali je niz
ostajao isti. Ovom metodom cemo izbjeći to, i uciniti da program radi jos
brze. Znaci svakim novim prolaskom kroz vanjsku petlju se PODNIZ unutar
unutrasnje petlje umanja za 1, i time dobijamo na vremenu. Ovo je malo teze
objasniti :) pa pogledajte graficki prikaz:

1 - 3 4 4 5   5 5 5   5 5   5
3 - 1 1 1 1   3 3 4   4 4   4
4 - 4 3 3 3   1 1 1   2 3   3
2 - 2 2 2 2   2 2 2   1 1   2
5 - 5 5 5 4   4 4 3   3 2   1

Vidite kako smo dosli do rezultata uz samo 11 permutacija, naspram onih 16 iz
prvog primjera. Znaci u prvom dijelu ovog grafickog prikaza, najden je najveći
broj niza i stavljen na vrh (broj 5). U drugom dijelu (primjetite da smo
smanjili broj okretanja sa 4 na 3 jer nas vise broj na vrhu ne interesuje, i
samo od preostalog niza od 4 clana trazimo najveći i stavljammo na vrh... i
tako do kraja. */

for (i=0;i<(n-1);i++)
for (j=i+1;j<n;j++)
if (niz[j]>niz[i])
{
pom=niz[i];
niz[i]=niz[j];
niz[j]=pom;
}

// Ispis niza i kraj programa...
printf("\n Ispis niza: ");
for (i=0;i<n;i++) printf("[%d]", niz[i]);

getch();
return 0;
}
```

```
* * * C * C * * O * M * G * C * C * G * M * O * * C * C * * *
* * C C * * R G A M N * * + * * N M A G R * * C C * *
* C * * P * R * I * * * + * * * I * R * P * * C *
```

Autor: Murtic Nirmel
www.etf.ba

45. Unosi se sa ulaza broj clanova. Zatim se unese IME, PREZIME i broj bodova svakog clana. Na izlazu treba ispisati clanove sortirane po broju bodova.

RJEŠENJE:

```
# include <stdio.h>
// Definisiranje strukture
struct struktura {
char ime[100];
char prezime[100];
int bodovi;
};

main(){
// Definisiranje varijabli
int n,i,j;
struct struktura spisak[100];
struct struktura pom;
// Unos broja clanova niza
printf("\n Unesite broj imena: "); scanf("%d", &n);
/* Unos imena, prezimena, bodova i direktan upis u strukturu*/
for (i=0;i<n;i++){
printf("\n Unesi %d ime:", i+1);
scanf("%s", &spisak[i].ime);
printf(" i prezime:");
scanf("%s", &spisak[i].prezime);
printf(" i bodove:");
scanf("%d", &spisak[i].bodovi);
}
/* Proces sortiranja */
for(i=0;i<(n-1);i++)
for(j=0;j<(n-1);j++)
if (spisak[j].bodovi<spisak[j+1].bodovi)
{
pom=spisak[j];
spisak[j]=spisak[j+1];
spisak[j+1]=pom;
}
// Ispis niza i kraj programa...
for (i=0;i<n;i++) printf("\n%d. [%s %s]: %d", i+1, spisak[i].prezime,
spisak[i].ime, spisak[i].bodovi);

getch();
return 0;
}
```

* * * C * C * * O * M * G * C * C * G * M * O * * C * C * * *
* * C C * * R G A M N * * + * * N M A G R * * C C * *
* C * * P * R * I * * * + * * * I * R * P * * C *

Autor: Murtic Nirmel
www.etf.ba

... Zadaci iz Cpp-a ...

```
***C**C**O**M**G**C**C**G**M**O**C**C**
**C**C**R**G**A**M**N**++**N**M**A**G**R**C**C**
**C**C**P**R**I**++**I**R**P**C**
```

Autor: Murtic Nirmel
www.etf.ba

46. Napisati program u cpp-u, koji sa ulaza prima dva cijela broja, a kao rezultat daje **zbir**, **razliku**, **proizvod** i **kolicnik**.

RJEŠENJE:

```
// Dva jednostavnija programa za pocetak, da se osjeti razlika c-a i cpp-a.

// Komentar iz cpp-a, 2 kose crte na pocetku svakog reda...
/* Komentar iz c-a... moze se koristiti i u cpp-u */
// U cpp-u se biblioteke iz c-a zapisuju na sljedeci nacin...
// u c-u: #include <stdio.h>
// u cpp-u: #include <cstdio> ili #include <cmath>
// znaci ispred biblioteke se dodaje slovo "c", a nastavak ".h" brisemo...
// Glavna biblioteka cpp-a, kao stdio.h u c-u...
#include <iostream>

using namespace std;

// Ova linija je kao i u c-u, osim sto uvijek moramo stavljati tip funkcije,
// u ovom slucaju "int" ispred, ako ne navedemo, program nece raditi... :)
int main() {

// Definisavanje varijabli (isto kao i u c-u, barem za sada ;) )...
int a,b;

// Funkcija za ispis na ekranu (ne samo ekranu, ali ce nam uglavnom za to
// koristiti, barem u pocetku... poput printf() u c-u)
// "<<" - citaj, salji na (ekran), primjetite da nema zagrada kao u c-u
// Moguce je i nadovezivanje vise znakova "<<"...
// "endl" - prijelaz u novi red, kao "\n" iz c-a, mada mozemo i dalje
// koristiti "\n" ali ako zelimo program cisto u duhu cpp-a onda cemo
// od sada na dalje i ubuduće koristiti endl... ;)
cout << "Unesite dva cijela broja: " << endl;

// Funkcija za unos podataka (kao scanf() u c-u)...
// ">>" - citaj, salji u (program), unos je mnogo jednostavniji od onog iz c-a
// u cpp-u smo postedjeni navodnika, definisanja tipova podataka, i znaka & :)
cin >> a >> b;

// Ispis zbira, razlike, proizvoda i kolicnika na ekran (sve su „integeri“
// osim kolicnika, koji je realan (double))...
cout << "Zbir brojeva je:" << a + b << ", razlika: " << a - b << ", proizvod: "
    << a * b << ", kolicnik: " << (double)a/b << endl;

// Na ovu funkciju "ignore" se naviknite, ako zelite da vidite ipak na ekranu
// rezultat, jer ako navedete samo getch(), nece ici bas najbolje, nemorate
// staviti 999, moze i neki drugi broj ;)
cin.ignore(999, '\n');

// Kao getch() u c-u, cekanje da se unese neki znak sa tastature...
getch();
// Vraca se 0 iz main funkcije (onako radi reda ;) )...
return 0;
// Kraj programa...
}
```

```
***C*C*O*M*G*C*C*G*M*O*C*C**
* * C C * * R G A M N * * + * * N M A G R * * C C * *
* C * * P * R * I * * * + * * * I * R * P * * C *
```

Autor: Murtic Nirmel
www.etf.ba

47. Napisati program koji ce ispisati na ekranu zeljeni broj tackica, broj tackica se unosi na ulazu...

RJEŠENJE:

```
// Ukljućivanje biblioteke u program
#include <iostream>

using namespace std;

// Definisavanje glavne funkcije programa
int main(){

// Definisavanje varijabli
int dots;

// Ispis odgovarajuće poruke na ekranu
cout << "Koliko tackica zelite: ";

// Beskonacna for petlja, prekinut cemo je sa funkcijom break...
for(;;) {

// Unos broja...
cin >> dots;

// Ukoliko je unesena vrijednost broj izlazi se iz for petlje
// u suprotnom se ispisuje odgovarajuća poruka, brise ulazni bufer
// i znakovi upisani u bufer i ponovo se unosi broj...
if(cin) break;
cout << "Ne zafrkajte se :P, unesite broj: " << endl;
cin.clear();
cin.ignore(10000, '\n');
}

// Nakon sto se unese ispravan broj, ispis poruke na ekran...
cout << dots << " tackica, samo za vas :) -->" << endl;

// I nakon poruke, jos ispis tackica...
for(int brojac=0;brojac<dots;brojac++) cout << ".";

// Brisanje ulaznog bufera i kraj programa...
cin.ignore(1000, '\n');
getchar();
return 0;
}

// P.S. Zadatak se moze rijesiti na 1000000... nacina... probajte drugacije ;)
```



```
* * * C * C * * O * M * G * C * C * G * M * O * * C * C * * *
* * C C * * R G A M N * * + + * * N M A G R * * C C * *
* C * * P * R * I * * * + * * * I * R * P * * C *
```

Autor: Murtic Nirmel
www.etf.ba

49. Sa ulaza se unose dimenzije 2 vektora (iste dimenzije), a zatim se izracuna treci vektor koji predstavlja zbir (prvi clan prvog vektora plus prvi clan drugog vektora...) dva unesena vektora.

RJEŠENJE:

```
// Pocetak programa, i ukljucivanje biblioteka u program
#include <iostream>
#include <vector>

using namespace std;

int main(){

/* Definisanje varijabli (trebalo bi ici ovo tek kad se koristi varijabla u
programu, ali ja ovako naviko :) */
vector<int> a(100), b(100), c;
int broj;

// Unos dimenzija vektora
cout << "Unesi broj elemenata vektora [1-100]: ";
cin >> broj;

// Unos elemenata prvog vektora
cout << "Unesi elemente prvog vektora: " << endl;
for(int i=0; i<broj; i++){
cin >> a[i];
}

// Unos elemenata drugog vektora
cout << "Unesi elemente drugog vektora: " << endl;
for(int i=0; i<broj; i++){
cin >> b[i];
}

/* Racunanje treceg vektora, tako sto se sabiraju odgovarajuci elementi,
vektora a i b... */
for(int i = 0; i < a.size(); i++) c.push_back(a[i] + b[i]);

// Ispis vektora c...
cout << "\"Zbir\" vektora je vektor: (";
for(int i=0; i<broj; i++) cout << c[i] << ", ";
cout << ")";

// Kraj programa...
cin.ignore(9999, '\n');
getchar();
return 0;
}
```



```
***C*C*O*M*G*C*C*G*M*O*C*C**
**C*C**R*G*A*M*N**++**N*M*A*G*R**C*C**
*C**P*R*I**++**I*R*P*C*
```

Autor: Murtic Nirmel
www.etf.ba

51. Napisati program koji racuna sumu trocifrenih prirodnih brojeva, kojima je zadnja cifra 2 puta veca od prve cifre. Rezultat ispisati na ekran.

RJEŠENJE:

```
#include <iostream>

using namespace std;

main(){

// Ispis poruke "Brojim..." :)
cout << "Brojim... :)" << endl;

// Definisanje varijable "suma" tipa int...
int suma=0;

/* Trostruka petlja, i se vrti od 0 do 9 i predstavlja cifru stotica, j se
isto vrti od 0 do 9 i predstavlja cifru desetice, a k predstavlja cifru
jedinica. Kada je uslov u trecoj for petlji zadovoljen tada se "suma" uvecava
za taj broj, ako nije zadovoljen prelazi se na iduci broj. */
for (int i=1;i<10;i++)
for (int j=0;j<10;j++)
for (int k=0;k<10;k++) {
if ((k>2*i)) suma=suma+(i*100+j*10+k);

}
// Ispis sume na ekranu...
cout << "Suma je:" << suma << endl;
getchar();
return 0;
}
```

```
* * * C * C * * O * M * G * C * C * G * M * O * * C * C * * *
* * C C * * R G A M N * * + * * N M A G R * * C C * *
* C * * P * R * I * * * + * * * I * R * P * * C *
```

Autor: Murtic Nirmel
www.etf.ba

52. Napisati program koji ucitani tekst štampa naopako. Tekst se ucitava s tastature sve do znaka za novi red.

RJEŠENJE:

```
#include<iostream>
#include<cstring>

using namespace std;

int main(){

// Ispis poruke "Unesite tekst"...
cout << "Unesite tekst: ";

// Definisanje i unos teksta koristeći funkciju getc() iz biblioteke "cstring"
char tekst[100];
gets(tekst);

// Pridruzivanje duzine teksta varijabli "duzina", pomocu funkcije strlen()...
int duzina=strlen(tekst);

// Ispis teksta "slovo po slovo" krenuvši sa kraja niza tekst[]...
for(int i=duzina-1;i>=0;i--) cout << tekst[i];

// Kraj programa
getchar();
return 0;
}
```

```
***C**O**M**G**C**G**M**O**C**C**
**C**R**G**A**M**N**++**N**M**A**G**R**C**C**
**C**P**R**I**++**I**R**P**C**
```

Autor: Murtic Nirmel
www.etf.ba

53. Napisati program koji ce izracunati ekvivalentnu otpornost paralelno vezanih otpornika.

RJEŠENJE:

```
#include <iostream>
#include <vector>

using namespace std;

/* Funkcija otpornici() koja prima broj otpornika, a kao rezultat vraca
vektor, koji sadrzi vrijednosti svih otpornika */
vector<int> unos(int broj){
vector<int> vrati(broj);
cout << endl << "Unesi vrijednosti otpornika: " << endl;
for(int i=0; i<broj; i++) {
cout << "Otpor: [" << i+1 << "]" << endl;
cin >> vrati[i];
}
return vrati;
}

/* Funkcija izracunaj() koja prima vektor koji sadrzi vrijednosti otpornika a
vraca vrijednost ekvivalentnog otpora */
float izracunaj(vector<int> a){
float suma;
for(int i=0; i<a.size(); i++) {
suma=suma+1./a[i];
}
return suma=1./suma;
}

int main(){

// Unos broja otpornika
cout << "Unesi broj otpornika: ";
int broj;
cin >> broj;

// Unos vrijednosti otpornika
vector<int> otpornici=unos(broj);

// Ispis ekvivalentnog otpora
cout << "Ekvivalentni otpor je: " << izracunaj(otpornici) << endl;

// Kraj programa
cin.ignore(9999, '\n');
getchar();
return 0;
}
```

```
***C*C*O*M*G*C*C*G*M*O**C*C**
* * C C * * R G A M N * * + * * N M A G R * * C C * *
* C * * P * R * I * * * + * * * I * R * P * * C *
```

Autor: Murtic Nirmel
www.etf.ba

54. Napišite funkciju "UnesiBroj" bez parametara, koja od korisnika očekuje da unese neki realni broj. Ukoliko je zaista unesen realan broj, funkcija treba da ga vrati kao rezultat iz funkcije. Ukoliko nije unesen broj, funkcija treba ispisati "Neispravan unos. Molimo, pokušajte ponovo." i ponovo tražiti broj, sve dok se zaista ne unese ispravan broj (prema tome, funkcija ne završava dok se ne unese ispravan broj). Napisanu funkciju iskoristite u programu koji traži da se unesu koeficijenti a , b i c linearne jednačine $ax + b = c$, a koji zatim nalazi i ispisuje njeno rješenje, pod uvjetom da ono postoji, ili odgovarajuću poruku u suprotnom.

RJEŠENJE:

```
# include <iostream>

using namespace std;

// Funkcija za unos broja
double Unesibroj(){
// Definisanje broja tipa double
double broj;
// Beskonacna for petlja
for(;;) {
cout << endl << "Unesite realan broj: ";
cin >> broj;
// Prekid for petlje kada se unese realan broj
if(cin) break;
// Ispis poruke, ako unos nije ispravan
cout << "Neispravan unos. Molimo, pokusajte ponovo.";
// Praznjenje bufera
cin.clear();
cin.ignore(10000, '\n');
}
// Vracanje broja iz funkcije
return broj;
}

int main(){

// Unos brojeva
double a=Unesibroj();
double b=Unesibroj();
double c=Unesibroj();
// Racunanje vrijednosti x
double x=(c-b)/a;
// Ispis rjesenja
if (a==0) cout << "Jednacina nema rjesenja" << endl;
else cout << "Rjesenje jednacine je: " << x << endl;
// Kraj programa
getchar();
return 0;
}
```

```
***C**C**O**M**G**C**C**G**M**O**C**C**  
**C**C**R**G**A**M**N**++**N**M**A**G**R**C**C**  
*C**P**R**I**++**I**R**P**C*
```

Autor: Murtic Nirmel
www.etf.ba

55. Napišite program koji traži da se sa tastature unese jedan kompleksan broj z i dva realna broja x i y , a koji zatim računa i ispisuje vrijednost izraza:

$$\left[x + \frac{z^2(2+3i)}{x-yi} \right]^3$$

RJEŠENJE:

```
# include <iostream>
# include <complex>

using namespace std;

int main(){

// Definisanje varijabli
complex<double> z;
double x, y;

// Unos kompleksnog broja
cout << "Unesi kompleksan broj z: " << endl;
cin >> z;

// Unos dva realna broja
cout << "Unesi dva realna broja: " << endl;
cin >> x >> y;

// Ispis rezultata
cout << "Vrijednost izraza: " << pow(x+((z*z*complex<double>(2, 3))/
(complex<double>(x, -y))),3) << endl;

// Kraj programa
getchar();
return 0;
}
```

```
* * * C * C * * O * M * G * C * C * G * M * O * * C * C * * *
* * C C * * R G A M N * * + * * N M A G R * * C C * *
* C * * P * R * I * * * + * * * I * R * P * * C *
```

Autor: Murtic Nirmel
www.etf.ba

56. Napraviti program za rješavanje kvadratne jednačine koji koristi kompleksne promjenljive. Pri tome, rješenja čiji je imaginarni dio jednak nuli treba ispisati tako da se ispisuju čisto kao realni brojevi, a ne kao uređeni parovi realnih brojeva.

RJEŠENJE:

```
#include <iostream>
#include <cmath>
#include <complex>

using namespace std;

int main() {

// Definisavanje i učitavanje varijabli
complex<double> a, b, c;
cout << "Unesi koeficijente:\n";
cin >> a >> b >> c;

// Racunanje diskriminante
complex<double> d=b*b-complex<double>(4,0)*a*c;

// Racunanje x1 i x2
complex<double>x1 = (-b-sqrt(complex<double>(d)))/(complex<double>(2,0)*a);
complex<double>x2 = (-b+sqrt(complex<double>(d)))/(complex<double>(2,0)*a);

/* Ako nema imaginarnog dijela onda ide dio u „if“ petlji, u suprotnom ide
„else“ dio */
if (imag(x1)!=0&&imag(x2)!=0) cout << "x1 = " << x1 << "\nx2 = " << x2 <<
endl; else {
if (imag(x1)==0) cout << "x1 = " << real(x1) << endl;
if (imag(x2)==0) cout << "x2 = " << real(x2) << endl;
}

// Kraj programa
getchar();
return 0;
}
```

```
***C*C*O*M*G*C*C*G*M*O*C*C**
* * C C * * R G A M N * * + * * N M A G R * * C C * *
* C * * P * R * I * * + * * I * R * P * * C *
```

Autor: Murtic Nirmel
www.etf.ba

57. Napišite program koji traži da se prvo unese prirodan broj "n", a nakon toga elementi vektora "a" koji ima "n" cjelobrojnih elemenata. Zatim, program treba da kreira dva nova vektora "b" i "c", i da u vektor "b" prepíše sve parne brojeve iz vektora "a", a u vektor "c" sve neparne brojeve iz vektora "a". Konačno, program treba da u jednom redu ispiše elemente vektora "b", a u drugom redu elemente vektora "c". Brojevi trebaju biti međusobno razdvojeni zarezom, pri čemu iza posljednjeg broja u svakom redu ne treba da bude zarez.

RJEŠENJE:

```
#include <iostream>
#include <vector>

using namespace std;

int main() {

// Definisanje varijabli...
int n;
vector<int> a, b, c;

// Unos prirodnog broja n...
cout << "Unesite prirodan broj \"n\": ";
cin >> n;

// Unos članova vektora a...
for (int i=0;i<n;i++) {
int upisi;
cout << "Unesi " << i+1 << " clan: ";
cin >> upisi;
a.push_back(upisi);
}

// Kreiranje vektora b i c od vektora a...
for(int i=0;i<n;i++) if (a[i]%2==0) b.push_back(a[i]); else c.push_back(a[i]);

// Ispis vektora na ekranu
if (b.size()>0) cout << "Ispis vektora b: " << endl;
for (int i=0;i<b.size();i++) if (i!=(b.size()-1)) cout << b[i] << ", ";
else cout << b[i] << endl;
if (c.size()>0) cout << "Ispis vektora c: " << endl;
for (int i=0;i<c.size();i++) if (i!=(c.size()-1)) cout << c[i] << ", ";
else cout << c[i] << endl;

// Kraj programa
getchar();
return 0;
}
```

```
***C*C**O*M*G*C*C*G*M*O**C*C**  
**C*C**R*G*A*M*N**++**N*M*A*G*R**C*C**  
*C**P*R*I**++**I*R*P**C*
```

Autor: Murtic Nirmel
www.etf.ba

58. Koristeci tip podataka "string" definiran u istoimenoj standardnoj biblioteci, napišite program koji traži da se sa tastature unesu dvije recenice, a zatim na ekranu ispisuje koliko svaka od recenica sadrži znakova, koja od ove dvije recenice dolazi prije po abecednom poretku, kao i recenicu koja se sastoji od ove dvije recenice sastavljene zajedno. Takodjer, ispišite prvu od ove dvije recenice u obrnutom poretku (od kraja ka pocetku). Pri tome, za tu svrhu koristite operacije koje su podržane za tip podataka "string".

RJEŠENJE:

```
# include <iostream>  
# include <string>  
  
using namespace std;  
  
main(){  
  
    // Definisanje dva stringa  
    string rec1, rec2;  
  
    // Unos dva stringa  
    cout << "Unesi prvu recenicu: ";  
    getline(cin, rec1);  
    cout << "Unesi drugu recenicu: ";  
    getline(cin, rec2);  
  
    // Ispis broja znakova prve i druge recenice  
    cout << "Prva recenica sadrzi: " << rec1.length() << " znakova..." << endl;  
    cout << "Druga recenica sadrzi: " << rec2.length() << " znakova..." << endl;  
  
    // Ispis recenica po abecednom redu  
    if (rec1<rec2) cout << "1: " << rec1 << endl << "2: " << rec2;  
    else cout << "1: " << rec2 << endl << "2: " << rec1;  
  
    // Sastavljanje prve i druge recenice u jednu  
    cout << endl << "Zajedno: " << rec1 + " " + rec2;  
  
    // Ispis prve recenice naopako  
    cout << endl << "Prva naopako: ";  
    for(int i=rec1.length()-1;i>=0;i--) cout << rec1[i];  
  
    // Kraj programa  
    cout << endl;  
    getchar();  
    return 0;  
}
```

* * * C * C * * O * M * G * C * C * G * M * O * * C * C * * * *
* * C C * * R G A M N * * + * * N M A G R * * C C * * *
* C * * P * R * I * * * + * * * I * R * P * * C *

Autor: Murtic Nirmel
www.etf.ba

59. Napišite funkciju koja racuna vrijednost funkcije $f(x, n)$ definirane sljedećim izrazom, pri čemu su x i n realni argumenti:

$$f(x, n) = \sum_{i=0}^n \frac{x^i}{\sqrt{x(x+i) - (-1)^i i^2}}$$

U slučaju da funkcija nije definirana, ona treba da baci izuzetak. To se može desiti u tri slučaja: ako n nije prirodan broj ili nula (tada suma nema smisla), zatim ukoliko se pod korijenom pojavi negativan broj, i konačno, ukoliko nazivnik dobije vrijednost nula. Za svaki od ova tri slučaja treba baciti različite tipove izuzetaka. Funkciju testirajte u glavnom programu koji u beskonačnoj petlji traži da se sa tastature unose vrijednosti argumenata x i n , a koji potom ispisuje vrijednost funkcije ili odgovarajuću poruku o grešci (na osnovu bacenog izuzetka). Pri testiranju obavezno pronadjite takve vrijednosti argumenata za koje će se desiti svaki od tri moguća izuzetka.

RJEŠENJE:

```
# include <iostream>
# include <cmath>

using namespace std;

double racunaj(double x, double n){

// y je konacna suma, na pocetku je 0...
double y=0;

// Ako je n<=0 baca se izuzetak
if (n<=0) throw "Greska: n mora biti vece od 0...";

// Racunanje sume
for (double i=0; i<=n; i++){

// Ako je podkorjen negativan baca se izuzetak
if (x*(x+i)-(pow(-1,i)*i*i)<0) throw "Greska: Korjen iz negativnog broja...";

// Ako je nazivnik nula baca se izuzetak
if (sqrt(x*(x+i)-pow(-1,i)*i*i)==0) throw "Greska: Nazivnik nemoze biti
nula...";

// Formula
y+=(pow(x,i))/sqrt(x*(x+i)-pow(-1,i)*i*i);
}

// Vracanje rezultata
return y;
}
```

```
***C*C*O*M*G*C*C*G*M*O*C*C**
**C*C**R*G*A*M*N**++**N*M*A*G*R**C*C**
*C**P*R*I**++**I*R*P*C*
```

Autor: Murtic Nirmel
www.etf.ba

```
main(){

// Try blok...
try{

// Definisanje varijabli i unos varijabli
double x,n;
cout << "Unesite n: "; cin >> n;
cout << "Unesite x: "; cin >> x;

// Ispis rezultata uz poziv funkcije
cout << endl << "Rezultat je: " << racunaj(x,n);
}

// Catch blok, hvata baceni izuzetak i ispisuje error poruku...
catch(const char poruka[]) {
cout << endl << poruka << endl;
}

// Kraj programa
cout << endl;
getchar();
return 0;
}
```

```
***C**O**M**G**C**G**M**O**C**C**
**C**R**G**A**M**N**++**N**M**A**G**R**C**C**
**C**P**R**I**++**I**R**P**C**
```

Autor: Murtic Nirmel
www.etf.ba

60. (*) Napisati program koji simulira Paint. Rad u konzoli. Jedan piksel predstavlja jedan znak '*'. Kursor koji crta po ekranu kreće se pomoću strelica. Boja piksela se mijenja pomoću tipke **ENTER**. A gumicu uključujemo na tipku **SPACE**. Program ima za cilj podsjetiti korisnike **C++** da se i u **C++** može formatirati i uljepšavati tekst. Poenta zadatka je razmotriti funkcije za lociranje kursora na određenu poziciju na ekranu kao i funkciju za mijenjanje boje teksta.

RJEŠENJE:

```
// Ovaj program je sa zvjezdicom, nije tezak, ali su uključene neke funkcije
za koje možda nikada necete ni cuti, i namijenjen je onima kojima nedostaju
neke komande iz PASCALA, kao LOCATE i TEXTCOLOR ;)

#include <iostream>
#include <windows.h> // uključeno zbog funkcija za rad sa bojom i koordinatama
#include <conio.h> // uključeno zbog funkcije kbhit()

using namespace std;

/* Funkcija koja mijenja boju teksta. Prima broj veći od 0. Svaki broj
predstavlja neku boju. Npr. 0 = crna, 12=crvena, 10=zelena, 9=plava... */
void PromjeniBoju(int boja){
    // ovo za sad samo zapamtite ;)
    HANDLE Konzola = GetStdHandle(STD_OUTPUT_HANDLE);
    // Promjena boje teksta u boju 'boja' koju je funkcija primila...
    SetConsoleTextAttribute(Konzola, boja);
}

/* Funkcija koja prima koordinate kursora tipa COORD, i postavlja kursor na tu
poziciju */
void LocirajKursor(COORD kordinate){
    HANDLE Konzola = GetStdHandle(STD_OUTPUT_HANDLE);
    // Postavljanje pozicije kursora na poziciju 'kordinate' koju je
funkcija primila...
    SetConsoleCursorPosition(Konzola,kordinate);
}

int main()
{
    // Definisiranje koordinata tipa COORD...
    COORD kordinate;
    // Definisiranje varijable boja i postavljanje na vrijednost 10(svijetlo-zelena)
    int boja=10;
    // Poziva se funkcija PromjeniBoju() koja će promijeniti boju teksta u zelenu
    PromjeniBoju(10);
    // Postavljanje koordinata kursora na poziciju (0,0)...
    kordinate.X = 0; kordinate.Y = 0;
    // Poziv funkcije koja će locirati kursor...
    LocirajKursor(kordinate);
}
```

```
*** C * C * * O * M * G * C * C * G * M * O * * C * C * * *
* * C * C * * R * G * A * M * N * * * + * * * N * M * A * G * R * * * C * C * *
* C * * P * R * I * * * + * * * I * R * P * * C *
```

Autor: Murtic Nirmel
www.etf.ba

```
// Ispis LEGENDE programa
cout << "Strelicama se pomjera zvjezdica...           Izlaz -->> ESC";
kordinate.X = 0; kordinate.Y = 1;
LocirajKursor(kordinate);
PromjeniBoju(9);
cout << "Enter mijenja boju zvjezdice...";
kordinate.X = 0; kordinate.Y = 2;
LocirajKursor(kordinate);
PromjeniBoju(12);
cout << "Space ukljucuje gumicu, iskljucuje se na ENTER...";

// Kursor se postavlja na poziciju (40,10) i tu se printa '*'
kordinate.X = 40; kordinate.Y = 10;
LocirajKursor(kordinate);
cout << "*";

/* Poziv beskonacne petlje u okviru koje se i izvrsava cio program, petlja se
moze normalnim putem prekinuti samo pritiskom na tipku ESC, koja predstavlja
izlaz iz programa */
for(;;)
// Provjera da li je pritisnuta neka tipka na tastaturi
    if (kbhit() != 0){
        // Ako je pritisnuta tipka, pridruzivanje ASCII koda tipke varijabli 'c'
        char c = getch();
        if (c==(00,72)) kordinate.Y-=1; // (00,72) ASCII kod strelice gore...
        if (c==(00,75)) kordinate.X-=1; // (00,75) ASCII kod strelice lijevo...
        if (c==(00,80)) kordinate.Y+=1; // (00,80) ASCII kod strelice dole...
        if (c==(00,77)) kordinate.X+=1; // (00,77) ASCII kod strelice desno...

        /* Kako bismo onemogucili "izlijetanje" zvjezdice van okvira ekrana,
        uvodimo sljedece restrikcije */
        if (kordinate.X==0) kordinate.X=1;
        if (kordinate.X==79) kordinate.X=78;
        if (kordinate.Y==3) kordinate.Y=4;
        if (kordinate.Y==50) kordinate.Y=49;

        // Ako se pritisne tipka SPACE (32) boja se postavlja na 0 (crna)
        if (c==32) boja=0;

        // Ako se pritisne tipka ESC (27) izlazi se iz programa
        if (c==27){
            system("cls"); // Brisanje ekrana
            boja=0; // Postavljanje boje na 0 (crna)

            // Beskonacna for petlja koja se prekida bilo kojom tipkom...
            for(;;){
                if (kbhit() != 0) return 0;

                // Poruke na izlazu
                kordinate.X = 30; kordinate.Y = 14;
                LocirajKursor(kordinate);
                PromjeniBoju(boja);
                cout << "Autor: Murtic Nirmel";
                kordinate.X = 26; kordinate.Y = 15;
                LocirajKursor(kordinate);
                PromjeniBoju(boja);
                cout << "Email: matrix.bih@gmail.com";
            }
        }
    }
}
```

Email: matrix.bih@gmail.com ; MSN: nirmel_m@hotmail.com

```
*** C * C * * O * M * G * C * C * G * M * O * * C * C * * *
* * C C * * R G A M N * * + * * N M A G R * * C C * *
* C * * P * R * I * * * + * * * I * R * P * * C *
```

Autor: Murtic Nirmel
www.etf.ba

```
        // Naizmjenicna izmjena boja, daje simpatican efekat ;)
        boja++;
        if (boja==12) boja=0;
    }
}

// Ako se pritisne tipka ENTER boja se umanjuje za 1...
if (c==13) boja--;

// Ako je boja manja od 0, vraca se na 12...
if (boja<0) boja=12;

// Lokacija kursora, promjena boje i printanje zvjezdice...
LocirajKursor(kordinate);
PromjeniBoju(boja);
cout << "*";
}

/* Ako nije pritisnuta tipka, printa se zvjezdica... ako se pitate zasto ovaj
dio uopste postoji, pa samo zato da bi kursor „treptio“ u suprotnom, kada
brisemo ne bismo vidjeli na kojoj je poziciji kursor ;) */
else
{
    LocirajKursor(kordinate);
    cout << "*";
}

// Kraj programa, do ovoga se nemoze doci, barem ne normalnim putem... ;)
return 0;
}
```

```
***C*C*O*M*G*C*C*G*M*O**C*C**
**C*C**R*G*A*M*N**++**N*M*A*G*R**C*C**
*C**P*R*I**++**I*R*P**C*
```

Autor: Murtic Nirmel
www.etf.ba

61. Napišite dvije preklopljene verzije funkcije "RadijaniUStepene" koje pretvaraju radijane u stepene. Prva verzija prihvata dva realna parametra "alfa" i "stepeni", dok druga verzija prihvata četiri parametra nazvana "alfa", "stepeni", "minute" i "sekunde", od kojih je prvi realan a ostali cjelobrojni. U oba slucaja prvi parametar predstavlja neki ugao u radijanima. Prva verzija funkcije treba da pretvori tu vrijednost u stepene (kao realan broj) i smjesti pretvorenu vrijednost u drugi parametar. Druga verzija funkcije treba da pretvori vrijednost ugla u stepene, minute i sekunde i da smjesti pretvorenu vrijednost redom u drugi, treci i cetvrti parametar. Na primjer, ukoliko se izvrse naredbe:

```
RadijaniUStepene(1.326, s);
```

```
RadijaniUStepene(1.326, x, y, z);
```

u promjenljive "s", "x", "y" i "z" (pod pretpostavkom da su ispravno deklarirane) treba da se smjeste brojevi **75.9742, 75, 58 i 27**, jer je **1.326 rad = 75.9742° = 75° 58' 27"**. Napišite i kratku glavnu funkciju ("main") u kojoj cete demonstrirati napisanu funkciju.

RJEŠENJE:

```
# include <iostream>
```

```
using namespace std;
```

```
/* Prima realan broj alfa, i stvara se referenca &stepeni na s koji je nedefinisan u startu, ali u funkciji dobije neku vrijednost preko reference */
```

```
void RadijaniUStepene(double alfa, double &stepeni) {  
    stepeni=alfa/3.1415*180.; }
```

```
/* Prima realan broj alga, i stvaraju se 3 reference preko kojih dobijemo vrijednosti za stepene, minute i sekunde, vrijednosti se pamte i zapisuju u varijable koje su poslate i u jednoj i u drugoj funkciji jer promjena reference unutar funkcije mijenja i varijablu van funkcije na koju je stvorena referenca*/
```

```
void RadijaniUStepene(double alfa, int &stepeni, int &minute, int &sekunde){  
    double pom;  
    RadijaniUStepene(alfa,pom);  
    stepeni=int(pom);  
    pom=pom-stepeni;  
    pom=pom*60;  
    minute=int(pom);  
    pom=pom-minute;  
    sekunde=int(pom*60);  
}
```

```
***C**C**O**M**G**C**C**G**M**O**C**C**  
**C**C**R**G**A**M**N**++**N**M**A**G**R**C**C**  
*C**P**R**I**++**I**R**P**C*
```

Autor: Murtic Nirmel
www.etf.ba

```
int main(){  
// Tri cijela broja x,y,z (stepeni, minute, sekunde respektivno)  
int x,y,z;  
// Dva realna broja (ugao i stepeni respektivno)  
double r,s;  
// Unos ugla  
cout << "Unesi x: "; cin >> r;  
// Funkcija koja "vraca" samo stepene i to u realnom obliku  
RadijaniUStepene(r, s);  
// Funkcija koja vraca stepene, minute, sekunde u cjelobrojnom obliku  
RadijaniUStepene(r, x, y, z);  
// Ispis stepeni u realnom obliku  
cout << "Stepeni: " << s << endl;  
// Ispis stepeni, minuta, sekundi u cjelobrojnom obliku  
cout << "Stepeni: " << x << ", minute:" << y << ", sekunde: " << z << endl;  
// Kraj programa  
getchar();  
return 0;  
}
```

```
* * * C * C * * O * M * G * C * C * G * M * O * * C * C * * *
* * C C * * R G A M N * * + * * N M A G R * * C C * *
* C * * P * R * I * * * + * * * I * R * P * * C *
```

Autor: Murtic Nirmel
www.etf.ba

62. Napišite generičku funkciju "UnosBroja" sa tri parametra. Funkcija treba da omogući pouzdano unošenje brojeva u program, uz potpunu kontrolu grešaka pri unosu. Prvi i drugi parametar su tipa "string". Pri tome, prvi parametar predstavlja tekst koji se ispisuje korisniku kao obavijest da treba unijeti broj (prompt), dok drugi parametar predstavlja tekst koji se ispisuje korisniku kao upozorenje u slučaju da unos nije ispravan. Treći parametar je referenca na proizvoljni numerički tip, a predstavlja promjenljivu u koju će se smjestiti uneseni broj. Na primjer, funkcija se može pozvati na sljedeći način:

```
UnosBroja("Unesi prvi broj: ", "Neispravan unos!\n", prvi_broj);
```

Funkcija treba da traži unos od korisnika sve dok unos ne bude ispravan. Napisanu funkciju demonstrirajte u testnom programu koji od korisnika traži da unese realni broj x i cijeli broj n , a zatim računa i ispisuje vrijednost stepena x^n .

RJEŠENJE:

```
# include <iostream>
# include <cmath>
# include <string>

using namespace std;
/* Generička funkcija, u zavisnosti od tipa treceg parametra koji je poslat u
funkciju zavisi i koji to tip predstavlja tip 'tip' */
template<typename tip>
/* Primaju se dva stringa p1 i p2 i referenca na treci parametar pri pozivu
funkcije tipa 'tip' */
void UnosBroja(string p1, string p2, tip &t){
    for (;;) {
        cout << p1; cin >> t; // Ispis poruke i unos broja
        /* Ako je pravilan unos prekida se funkcija, referenca se promjenila, mijenja
se i treci parametar funkcije koristen pri pozivu */
        if (cin) break;
    }
    // Ako nije ispravan unos printa se greska i cisti buffer i ponavlja se proces
    cout << p2;
    cin.clear();
    cin.ignore(10000, '\n');
}
int main(){
    // Definisanje cijelog i realnog broja
    int prvi_broj;
    double drugi_broj;
    /* Poziv funkcije sa doticnim parametrima, jednom kao treci parametar ide cio
broj a drugi put realan */
    UnosBroja("unesi prvi broj: ", "Neispravan unos!\n", prvi_broj);
    UnosBroja("unesi drugi broj: ", "Neispravan unos!\n", drugi_broj);
    // Ispis rezultata na osnovu dobijenih podataka
    cout << "Rezultat: " << pow(prvi_broj, drugi_broj) << endl;
    // Kraj programa
    getchar();
    return 0;
}
```

```
***C**C**O**M**G**C**C**G**M**O**C**C**
**C**C**R**G**A**M**N**++**N**M**A**G**R**C**C**
**C****P**R**I**++**I**R**P**C**
```

Autor: Murtic Nirmel
www.etf.ba

63. Napišite generičku funkciju "BrojZajednickih" sa četiri parametra "niz1", "n1", "niz2" i "n2". Parametri "niz1" i "niz2" su klasični nizovi (ne vektori) proizvoljnog ali istog tipa elemenata (tj. tip elemenata u oba niza je isti), a "n1" i "n2" predstavljaju brojeve elemenata u tim nizovima respektivno. Funkcija treba da kao rezultat vrati broj elemenata koji se javljaju kao zajednički elementi i u jednom i u drugom nizu (radi jednostavnosti, podrazumijevati da nizovi ne sadrže elemente koji se ponavljaju). Na primjer, neka su date sljedeće deklaracije:

```
int a[5] = {3, 7, 2, 1, 5};
int b[6] = {4, 6, 7, 8, 1, 3};
```

Tada sljedeća naredba:

```
cout << Zajednicki(a, 5, b, 6);
```

treba da ispiše broj 3, jer se tri elementa (3, 7 i 1) pojavljuju u oba niza. Napišite i kratki testni program u kojem ćete demonstrirati napisanu funkciju.

RJEŠENJE:

```
# include <iostream>

using namespace std;

/* Genericka funkcija u kojoj je definisan tip kao sto je tip varijable koja
se salje u funkciju */
template<typename tip>
int BrojZajednickih(tip niz1, int n1, tip niz2, int n2){
/* 'br' je brojac, na pocetku 0, kada bude i-ti clan prvog niza jednak bilo
kojem (citaj j-tom) clanu drugog niza, tada se br uvecava za 1*/
int br=0;
for (int i=0;i<n1;i++) for (int j=0;j<n2;j++) if (niz1[i]==niz2[j]) br++;
// Vracanje brojaca
return br;
}
int main(){
// Definisane duzine nizova i njihov unos
int n1, n2;
cout << "Unesi duzinu prvog niza: "; cin >> n1;
cout << "Unesi duzinu drugog niza: "; cin >> n2;
// Definisane nizova i njihov unos
int niz1[n1], niz2[n2];
cout << "Unesi clanove prvog niza: " << endl;
for (int i=0; i<n1; i++){
    cout << i+1 << ": "; cin >> niz1[i]; }
cout << "Unesi clanove drugog niza: " << endl;
for (int i=0; i<n2; i++){
    cout << i+1 << ": "; cin >> niz2[i]; }
// Ispis rezultata uz poziv funkcije
cout << "Broj istih elemenata" << BrojZajednickih(niz1,n1,niz2,n2) << endl;
// Kraj programa
getchar();
return 0;
}
```

```
***C*C*O*M*G*C*C*G*M*O*C*C***
**C*C**R*G*A*M*N**++**N*M*A*G*R**C*C**
*C**P*R*I**++**I*R*P**C*
```

Autor: Murtic Nirmel
www.etf.ba

64. Napišite generičku funkciju "Izvrni" sa jednim parametrom, koji predstavlja vektor elemenata proizvoljnog tipa. Funkcija treba da *ispremješta* elemente vektora tako da prvi element postane posljednji, a posljednji prvi. Na primjer, ukoliko se u glavnom programu izvrši sekvenca naredbi:

```
vector<int> a(5)
a[0] = 3; a[1] = 5; a[2] = 6; a[3] = 1; a[4] = 2;
Izvrni(a, 5);
for(int i = 0; i < 5; i++) cout << a[i] << " ";
```

ispis na ekranu treba da bude "2 1 6 5 3". Funkcija treba da bude realizirana da vrši premještanje "u mjestu", odnosno da ne koristi *nikakav pomoćni vektor* osim vektora prenesenog kao parametar (uputa: razmijenite prvi element sa posljednjim, drugi sa pretposljednji, itd.). Napišite i kratki glavni program u kojoj ćete demonstrirati napisanu funkciju na sekvenci brojeva koji se unose sa tastature.

Oprez: funkcija izvrni ne smije sama po sebi ništa da ispisuje na ekran. Njeno dejstvo treba samo da bude *izmjena poretka elemenata u vektoru*, koje će kasnije na ekran ispisati neko drugi (npr. "main" funkcija).

RJEŠENJE:

```
# include <iostream>
# include <vector>

using namespace std;

// Genericka funkcija izvrni

template<typename niz>
// Prima se vektor u funkciju
vector<niz> izvrni(vector<niz> a){

/* Vrti se petlja od nultog clana do sredine vektora i izmjenjuju se clanovi
simetricno */
    for (int i=0;i<a.size()/2;i++){
        niz pom;
        pom=a[i];
        a[i]=a[a.size()-1-i];
        a[a.size()-1-i]=pom;
    }

// Vracanje obrnutog vektora
return a;
}
```

```
***C* C* *O* M* G* C* C* G* M* O* *C* C* * *
* * C C * * R G A M N * * + * * N M A G R * * C C * *
* C * * P * R * I * * * + * * * I * R * P * * C *
```

Autor: Murtic Nirmel
www.etf.ba

```
int main(){

// Definisiranje duzine vektora, vektora i unos clanova vektora
vector<char> a;
int n;
cout << "Unesi duzinu vektora: "; cin >> n;
cout << "Unesi clanove vektora: " << endl;
for (int i=0; i<n; i++){
char pom;
cout << i+1 << ": ";
cin >> pom;
a.push_back(pom);
}

// Poziv funkcije izvрни() u koju saljemo vektor
a=izvrni(a);

// Ispis rezultata
for(int i = 0; i < a.size(); i++) cout << a[i] << " " << endl;

// Kraj programa
getchar();
return 0;
}
```

```
***C*C*O*M*G*C*C*G*M*O*C*C**
*CC*RGAMN**++*NMAGR**CC**
*C**P*R*I**++**I*R*P**C*
```

Autor: Murtic Nirmel
www.etf.ba

65. Napišite generičku funkciju "Odstrani" sa dva parametra "v" i "prag". Prvi parametar je vektor elemenata koji pripadaju konceptu uporedivih elemenata (tj. elemenata za koje su definirane operacije poređenja, tako da se mogu porediti po veličini). Funkcija treba da odstrani sve elemente iz zadanog vektora čija je vrijednost manja od vrijednosti parametra "prag". Na primjer, neka vektor "a" realnih brojeva sadrži redom vrijednosti 3.5, -7.11, 2, 1.3, -5, -6.12, 4, 5, -6.3 i 0.

Nakon poziva funkcije:

Odstrani(a, 0);

elementi vektora "a" treba da postanu 3.5, 2, 1.3, 4, 5 i 0 (odstranjeni su elementi manji od nule). Funkcija "Odstrani" ne treba da koristi nikakav pomoćni vektor osim samog parametra "v". Također, za realizaciju ove funkcije ne smijete koristiti nikakve koncepte koji nisu obrađivani na predavanjima (npr. funkciju "erase", "iteratore", itd.). Napišite i kratki testni program u kojem ćete demonstrirati napisanu funkciju na sekvenci realnih brojeva koji se unose sa tastature (vrijednost parametra "prag" se također unosi sa tastature).

RJEŠENJE:

```
# include <iostream>
# include <vector>

using namespace std;

// Generička funkcija odstrani
template<typename niz>

// Prima se vektor i prag u funkciju
vector<niz> odstrani(vector<niz> a, niz prag){

    int br=0; // br je brojac koji predstavlja broj clanova manjih od praga

    /* Vrti se petlja od nultog clana do kraja i ako je i-ti clan vektora manji od
    praga tada se br uveca za 1, a ako nije tada se broj upise na poziciju koja je
    za br puta manja od trenutne pozicije, znaci ako smo dosli do 5 clana i do
    tada su nadjena dva clana manji od praga tada ce se 5 clan upisati na 5-2=3
    poziciju, a na kraju ce se jos velicina vektora "umanjiti" za br i dobili smo
    vektor koji smo trebali dobiti :) */
    for (int i=0; i<a.size(); i++) {
        if (a[i]<prag) br++; else a[i-br]=a[i];
    }
    a.resize(a.size()-br);

    // Vracanje obrnutog vektora
    return a;
}
```

```
* * * C * C * * O * M * G * C * C * G * M * O * * C * C * * * *
* * C C * * R G A M N * * + * * N M A G R * * C C * *
* C * * P * R * I * * * + * * * I * R * P * * C *
```

Autor: Murtic Nirmel
www.etf.ba

```
int main(){

// Definisiranje duzine vektora, vektora i unos clanova vektora
vector<int> a;
int n;
cout << "Unesi duzinu vektora: "; cin >> n;
cout << "Unesi clanove vektora: " << endl;
for (int i=0; i<n; i++){
int pom;
cout << i+1 << ": ";
cin >> pom;
a.push_back(pom);
}

// Definisiranje praga i unos praga
int prag;
cout << "Unesi prag: "; cin >> prag;

// Poziv funkcije izvorni() u koju saljemo vektor i prag
a=odstrani(a, prag);

// Ispis rezultata
for(int i = 0; i < a.size(); i++) cout << a[i] << " ";

// Kraj programa
getchar();
return 0;
}
```

```
***C*C*O*M*G*C*C*G*M*O*C*C**
**C*C**R*G*A*M*N**++**N*M*A*G*R**C*C**
*C**P*R*I**+***I*R*P**C*
```

Autor: Murtic Nirmel
www.etf.ba

66. Poznata Moievrova teorema za kompleksne brojeve tvrdi da je za ma kakve realne brojeve x i n :

$$(\cos x + i \sin x)^n = \cos nx + i \sin nx$$

Napisati program koji će za realne brojeve x i n unesene sa tastature izračunati lijevu i desnu stranu ovog izraza i ispisati ih na ekranu, koristeći kompleksni tip podataka. Na nekoliko primjera provjeriti da li se zaista dobijaju jednake vrijednosti.

RJEŠENJE:

```
# include <iostream>
# include <cmath>
# include <complex>

using namespace std;

int main() {

// Definisanje i unos dva realna broja
double x, n;
cout << "Unesi realan broj \"x\": ";
cin >> x;
cout << "Unesi realan broj \"n\": ";
cin >> n;

// Racunanje prvog dijela (prvidio je rezultat)
complex<double> prvidio(cos(x), sin(x));
prvidio=pow(prvidio,n);

// Racunanje drugog dijela (drugidio je rezultat)
complex<double> drugidio(cos(n*x), sin(n*x));

// Ispis rezultata
cout << "Lijeva strana jednacine je: " << prvidio << " a desna: " << drugidio
<< endl;
getchar();
return 0;
}
```

```
* * * C * C * * O * M * G * C * C * G * M * O * * C * C * * *
* * C C * * R G A M N * * + * * N M A G R * * C C * *
* C * * P * R * I * * * + * * * I * R * P * * C *
```

Autor: Murtic Nirmel
www.etf.ba

67. Deklarirajte pobrojani tip "Dani" takav da promjenljive ovog tipa mogu uzimati samo vrijednosti "Ponedjeljak", "Utorak", "Srijeda", "Cetvrtak", "Petak", "Subota" i "Nedjelja". Zatim napišite funkciju "StampajKalendar" koja ne vraća nikakav rezultat, a zahtijeva dva parametra. Prvi parametar "broj_dana" je tipa "int", dok je drugi parametar "pocetni_dan" tipa "Dani". Parametar "broj_dana" određuje broj dana u mjesecu, dok parametar "pocetni_dan" određuje dan u sedmici kojim započinje taj mjesec. Funkcija "StampajKalendar" štampa kalendar za taj mjesec. Na primjer, ukoliko želimo odštampati kalendar za mjesec koji ima 31 dan, a počinje srijedom, to možemo uraditi sljedećim pozivom:

StampajKalendar(31, Srijeda);

Taj poziv treba da na ekranu proizvede ispis poput sljedećeg:

```
  P  U  S  Č  P  S  N
      1  2  3  4  5
6   7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31
```

RJEŠENJE:

```
# include <iostream>
# include <iomanip>

using namespace std;
// Definisavanje pobrojanog tipa Dani
enum Dani {Ponedjeljak, Utorak, Srijeda, Cetvrtak, Petak, Subota, Nedjelja};

// Funkcija koja stampa kalendar
void StampajKalendar(int broj_dana, Dani pocetni_dan){
cout << endl << endl << "  P  U  S  C  P  S  N" << endl;

// Stampanje odgovarajućeg broja praznih mjesta na početku
for (int i=0; i<pocetni_dan; i++) cout << setw(3) << " ";

// Dodjeljivanje odgovarajuće vrijednosti varijabli b
int b=pocetni_dan;

// Ispis rednih brojeva dana
for (int i=1; i<=broj_dana; i++) {
cout << setw(3) << i;
b++;
// Prelazak u novi red nakon ispisa svakih 7 brojeva
if (b==7) { b=0; cout << endl; }
}

}
```

```
***C*C*O*M*G*C*C*G*M*O*C*C**
* C C * R G A M N * * + * * N M A G R * * C C * *
* C * * P * R * I * * * + * * * I * R * P * * C *
```

Autor: Murtic Nirmel
www.etf.ba

```
int main(){

// Definisiranje varijabli
int broj_dana, redni_broj_dana;
Dani pocetni_dan;

// Unos rednog broja dana od kojeg pocinje mjesec
while(redni_broj_dana<=0||redni_broj_dana>=8){
cout << "Unesi redni broj dana u sedmici kojim pocinje mjesec (1-7): ";
cin >> redni_broj_dana;
}

// Unos broja dana u mjesecu
while(broj_dana<=27||broj_dana>=32){
cout << "Unesi broj dana u mjesecu (28-31): ";
cin >> broj_dana;
}

// Dodjeljivanje odgovarajuće vrijednosti varijabli pocetni_dan
pocetni_dan=Dani(redni_broj_dana-1);

// Poziv funkcije i slanje argumenata
StampajKalendar(broj_dana, pocetni_dan);

// Kraj programa
getchar();
return 0;
}
```

```
* * * C * C * * O * M * G * C * C * G * M * O * * C * C * * *
* * C * C * * R * G * A * M * N * * * + * * * N * M * A * G * R * * * C * C * *
* * C * * P * R * I * * * + * * * I * R * P * * * C * *
```

Autor: Murtic Nirmel
www.etf.ba

68. Napišite program koji traži da se prvo unese prirodan broj "n", a nakon toga elementi vektora "a" koji ima "n" cjelobrojnih elemenata. Zatim, program treba da kreira novi vektor "b" u koji će biti prepisani elementi vektora "a" izostavljajući duplikate (tj. brojeve koji se ponavljaju). Na kraju, program treba da ispiše elemente vektora "b", kao i broj elemenata koji se ponavljaju. Na primjer, ukoliko se sa tastature unese niz od **15** brojeva: **7, 10, 4, 2, 4, 4, 5, 6, 7, 3, 9, 1, 8, 6 i 7** program treba kao elemente vektora "b" da ispiše brojeve **7, 10, 4, 2, 5, 6, 3, 9, 1 i 8**, kao i da ispiše da se **3** broja ponavljaju (to su brojevi **7, 4 i 6**).

RJEŠENJE:

```
# include <iostream>
# include <vector>

using namespace std;

int main(){
// Definisanje varijabli
int ponavljanja=0;
vector<int> a, b, pom;

// Unos broja clanova vektora
cout << "Unesi prirodan broj \"n\": ";
int n;
cin >> n;

// Unos clanova vektora
cout << "Unesi clanove vektora: ";
for (int i=0; i<n; i++){
int broj;
cin >> broj;
a.push_back(broj);
}

for (int i=0; i<n; i++){
int istina=0;

// Ako broj a[i] vec postoji u pomocnom nizu onda se on preskace...
for (int k=0; k<pom.size(); k++) if (a[i]==pom[k]) istina=1;
if (istina==1) continue;

/* Ako je broj a[i] jednak nekom od ostalih onda se varijabla "ponavljanja"
uveca za 1, a broj a[i] se upisuje u pomocni niz u koji su zapisani brojevi
koji se ponavljaju...*/
for (int j=0; j<n; j++) if ((a[i]==a[j])&&(i!=j)) { pom.push_back(a[i]);
ponavljanja++; break;}
}
```

```
***C*C*O*M*G*C*C*G*M*O*C*C***
**C**R*G*A*M*N**++**N*M*A*G*R**C**
**C**P*R*I**++**I*R*P**C**
```

Autor: Murtic Nirmel
www.etf.ba

```
// Prepis broja a[i] u vektor b
b.push_back(a[i]);
}

// Ispis clanova vektora b i odgovarajuće poruke...
for (int i=0; i<b.size(); i++) cout << b[i] << " ";
cout << endl << "Imamo " << ponavljanja << " ponovljenih brojeva, to su: ";

// Ispis clanova koji su se ponavljali
for (int i=0; i<pom.size(); i++) cout << pom[i] << " ";
cout << endl;

// Kraj programa
getchar();
return 0;
}
```

```
***C*C*O*M*G*C*C*G*M*O*C*C**
*CC*RGAMN**++*NMAGR**CC*
*CP*RI**+**IR*P*C*
```

Autor: Murtic Nirmel
www.etf.ba

69. Napišite funkcije "SviElementiSuJednaki" i "SviElementiSuRazliciti". I jedna i druga funkcija primaju kao parametar jedan vektor realnih brojeva, a vraćaju kao rezultat logičku vrijednost (tj. vrijednost tipa "bool"). Funkcija "SviElementiSuJednaki" vraća vrijednost "true" ako i samo ako su svi elementi vektora međusobno identični, a u suprotnom vraća "false". Funkcija "SviElementiSuRazliciti" vraća vrijednost "true" ako i samo ako su svi elementi vektora međusobno različiti (tj. ukoliko nikoja dva elementa nisu međusobno jednaka), a u suprotnom vraća "false". Napisane funkcije testirati u glavnom programu ("main" funkciji) na vektoru čiji se elementi unose sa tastature.

RJEŠENJE:

```
# include <iostream>
# include <vector>

using namespace std;

// Funkcija SviElementiSuJednaki()...
bool SviElementiSuJednaki(vector<double> vektor){
    bool istina=true;
    double pom=vektor[0];
    for (int i=1;i<vektor.size();i++) if (pom!=vektor[i]){istina=false; break;}
    return istina;
}

// Funkcija SviElementiSuRazliciti()...
bool SviElementiSuRazliciti(vector<double> vektor){
    bool istina=true;
    for (int i=0; i<vektor.size(); i++)
        for (int j=i+1; j<vektor.size(); j++)
            if (vektor[i]==vektor[j]) { istina=false; break; }
    return istina;
}

main(){
    vector<double> vektor;

    // Unos broja clanova vektora
    cout << "Unesi prirodan broj \"n\": ";
    int n;
    cin >> n;

    // Unos clanova vektora
    cout << "Unesi clanove vektora: ";
    for (int i=0; i<n; i++){
        int broj;
        cin >> broj;
        vektor.push_back(broj);
    }
}
```

```
*** C * C * * O * M * G * C * C * G * M * O * * C * C * * *
* * C * C * * R * G * A * M * N * * * + * * * N * M * A * G * R * * * C * C * * *
* C * * P * R * I * * * + * * * I * R * P * * * C * *
```

Autor: Murtic Nirmel
www.etf.ba

```
// Ispis odgovarajucih poruka
if(SviElementiSuJednaki(vektor)) cout << "Svi elementi su identicni... ";
if(SviElementiSuRazliciti(vektor)) cout << "Svi elementi su razliciti... ";

// Kraj programa
getchar();
return 0;
}
```

```
***C* C* *O* M* G* C* C* G* M* O* *C* C* * *
* * C* C* * R G A M N * * + * * N M A G R * * C C * *
* C * * P * R * I * * * + * * * I * R * P * * C *
```

Autor: Murtic Nirmel
www.etf.ba

70. Napišite funkciju sa jednim argumentom n koja kreira dinamički niz od n cijelih brojeva, popunjava ga sa prvih n Fibonacijevih brojeva, i vraća kao rezultat pokazivac na prvi element kreiranog niza. Napisanu funkciju testirajte u kratkom testnom programu koji prvo sa tastature unosi broj n , zatim ispisuje elemente kreiranog niza, i konacno, uništava kreirani niz.

RJEŠENJE:

```
#include <iostream>

using namespace std;

/* Funkcija Kreiraj koja prima prirodan broj n, kreira dinamički niz od n
elemenata, popunjava se niz a vraća se pokazivac na prvi član*/
int *Kreiraj(int n){
// Definisane i kreiranje dinamičkog niza niz[]
int *niz = new int[n];
// Popunjavanje niza fibonacijevim ciframa
for(int i=0; i<n; i++){
if (i==0) niz[i]=0;
else if (i==1) niz[i]=1;
else niz[i]=niz[i-1]+niz[i-2];
}
// Vraćanje pokazivaca iz funkcije
return niz;
}

int main(){
/* U funkciji postoji mogućnost da bude bacen izuzetak, koji ćemo uhvatiti u
catch bloku i ispisati grešku. */
try {
// Unos broja članova niza
int n;
cout << "Unesite broj članova niza";
cin >> n;
/* Definisane dinamičkog niza "dinamickiniz[]" koji će poprimiti vrijednosti
dinamičkog niza kojeg vrati funkcija Kreiraj(n) */
int *dinamickiniz=Kreiraj(n);
/* Ispis članova niza (fibonacijev niz) */
for (int i=0;i<n;i++) if (i!=(n-1)) cout << dinamickiniz[i] << ", ";
else cout << dinamickiniz[i] << endl;
// Unistavanje dinamičkog niza...
delete[] dinamickiniz;
}
// Dio koji ide ako se baci izuzetak...
catch(...) {
cout << "Problem: Nema dovoljno memorije!\n";
}
// Kraj programa
getchar();
return 0;
}
```

```
***C*C*O*M*G*C*C*G*M*O*C*C**
**C*C**R*G*A*M*N**++**N*M*A*G*R**C*C**
* C * * P * R * I * * * + * * * I * R * P * * C *
```

Autor: Murtic Nirmel
www.etf.ba

71. Napišite program koji od korisnika traži da unese dimenzije kvadratne matrice n , a zatim da unese dvije matrice formata $n \times n$ (pretpostavite da su elementi matrica realni brojevi). Program nakon toga treba da ispiše zbir dvije unesene matrice. U programu treba da se nalaze sljedeće funkcije: "KreirajMatricu", "UnesiMatricu", "SaberMatrice", "IspisiMatricu" i "UnistiMatricu". Funkcija "KreirajMatricu" prima kao parametre dvojni pokazivač koji služi za pristup dinamički kreiranoj matrici (u nastavku ćemo ovaj pokazivač prosto zvati **dinamička matrica**), kao i dimenziju matrice n . Funkcija treba da alocira prostor za matricu formata $n \times n$, i dodijeli adresu alociranog prostora pokazivaču koji se koristi za pristup njenim elementima (za tu svrhu, odgovarajući formalni parametar mora biti **referenca na dvojni pokazivač**). Funkcija "UnesiMatricu" popunjava matricu elementima unesenim sa tastature, a prima kao parametre dinamičku matricu i dimenziju n . Funkcija "SaberMatrice" prima kao parametre dvije dinamičke matrice i dimenziju n . Ova funkcija treba da *kreira* novu dinamičku matricu (pozivom funkcije "KreirajMatricu"), da je popuni zbirom dvije dinamičke matrice koje su joj proslijeđene kao parametri, i da vrati kao rezultat dvojni pokazivač koji služi za pristup elementima novokreirane matrice. Funkcija "IspisiMatricu" kao parametre prima dinamičku matricu i dimenziju n , a ispisuje elemente matrice na ekran. Konačno, funkcija "UnistiMatricu" unistava dinamički kreiranu matricu koja joj se prosljeđuje kao parametar (zajedno sa dimenzijom n). Funkcija "KreirajMatricu" treba da baci izuzetak u slučaju da kreiranje ne uspije. Pri tome, ova funkcija mora da vodi računa da u slučaju da dođe do bacanja izuzetka "**počisti iza sebe**" sve uspjele alokacije, tako da ne dođe do curenja memorije. Bačeni izuzetak treba hvatati u glavnom programu. Obavezno testirajte slučaj kada alokacija ne uspijeva (unosom prevelikog broja n).

RJEŠENJE:

```
#include <iostream>

using namespace std;

// Funkcija koja unistava matricu odredjenih dimenzija koju prima kao
// parametar...
int UnistiMatricu(double **MATRIX, int n){
if (MATRIX == 0) { cout << "Ova matrica nije nikad kreirana, pa je necemo
brisati... \n"; return 1; }
cout << "Brisem matricu... \n";
for(int i = 0; i < n; i++) delete[] MATRIX[i];
delete[] MATRIX;
cout << "Uspjesno izbrisana matrica... \n";
}
```

Email: matrix.bih@gmail.com ; MSN: nirmel_m@hotmail.com

```
***C*C*O*M*G*C*C*G*M*O*C*C**
**C**R*G*A*M*N**++**N*M*A*G*R**C*C**
*C**P*R*I**++**I*R*P**C*
```

Autor: Murtic Nirmel
www.etf.ba

```
/* Funkcija koja kreira matricu dimenzija n x n, baca izuzetak ako nemoze
kreirati... */
void KreirajMatricu(double **&MATRIX, int n){
MATRIX = new double*[n];
for(int i = 0; i < n; i++) MATRIX[i] = new double[n];
}

/* Funkcija koja prima praznu matricu, a vraca matricu koja je popunjena
elementima preko reference*/
void UnesiMatricu(double **&MATRIX, int n){
for (int i=0; i<n; i++)
for (int j=0; j<n; j++)
{
cout << "[" << i+1 << ", " << j+1 << "]: ";
cin >> MATRIX[i][j];
}
}

/* Funkcija koja prima dvije kvadratne matrice n-tih dimenzija, i sabire ih
tako sto razultat sabiranja pripisuje u prvu od dvije matrice, vraca se
pokazivac na prvu matricu */
/* Kreiranje trece dinamicke matrice smo izbjegli, da stedimo memoriju ;) */
double **SaberMatrice(double **MATRIX1, double **MATRIX2, int n){
for (int i=0; i<n; i++) for (int j=0; j<n; j++) MATRIX1[i][j]+=MATRIX2[i][j];
return MATRIX1;
}

/* Funkcija prima matricu i dimenzije matrice i ispisuje matricu na ekran */
void IspisiMatricu(double **MATRIX, int n){
for(int i=0; i<n; i++){
for(int j=0; j<n; j++){
cout.width(5);
cout << MATRIX[i][j];
}
cout << endl;
}
}
```

```
*** C * C * * O * M * G * C * C * G * M * O * * C * C * * *
* * C C * * R G A M N * * + * * N M A G R * * C C * *
* C * * P * R * I * * * + * * * I * R * P * * C *
```

Autor: Murtic Nirmel
www.etf.ba

```
int main(){

// Definisane tri dvostruka pokazivaca koji ce kasnije predstavljati matrice
double **MATRIX1=0, **MATRIX2=0, **MATRIX3=0;

// Definisane dimenzije matrice
int n;
try{
cout << "Unesi broj kolona/redova kvadratne matrice (samo jedan broj) : ";
cin >> n;
KreirajMatricu(MATRIX1, n);
KreirajMatricu(MATRIX2, n);
UnesiMatricu(MATRIX1, n);
UnesiMatricu(MATRIX2, n);
MATRIX3=SaberMatrice(MATRIX1, MATRIX2, n);
cout << "Zbir matrica: " << endl;
IspisiMatricu(MATRIX3, n);
}

// Ako je bacen izuzetak ispis error poruke...
catch(...) {
cout << "Problem: Nema dovoljno memorije za smjestanje matrice u memoriju!\n";
}

// Brisu se kreirane matrice...
UnistiMatricu(MATRIX1, n);
UnistiMatricu(MATRIX2, n);
UnistiMatricu(MATRIX3, n);

// Kraj programa
getchar();
return 0;
}
```

```
***C*C*O*M*G*C*C*G*M*O*C*C**
**C*C**R*G*A*M*N**++**N*M*A*G*R**C*C**
*C**P*R*I**++**I*R*P**C*
```

Autor: Murtic Nirmel
www.etf.ba

72. Napišite program koji traži od korisnika da unese niz rečenica, pri čemu se broj rečenica prethodno unosi sa tastature. Za svaku unesenu rečenicu dinamički alocirajte prostor, uz vođenje evidencije o adresi svake alocirane rečenice u dinamičkom nizu pokazivača na početke svake od rečenica. Nakon toga, treba ispisati unesene rečenice sortirane u abecedni poredak (odnosno u poredak po ASCII kodovima). Sortiranje obavite ručno, bilo kojim postupkom koji Vam je poznat (drugim riječima, nemojte koristiti gotove funkcije za sortiranje, poput funkcije **"sort"** iz biblioteke **"algorithm"**).

RJEŠENJE:

```
#include <iostream>
#include <string>
using namespace std;
int main(){
// Unos broja recenica
int broj_rec;
cout << "Unesite broj recenica: ";
cin >> broj_rec;
cin.ignore(10000, '\n'); // Ignorisanje nepotrebnih znakova iz bufera
char **recenice = new char*[broj_rec]; // Definisanje dvostrukog pokazivaca

for(int i=0; i<broj_rec; i++) { // For petlja u kojoj se unosi n recenica

// Definisanje pomocnog niza pom i unos recenice u njega
char pom[1000];
cout << i+1 << ": ";
cin.getline(pom, 1000);

// Definisanje niza na niz (matrica, i to grbava) (pomocu pokazivaca)
recenice[i] = new char[strlen(pom) + 1];

// Upisivanje recenice iz pom na i-to mjesto u nizu
strcpy(recenice[i], pom);
}

// Pomocni stringovi, uveo sam ih zbog efikasnije provjere operacija < >
string pom1, pom2;
// Proces sortiranja, metodom nalazenja najveceg i stavljanje na vrh
for (int i=0;i<broj_rec-1;i++) for (int j=i+1;j<broj_rec;j++){
pom1=recenice[i];
pom2=recenice[j];
if (pom1>pom2){
char pomocna[1000];
strcpy(pomocna, recenice[i]);
strcpy(recenice[i], recenice[j]);
strcpy(recenice[j], pomocna);
}}

// Ispis sortiranog niza
cout << endl << "Sortirano: " << endl;
for(int k = 0; k < broj_rec; k++) cout << k+1 << ": " << recenice[k] << endl;
// Kraj programa
getchar();
return 0;
}
```

```
***C**C**O**M**G**C**C**G**M**O**C**C**
**C**C**R**G**A**M**N**+**+**N**M**A**G**R**C**C**
**C****P**R**I**+**+**I**R**P**C**
```

Autor: Murtic Nirmel
www.etf.ba

73. Napišite funkcije "TragMatrice", "DaLiJeGornjaTrougaona" i "Transponovana". Sve tri funkcije primaju jedan parametar "A", koji predstavlja matricu realnih brojeva. Funkcija "TragMatrice" treba da kao rezultat vrati trag matrice (tj. sumu elemenata na glavnoj dijagonali) ukoliko je matrica kvadratna, a da baci izuzetak ukoliko nije (s obzirom da se trag definira samo za kvadratne matrice). Funkcija "DaLiJeGornjaTrougaona" treba da kao rezultat vrati logičku vrijednost "true" ukoliko je matrica gornja trougaona (tj. ako ima sve nule ispod glavne dijagonale), a "false" ako nije. Konačno, funkcija "Transponovana" treba da kao rezultat vrati transponovanu matricu u odnosu na matricu koja joj je proslijeđena kao argument. Napisane funkcije testirajte u glavnom programu na primjeru matrice čije dimenzije i elemente unosi korisnik putem tastature.

RJEŠENJE:

```
# include <iostream>
# include <vector>

using namespace std;

// Funkcija DaLiJeGornjaTrougaona...
bool DaLiJeGornjaTrougaona(vector<vector<double> > a){
    bool istina=true;
    for (int i=1; i<a.size(); i++)
        for (int j=0; j<i; j++)
            if (a[i][j]!=0) istina=false;
    return istina;
}

// Funkcija TragMatrice...
double TragMatrice(vector<vector<double> > a){
    if (a.size()!=a[0].size())
        throw "Matrica nije kvadratna, nema traga i ne moze biti trougaona!\n";
    double rez=0;
    for (int i=0; i<a.size(); i++) rez+=a[i][i];
    return rez;
}

// Funkcija Transponovana...
vector<vector<double> > Transponovana(vector<vector<double> > a){
    int pom1=a.size(); // Pocetni broj redova matrice
    int pom2=a[0].size(); // Pocetne broj kolona matrice
    // max - veca dimenzija (red ili kolona)
    int max=a[0].size();
    if (a.size(>a[0].size()) max=a.size();
    // Resize matrice na kvadratnu dimenzija [max][max]
    a.resize(max);
    for(int i = 0; i < max; i++) a[i].resize(max);

    // "Okretanje" odgovarajucih pozicija u matrici
    double priv;
    for (int i=0;i<max;++i)
        for (int j=i+1;j<max;++j)
```

```
***C*C**O*M*G*C*C*G*M*O**C*C**
**C*C**R*G*A*M*N**++**N*M*A*G*R**C*C**
*C**P*R*I**++**I*R*P**C*
```

Autor: Murtic Nirmel
www.etf.ba

```
{
priv = a[i][j];
a[i][j] = a[j][i];
a[j][i] = priv;
}

// Resize kolona na pocetne redove
a.resize(pom2);

// Resize redova na pocetne kolone
for(int i = 0; i < pom2; i++) a[i].resize(pom1);
return a;
}

int main(){
try {
// Unos dimenzija matrice
int r, k;
cout << "Unesite broj redova matrice: "; cin >> r;
cout << "Unesite broj kolona matrice: "; cin >> k;
// Definisanje matrice a
vector<vector<double> > a(r, vector<double>(k));
// Unos clanova matrice a
for (int i=0; i<r; i++)
for (int j=0; j<k; j++)
{
cout << "[" << i+1 << "," << j+1 << "]: ";
cin >> a[i][j];
}
// Ispis transponovane matrice
cout << endl << "Transponovana matrica: " << endl;
vector<vector<double> > tr=Transponovana(a);
for(int i=0; i<tr.size(); i++){
for(int j=0; j<tr[0].size(); j++) {
cout.width(5);
cout << tr[i][j];
}
cout << endl;
}
// Ispis traga matrice
double trag=TragMatrice(a);
cout << endl << "Trag matrice je: " << trag;
// Ispis poruke da li je matrica gornja trougaona ili ne
bool istina=DaLiJeGornjaTrougaona(a);
if (istina) cout << endl << "Matrica je gornja trougaona!"; else
cout << endl << "Matrica nije gornja trougaona!";
// Kraj try bloka
}
// catch blok
catch(const char poruka[]) {
cout << endl << poruka << endl;
}
// Kraj programa
return 0;
}
```

```
* * * C * C * * O * M * G * C * C * G * M * O * * C * C * * *
* * C C * * R G A M N * * + * * N M A G R * * C C * *
* C * * P * R * I * * * + * * * I * R * P * * C *
```

Autor: Murtic Nirmel
www.etf.ba

74. Napišite program koji prvo traži da se sa tastature unese rečenica, a zatim ispisuje svaku riječ te rečenice u posebnom redu, kao i informaciju koliko ta rečenica ima slova (ne računajući interpunkcijske znakove) i riječi. Obratite pažnju da riječi rečenice mogu biti razdvojene sa više od jednog razmaka, kao i da se na samom početku i na samom kraju rečenice mogu također nalaziti razmaci. Za realizaciju koristite tip "string" i odgovarajuće manipulacije sa stringovima.

RJEŠENJE:

```
# include <iostream>
# include <string>

using namespace std;
int main(){

// Unos recenice
cout << "Unesi recenicu: " << endl;
string recenica;
getline(cin, recenica);

// Ispis rijeci
cout << endl << "Rijeci iz recenice: " << endl << endl;
bool istina=false;

int br=0, brl=0;
for (int i=0; i<=recenica.length(); i++) {
    if (((recenica[i]>='a')&&(recenica[i]<='z'))||
        ((recenica[i]>='A')&&(recenica[i]<='Z'))
        { cout << recenica[i]; br++; istina=true; continue; }
    if (istina) {cout << endl; istina=false; brl++; }
}

// Ispis broja slova
cout << endl << "Recenica ima " << br << " slova i " << brl << " rijeci!";
// Kraj programa
return 0;
}
```

```
***C* C* *O* M* G* C* C* G* M* O* *C* C* * *
* * C* C* * R* G* A* M* N* * * + * * N* M* A* G* R* * * C* C* * *
* C* * * P* * R* * I* * * * + * * * I* * R* * P* * * C* *
```

Autor: Murtic Nirmel
www.etf.ba

75. Napišite funkciju "DaLiJePalindrom" koja za string koji joj je proslijeđen kao parametar ispituje da li je palindrom ili nije, i kao rezultat vraća odgovarajuću logičku vrijednost "true" ili "false". Pod palindromima smatramo riječi ili rečenice koje se isto čitaju sa obe strane (npr. "kapak"). Prilikom ispitivanja treba ignorirati eventualne razmake, interpunkcijske znake i razliku između velikih i malih slova, tako da rečenica "Ana voli Milovana" treba da bude prepoznata kao palindrom, iako bukvalno pročitana sa suprotnog kraja glasi "anavoliM ilov Ana". Još neki od poznatih palindromnih rečenica su "Udovica baci vodu", "E, sine, ženi se", "I jogurt ujutru goji", "I Dara za mast sama zaradi", "Jovi limaru gumu gura Milivoj", itd. Historijski najpoznatiji palindrom je latinska rečenica "Sator Arepo tenet opera rotas" za koju se u srednjem vijeku smatralo da ima magična svojstva. Napišite i kratki testni program ("main" funkciju) u kojem ćete testirati napisanu funkciju.

RJEŠENJE:

```
# include <iostream>
# include <string>

using namespace std;

// Funkcija DaLiJePalindrom...
bool DaLiJePalindrom(string a){
bool istina=true;
string b;
// Kopiranje slova i brojeva u niz b(char) izbacivanjem ostalih znakova...
for (int i=0; i<=a.length(); i++)
    if (((a[i]>='a')&&(a[i]<='z'))||
        ((a[i]>='A')&&(a[i]<='Z')))
        { if((a[i]>='a')&&(a[i]<='z')) a[i]+='A'-'a'; b+=a[i]; }
// Provjera da li je rijec "simetricna", tj. da li je palindrom...
for (int i=0, j=b.size(); i<b.size(); i++,j--) if (b[i]!=b[j-1]) istina=false;
return istina;
}

int main(){

// Unos recenice...
cout << "Unesi rijec ili recenicu: " << endl;
string recenica;
getline(cin, recenica);

// Poziv funkcije...
bool istina=DaLiJePalindrom(recenica);

// Ispis rezultata...
if (istina) cout << endl << "Palindrom je!!!";
else cout << endl << "Nije palindrom!!!";

// Kraj programa...
return 0;
}
```

```
* * * C * C * * O * M * G * C * C * G * M * O * * C * C * * * *
* * C C * * R G A M N * * + * * N M A G R * * C C * *
* C * * P * R * I * * * + * * * I * R * P * * C *
```

Autor: Murtic Nirmel
www.etf.ba

76. Napisati program koji traži od korisnika da unese spisak riječi (broj riječi se prethodno unosi sa tastature), a zatim ispisuje na ekran prvu i poslednju riječ iz spiska po abecednom poretku, kao i popis svih unesenih riječi, ali bez ispisivanja duplikata (tj. bez ispisivanja riječi koje su se već jednom ispisale). Program realizirati korištenjem vektora stringova, odnosno vektora čiji su elementi tipa **"string"**.

RJEŠENJE:

```
# include <iostream>
# include <string>
# include <vector>

using namespace std;

main(){

// Unos broja rijeci a zatim unos rijeci u vektor...
vector<string> a;
int n;
cout << "Unesi broj rijeci: "; cin >> n;
for (int i=0; i<n; i++){
cout << i+1 << ". ";
string pom;
cin >> pom;
a.push_back(pom);
}

// Sortiranje imena
string pom;
for (int i=0; i<(a.size()-1); i++)
for (int j=i+1; j<a.size(); j++)
if (a[j]<a[i]){
pom=a[i];
a[i]=a[j];
a[j]=pom;
}

// Ispis prve i zadnje rijeci
cout << "Prva rijec je: " << a[0] << " a zadnja " << a[a.size()-1] << endl;

// Ispis svih rijeci
cout << endl << "Ispis rijeci: " << endl;
a.push_back("");
for (int i=0, j=0; i<a.size()-1; i++){
if ((a[i]!=a[i+1])) { cout << j+1 << ". " << a[i] << endl; j++; }
}

// Kraj programa
return 0;
}
```

77. Iz numeričke matematike je poznato da se određeni integral neke funkcije $f(x)$ na intervalu (a, b) može približno izračunati uz pomoć tzv. *Simpsonovog pravila*, prema kojem je:

$$\int_a^b f(x) dx \approx \frac{h}{3} \left[f(a) + 4 \cdot \sum_{(k=1,3,5\dots)}^{n-1} f(a+k \cdot h) + 2 \cdot \sum_{(k=2,4,6\dots)}^{n-2} f(a+k \cdot h) + f(b) \right]$$

gdje je n broj podintervala na koji dijelimo interval (a, b) i koji mora biti paran (veći broj podintervala daje veću tačnost računanja), a h je dužina svakog podintervala, tj. $h=(b-a)/n$. Napišite funkciju "Integral" koja prima kao parametre f , a , b i n (f je funkcija čiji se integral računa) a koja kao rezultat daje približnu vrijednost integrala. Napisanu funkciju testirajte na primjerima integrala funkcije $\sin x$ na intervalu $(0, \pi)$, zatim funkcije x^3 na intervalu $(0, 10)$, i funkcije $1/x$ na intervalu $(1, 2)$. Testiranje izvršite za različite vrijednosti n i uporedite rezultate sa tačnim rezultatima. Utvrdite kolike su vrijednosti za n bile potrebne da se dobije rezultat tačan na 5 decimala za sva tri primjera.

RJEŠENJE:

```
#include <iostream>
#include <cmath> // zbog sin funkcije
using namespace std;

// Funkcija od koje trazimo integral x*x*x
double xnatrecu(double x){
    return x*x*x; // probajte sin(x) i 1/x takodjer
}

/* Funkcija prima kao parametar funkciju xnatrecu, te tri realna broja */
/* - a pocetak intervala; b - kraj intervala; n - preciznost */
double Integral(double F(double), double n, double a, double b){
    double dio1, dio2, h=(b-a)/n; // h - duzina intervala
    for(int k=1; k<=n-1; k+=2) dio1+=F(a+k*h); // racunanje prve sume
    for(int k=2; k<=n-2; k+=2) dio2+=F(a+k*h); // racunanje druge sume
    return (h/3)*(F(a)+4*dio1+2*dio2+F(b)); // Formula :)
}

int main(){
    double a,b,n;
    // Unos podataka
    cout << "Unesi broj podintervala (n): "; cin >> n;
    cout << "Unesi interval (a,b): "; cin >> a >> b;
    // Ispis rezultata
    cout << "Rezultat: " << Integral(xnatrecu, n, a, b);
    getchar();
    return 0;
}
```

```
***C*C*O*M*G*C*C*G*M*O*C*C**
* *C C * *R G A M N * * + * * N M A G R * * C C * *
* C * * P * R * I * * * + * * * I * R * P * * C *
```

Autor: Murtic Nirmel
www.etf.ba

78. Koristeći odgovarajuće funkcije iz biblioteke "algorithm", napišite program koji će za niz cijelih brojeva unesenih sa tastature ispisati:

- najveći i najmanji element niza;
- koliko puta se u nizu pojavljuje najmanji element;
- koliko u nizu cijelih brojeva unesenih sa tastature ima brojeva koji su potpuni kvadrati (tj. kvadrati nekog drugog cijelog broja).

Nakon toga, program treba prepisati u drugi niz sve elemente koji nisu potpuni kvadrati, i ispisati elemente tako formiranog niza.

Napomena: U programu nije uopće dozvoljeno koristiti petlje (osim za unos elemenata niza), već sve manipulacije treba ostvariti isključivo pozivima funkcija iz biblioteke "algorithm".

RJEŠENJE:

```
#include <iostream>
#include <algorithm>
#include <cmath>

using namespace std;

// Funkcija koja vraca true ako je broj puni kvadrat, u suprotnom false
bool jelkvadrat(int a){
    if(a==sqrt(a)*sqrt(a)) { return true; } else return false;
}

// Funkcija koja ispisuje broj koji prima
void Ispisi(int x) {
    cout << x << " ";
}

int main(){
    int A[100], B[100]; // Dva niza cijelih brojeva
    int n; // Broj clanova
    int brojac=0; // Broj clanova koji su puni kvadrati

    // Unos niza
    cout << "Unesite broj clanova niza: "; cin >> n;
    for (int i=0; i<n; i++){
        int br;
        cout << i+1 << ": ";
        cin >> br;
        A[i]=br;
        /* Ako je broj puni kvadrat onda se brojac uvecava, u suprotnom se upisuje u
        niz B[] */
        if(jelkvadrat(br)) brojac++; else B[i-brojac]=br;
    }
}
```

```
*** C * C * * O * M * G * C * C * G * M * O * * C * C * * *
* * C * C * * R * G * A * M * N * * * + * * * N * M * A * G * R * * * C * C * * *
* C * * P * R * I * * * + * * * I * R * P * * C *
```

Autor: Murtic Nirmel
www.etf.ba

```
// Ispis rezultata
cout << "Najmanji element: " << *min_element(A, A+n) << " a najveći: " <<
*max_element(A, A+n) << endl;

cout << "Najmanji element se ponavlja " << count(A, A+n, *min_element(A, A+n))
<< " puta." << endl;

cout << "Imamo " << brojac << " punih kvadrata!!!" << endl;

cout << "Niz brojeva koji nisu puni kvadrati: ";
for_each(B, B+(n-brojac), Ispisi);

// Kraj programa
getchar();
return 0;
}
```

```
* * * C * C * * O * M * G * C * C * G * M * O * * C * C * * *
* * C * C * * R G A M N * * * + * * * N M A G R * * C * C * *
* C * * P * R * I * * * + * * * I * R * P * * C *
```

Autor: Murtic Nirmel
www.etf.ba

79. Napišite generičku funkciju "Izvrni" koja prihvata iste parametre i obavlja isti zadatak kao funkcija "reverse" iz biblioteke "algorithm". Funkciju treba realizirati isključivo korištenjem pokazivačke aritmetike, bez korištenja indeksiranja i njegove trivijalne simulacije (tj. pisanja " $*(p+i)$ " umjesto " $p[i]$ "). Napišite i kratki testni program u kojem ćete testirati napisanu funkciju na dva fiksna niza, od kojih je jedan niz cijelih brojeva, a drugi niz znakova.

RJEŠENJE:

```
#include <iostream>
#include <algorithm>

using namespace std;

// Generička funkcija izvrni
template<typename TIP>
void Izvrni(TIP *p1, TIP *p2){

    int br=0; // br - broj članova niza

    p2--; // Pokazivac p2 se vrati na zadnji član

    while(*p1!=*p2) { p1++; br++; } // Racuna se dužina niza

    p1-=br; // Pokazivac p1 se vrati gdje je i bio :)

    // For petlja koja okreće članove niza
    for(int i=0; i<br/2; i++){
        TIP pr=*p1;
        *p1=*p2;
        *p2=pr;
        *p1++;
        *p2--;
    }
}

// Funkcija za ispis broja
void Ispisi(int x) {
    cout << x << " ";
}

int main(){
    int A[100]; // Definisavanje niza
    int n; // Broj članova niza
    // Unos niza
    cout << "Unesite broj članova niza: "; cin >> n;
    for (int i=0; i<n; i++){
        int br;
        cout << i+1 << ": ";
        cin >> br;
        A[i]=br;
    }
}
```

```
* * * C * C * * O * M * G * C * C * G * M * O * * C * C * * *
* * C C * * R G A M N * * + * * N M A G R * * C C * *
* C * * P * R * I * * * + * * * I * R * P * * C *
```

Autor: Murtic Nirmel
www.etf.ba

```
// Poziv funkcije koja izvrce niz
Izvrni(A, A+n);

// Ispis izvrnutog niza
cout << "Izvrnuti niz: ";
for_each(A, A+n, Ispisi);

// Kraj programa
getchar();
return 0;
}
```

```
***C* C* *O* M* G* C* C* G* M* O* *C* C* * *
* * C C * * R G A M N * * + * * N M A G R * * C C * *
* C * * P * R * I * * * + * * * I * R * P * * C *
```

Autor: Murtic Nirmel
www.etf.ba

80. Napišite program koji će na primjeru jednog fiksnog sortiranog niza od 50 elemenata i broja "n" unesenog sa tastature ispisati poziciju gdje se broj "n" nalazi unutar niza, odnosno činjenicu da broj "n" nije pronađen unutar niza. Koristite binarnu pretragu i odgovarajuće funkcije iz biblioteke „algorithm“.

RJEŠENJE:

```
#include <iostream>
#include <algorithm>

using namespace std;

int main(){
int n, istina=1;

// Definisavanje i popunjavanje niza ciframa od 50 do 1 (samo primjer)
int A[50];
for (int i=0;i<=50;i++) A[i]=50-i;

// Unos broja n
cout << "Unesite broj n: "; cin >> n;

// Ispis pozicije na kojoj se broj nalazi, ako se nalazi...
for (int i=0; i<=50; i++){
if (binary_search(A+(i-1), A+i, n))
{ cout << "Na poziciji: " << i << endl; istina=0; }
}
if (istina) cout << "Nema tog broja u nizu!!!" << endl;

// Kraj programa
getchar();
return 0;
}

// Rjesenje je glupo do bola kao i zadatak, pokusajte drugacije :)
```