

SKRIPTA IZ PROGRAMSKOG JEZIKA C

Uvod

Programski jezik C razvijen je u AT&T Bell laboratories, s ciljem da se stvori programski jezik koji će omogućiti brzo i efikasno programiranje. S obzirom da se ta firma bavila sistemskim programiranjem, novi programski jezik trebao je biti na dovoljno niskoj razini da riješi probleme koji su se do tada uglavnom rješavali u ASSEMBLER-u, ali i na dovoljno visokoj razini da bi se iskoristile sve prednosti jezika više razine. Da su u svojoj namjeni uspjeli, pokazuje niz programskih paketa pisanih u programskom jeziku C (OS UNIX)

Glavne postavke C-a postale su dostupne širem krugu korisnika 1978. godine, a danas se slobodno može reći da je C programski jezik br. 1.

U C program se mogu direktno uključivati naredbe pisane u ASSEMBLER-u, što (zajedno s mogućnošću direktnog pristupa pojedinim bitovima, bajtovima i blokovima memorije) C čini vrlo pogodnim za pisane sistemskih programa.

Programski jezik C sadrži sve naredbe neophodne za strukturirano programiranje (if, case, petlju...)

Osnovne karakteristike TURBO C-a

Programski paket TURBO C proizvod je programske kuće BORLAND.

U programski paket TURBO C uključen je uređivač teksta (EDITOR), prevodilac (COMPILER), program za povezivanje (LINKER) i program za otkrivanje i uklanjanje grešaka (DEBUGGER). Također postoji i mogućnost izvršavanja programa korak po korak, pa se u svakom trenutku mogu doznati vrijednosti pojedinih varijabli, što je pogodno u fazi ispitivanja logike programa.

2. Struktura programa u C-u

Najpogodniji način rješavanja programskih problema je rastavljanje problema u niz manjih, međusobno povezanih cjelina (strukturirano programiranje). Ovaj pristup rješavanja problema podržan je i u C-u. Programi u C-u sastoje se od niza međusobno povezanih funkcija. Broj funkcija nije ograničen. U programu je obavezna jedna i samo jedna funkcija - funkcija **main()**. Ona označava mjesto na kojem počinje izvršavanje programa. Za poziv funkcije dovoljno je navesti njezino ime. Sve funkcije koje se koriste u većini programa smještene su u biblioteke funkcija.

Nakon svake naredbe u C-u mora stajati znak ;, koji (kao i u Pascalu) označava kraj jedne i početak druge naredbe. Niz naredbi grupiranih u jednu logičku cjelinu predstavlja blok naredbi. Početak bloka označava se sa {, a završetak sa }.

Sve varijable koje se koriste u programu moraju se deklarirati, obično na početku programa ili funkcije, prije prve izvršne naredbe.

Tekst napisan između /* */ je komentar i ne utječe na tok izvršavanja programa.

Programski jezik C nije pozicijski (kao npr. FORTRAN), što znači da kod može početi u bilo kojoj koloni, te da se naredbe, izrazi i komentari mogu protezati u više programskih linija.

Primjer 1:

```
/*ucitati dva broja i ispisati veci (uvjetni operator)*/
#include <stdio.h>
#include <conio.h>
void main()
{
int a,b;
clrscr();
printf("\n Upisi dva broja: ");
scanf("%d,%d",&a,&b);
(a<b)? printf ("\n Veci je %d",b) : printf ("\n Veci je %d",a);
getch();
}
```

3. Editor turbo C-a

Ulaskom u direktorij u kojem je smješten programski jezik C (najčešće **BORLANDC**), potrebno je ući u direktorij **BIN**, te sa **bc** pokrenuti program. Nakon toga se na zaslonu pojavljuje tzv. osnovni zaslon, koji se sastoji od četiri dijela:

-osnovni izbor

-prozor za uređivanje teksta i status linija

-prozor za poruke prevodioca

-kratki pregled značenja vrućih tipki

Opcije osnovnog izbora aktiviraju se istovremenim pritiskom na tipku ALT i tipku početnog slova željene opcije, ili tipkom F10 i strelicama.

Izborom opcije **File** otvara se prozor s dodatnim izborom za rad s datotekama i direktorijima (new, open, save, save as, save all, change dir, print, dos shell, quit).

Standardne opcije za kopiranje, brisanje, undo, te show clipboard nalaze se u izborniku **Edit**.

Traženje i zamjena teksta neke su od opcija izbornika **Search**.

Opcija prevođenja i izvršavanje programa nalazi se u izborniku Run, a samo prevođenje u izborniku Compile.

U izborniku **Debug** potavljaju se razne opcije programa za pronalaženje i ispravljanje grešaka, a izbornik **Project** služi za aktiviranje i deaktiviranje programa čiji se kod nalazi u više datoteka. **Options** služi za postavljanje različitih opcija prevodioca i povezača, a **Window** za uređenje radnog okruženja.

4. Identifikatori, ključne riječi i tipovi podataka

Identifikatori su imena koja se dodjeljuju varijablama, funkcijama, i sl. Sastoje se od niza alfanumeričkih znakova (brojke i slova). Dva identifikatora su različita ako se razlikuju u prva 32 znaka. Prvi znak identifikatora mora biti slovo ili donja crta (_). TURBO C dozvoljava i upotrebu \$ kao jednog od znakova identifikatora, s tim da ne može biti na prvom mjestu. Identifikatori ne smiju biti ključne riječi programskog jezika C.

Programski jezik C razlikuje velika i mala slova, tako da su varijable varij1 i Varij1 dvije različite varijable.

Ključne riječi su identifikatori koji imaju točno određeno značenje. One, zajedno sa sintaksom programskog jezika, određuju što se i na koji način u određenom programskom jeziku može učiniti. Karakteristika programskog jezika C je relativno mali broj ključnih riječi (32+11)

Ključne riječi pišu se malim slovima.

Ključne riječi: auto, break, case, char, const, continue, default, do, double, else, enum, extern, float, for, goto, if, int, long, register, return, short, signed, sizeof, static, struct, switch, typedef, union, unsigned, void, volatile, while + asm, _cs, _ds, _es, _ss, cdecl, far, huge, interrupt, near, pascal.

Programski jezik C podržava pet osnovnih tipova podataka. To su: **znakovni, cjelobrojni, realni, realni dvostruke preciznosti i tip podataka koji ne sadrži vrijednost**. Ključne riječi koje ih određuju su: **char, int, float, double** i **void**. Tip podataka koji ne sadrži vrijednost svrstan je u osnovne tipove podataka jer se često koristi kod provjere tipova podataka, za definiranje funkcije bez argumenata i funkcija koje ne vraćaju nikakvu vrijednost.

Pravilnom izborom tipa podataka može se povećati efikasnost programa. Stoga tipove podataka treba zadavati ovisno o opsegu koji će određena varijabla poprimiti tokom izvođenja programa.

Osnovni tipovi podataka mogu se preciznije definirati pomoću modifikatora, koje ćemo spominjati kada se za to pojavi potreba.

Dozvoljene su gotovo sve vrste pretvaranja tipova podataka, a u izrazima su dozvoljene kombinacije različitih tipova podataka. Krajnji rezultat u kojem se nalaze različiti tipovi podataka ovisi o ugrađenim pravilima za pretvaranje tipova podataka, ili ga može zadati osobno programer.

Osim osnovnih postoji i nekoliko složenijih tipova podataka kao što su polja, strukture, polja struktura, unije i polja bitova. Omogućeno je i definiranje vlastitih tipova podataka.

Formati za učitavanje i ispis podataka:

Tip podataka	Format
cjelobrojni (int)	d
realni (float, double)	f
znakovni (char)	c
niz znakova (string)	s

5. Deklaracija varijabli, konstante i aritmetički operatori

Sve varijable koje se koriste u programu moraju se deklarirati. Deklaracijom varijable zadaje se **tip** i **ime** varijable. Opći oblik naredbe za deklaraciju je:

tip_podataka lista_varijabli;

Tip_podataka predstavlja jedan od osnovnih ili korisnički definiranih tipova podataka, a **lista_varijabli** predstavlja niz varijabli koje će u toku izvršavanja programa sadržavati vrijednosti zadanog tipa podataka, odvojenih zarezom.

Npr: `int a, b, c;`

Područje djelovanja određene varijable ovisi o mjestu na kojem je deklarirana. Deklaracija varijabli u C-u može biti izvan svih funkcija, na početku funkcije, deklaracija argumenata funkcije, te na početku bloka naredbi. Globalne varijable deklariraju se izvan svih funkcija i mogu se koristiti u cijelom programu, dok se lokalne varijable deklariraju na početku funkcije ili se deklariraju kao argumenti funkcije i dostupne su samo u toj funkciji (ili bloku naredbi). Izvan funkcije ili bloka lokalne varijable ne postoje. Imena lokalnih varijabli u različitim funkcijama (blokovima) mogu biti ista.

Definicijom globalne varijable automatski joj se dodjeljuje početna vrijednost 0, dok lokalne varijable ne sadrže početnu vrijednost i moraju se inicijalizirati posebnom naredbom nakon deklaracije ili tokom deklaracije, npr:

```
int a=0;    ili   int a;
                a=0;
```

Neinicijalizirane lokalne varijable sadrže neodređene vrijednosti.

Konstante su vrijednosti koje se zadaju jednom i ne mijenjaju se tokom izvođenja programa. S obzirom na tipove podataka razlikuju se cjelobrojne, realne i znakovne konstante. Znakovne konstante zadaju se unutar znaka '' i uglavnom se sastoje od jednog znaka, no postoje i znakovne konstante koje se sastoje od dva znaka. One počinju znakom \ i imaju posebno značenje. Koriste se za kontrolu ispisa. Neke od spomenutih konstanti, koje se češće koriste su:

```
\f   nova stranica
\n   novi red
\r   povratak na početak linije
```

Osim navedenih tipova konstanti programski jezik C definira i konstantu niza znakova, koja se sastoji od znakova unutar para navodnika, npr. "konstanta".

Aritmetički operatori

```
zbrajanje +
oduzimanje -
množenje *
dijeljenje /
modulo     % (ostatak cjelobrojnog dijeljenja)
```

Prioritet: modulo, množenje i dijeljenje, pa zbrajanje i oduzimanje. Kod operatora s istim prioritetom tok izvršavanja ide s lijeva na desno. Prioritet izvršavanja operacija može se mijenjati upotrebom zagrada.

Operatori inkrement i dekrement

Programski jezik C uvodi dva nova aritmetička operatora: inkrement (++) i dekrement (--), odnosno: uvećaj za jedan i umanj za jedan. Oba operatora se mogu pisati ispred ili iza varijable.

Primjer: `x=x+1; y=y-1;` je isto što i `x++; y--;` odnosno `++x; --y;`

Međutim, korištenjem navedenih operatora u izrazima stvari se mijenjaju. O položaju operatora u odnosu na varijablu ovisi kada će se vrijednost varijable povećati (ili smanjiti).

Znači, umjesto `y=x+1;` možemo pisati `y=x++;` ili `y=++x;`

Neka je `x=2`. Nakon izraza `y=++x;` vrijednost varijabli `x` i `y` je 3 (vrijednost varijable `x` se uveća za 1, pa se ta vrijednost pridruži varijabli `y`), a nakon izraza `y=x++;` vrijednost varijable `y` je 2, a vrijednost varijable `x` je 3 (najprije se trenutna vrijednost varijable `x` pridruži varijabli `y`, a zatim se vrijednost varijable `x` uveća za 1).

Zadaci za vježbu:

1. Izračunaj vrijednost izraza: `a+b*c-a%b` ako je:

- `a=2, b=3, c=6` Rj: 18
- `a=13, b=10, c=5` Rj: 60

2. Ako je vrijednost varijable $a=5$, kolike će biti vrijednosti varijabli a i b nakon izračunavanja izraza $b=a++$, a kolike nakon izračunavanja $b=++a$?

Rj:

a) $a=6$, $b=5$

b) $a=6$, $b=6$

6. Operator pridruživanja i pretvorba tipova podataka

Operator pridruživanja u programskom jeziku C je $=$. Kao što znamo, njegova je uloga da vrijednost desne strane izraza pridruži varijabli na lijevoj strani izraza. Operator pridruživanja ima najniži prioritet izvršavanja.

Opći oblik naredbe za pridruživanje vrijednosti je **varijabla=izraz;**

Operator pridruživanja se može naći u kombinaciji s aritmetičkim operatorima, pa takve kombinacije čine jedinstvene konstrukcije aritmetičkih izraza karakteristične za programski jezik C, koje na prvi pogled izgledaju nejasno, no, ako se ima u vidu **redosljed izvršavanja operacija pridruživanja koji je s desna na lijevo** uočava se da takav način zapisa predstavlja skraćeni oblik pisanja aritmetičkih izraza.

Primjer: $a+=2$; isto je što i $a=a+2$;

$a-=3$; znači $a=a-3$;

$a/=10$; je $a=a/10$;

$a\%=3$; je $a=a\%3$;

$a*=10+b$; znači $a=a*10+b$;

$a+=++b+20$; je $a=a+b+1+20$;

Pretvorba tipova podataka

Tip rezultata aritmetičkih izraza gdje su operandi istog tipa odgovara tipu operanda. Npr. ako su u izrazu $a+b$, a i b cjelobrojne varijable, tada će i rezultat biti cjelobrojnog tipa.

U toku izračunavanja izraza u kojima se nalaze različiti tipovi podataka, ti se tipovi svode na isti tip prema ugrađenim pravilima za pretvaranje tipova podataka koja su sastavni dio programskog jezika C i orjentirana su prema višem tipu podataka.

Tako vrijedi:

1. Svi podaci `char` i `short` pretvaraju se u `int`

2. Ako u izrazu postoji bilo koji operand `double`, tada se svi podaci pretvaraju u `double` i rezultat je tipa `double`.

3. Ako u izrazu postoji bilo koji operand tipa `long`, tada se svi podaci pretvaraju u `long` i rezultat je tipa `long`.

4. Ako je bilo koji od operanada u izrazu deklariran kao `unsigned`, svi ostali operandi pretvaraju se u `unsigned` i rezultat je tipa `unsigned`.

5. Ako su svi operandi tipa `int`, tada je i rezultat tipa `int`.

Korištenje operatora pridruživanja također pokreće automatsko pretvaranje podataka, npr: ako je i cjelobrojna varijabla (`int`), a j realna (`float`), izrazom $j=i$ cjelobrojna vrijednost varijable i najprije će se pretvoriti u realnu, a zatim pridružiti varijabli j . Isto tako, u izrazu $i=j$; varijabli i će se pridružiti cjelobrojna vrijednost varijable j dobivena odbacivanjem decimalnog dijela.

7. Učitavanje i ispis podataka

Programski jezik C nema ugrađene ulazno/izlazne funkcije, već se ulaz i izlaz podataka obavljaju preko funkcija koje se nalaze u standardnoj ulazno/izlaznoj biblioteci. Programi koji koriste standardne ulazno/izlazne funkcije morju sadržavati liniju

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

koja se obično nalazi na početku programa. U datoteci stdio.h nalaze se sve U/I funkcije.

Standardna funkcija za **ispis** podataka je **printf()**

Ispis niza znakova: **printf ("niz znakova");**

Primjer 1:

```
#include<stdio.h>
main()
{
printf ("Od danas programiramo u C-u");
}
```

Program se pokreće kombinacijom tipki **CTRL-F9** ili preko izbornika birajući opciju **Run→Run**. @elimo li samo provjeriti ima li naš program grešaka (naravno, ne logičkih), biramo **Compile→Compile (ALT-F9)**

Nakon izvođenja programa C se vraća u prozor u kojem je kod programa, a prozor sa rezultatima ostaje nevidljiv. @elimo li vidjeti rezultate izvođenja programa treba pritisnuti kombinaciju tipki ALT-F5.

Zadatak:

Ispisati na ekran svoje ime, prezime i adresu, svaki podatak u novi red.

Ispis vrijednosti varijabli:

printf("lista formata",lista varijabli);

npr: ispisati vrijednosti varijabli a i b koje su cjelobrojnog tipa.

```
printf("%d %d",a,b);
```

želimo li da u ispisu varijable budu odvojene zarezom pisat ćemo:

```
printf("%d, %d",a,b);
```

Dakle, izgled ispisa na zaslonu uređuje se unutar dvostrukih navodnika, a u listi varijabli dolazi samo popis varijabli. Redosljed navođenja formata mora odgovarati redosljedu navođenja imena varijabli. Pogrešno zadan format u odnosu na deklaraciju varijable rezultirat će pogrešnim ispisom.

Primjer: izvođenjem naredbe

```
printf("\n rezultat izraza %d+%d je %d", a,b,c)
```

ako je a=2, b=3 na zaslon će se ispisati:

rezultat izraza 2+3 je 5

/n znači da će se prije ispisa prijeći u novi red.

Funkcije za učitavanje:

Funkcija za učitavanje jednog znaka s tipkovnice: **getch()** (njezin prototip nalazi se u datoteci **conio.h**).

Primjer:

```
#include<stdio.h>
#include<conio.h>
main()
{
printf ("Od danas programiramo u C-u");
getch();
}
```

Ovakvom upotrebom funkcije getch() postiže se to da nakon izvršavanja posljednje naredbe ispred funkcije getch program čeka učitavanje znaka s tipkovnice, odnosno ekran sa rezultatima izvođenja programa ostaje vidljiv do pritiska bilo koje tipke.

Funkcija za formatirani unos podataka je **scanf()**.

scanf("lista formata", lista varijabli);

kao i kod funkcije printf() u listi formata upisuju se formati za tip varijabli, a u listi varijabli popis imena varijabli istim redoslijedom kao u popisu formata. S obzirom da se u C-u varijable koje se nalaze u popisu varijabli prenose putem adrese, osim imena varijable treba dodati i adresni operator & ispred imena, npr:

```
scanf("%d,%d", &a,&b);
```

Napomena: Ako su oznake formata odvojene zarezom, prilikom učitavanja vrijednosti varijabli potrebno ih je također odvojiti zarezom. Ako nema zareza vrijednosti se odvajaju razmakom.

Primjer: Za zadane duljine stranica pravokutnika ispisati opseg i površinu danog pravokutnika.

```
#include <stdio.h>
#include <conio.h>
main ()
{
float a,b,p,o;
clrscr();
printf("\n upisi duljine stranica a i b \n");
scanf("%f%f", &a, &b);
o=2*(a+b);
p=a*b;
printf("\n Povrsina pravokutnika iznosi %.2f",p);
printf("\n Opseg pravokutnika iznosi %.2f",o);
getch();
}
```

8. Jednostavni programi-vježba

1. Učitaj dva cijela broja. Izračunaj i ispiši njihov zbroj, razliku i umnožak.
2. Učitaj troznamenkasti broj i ispiši njegove znamenke.
3. Napiši program koji rješava dvije jednačbe s dvije nepoznanice:
 $ax+by=e$, $cx+dy=f$ (a, b, c, d, e i f se učitavaju)

Rješenja:

1.

```
#include<stdio.h>
#include<conio.h>
int a,b,s,r,u;
main ()
{
clrscr();
printf("\n upisi brojeve a i b: ");
scanf("%d,%d", &a, &b);
s=a+b;r=a-b;u=a*b;
printf("\n zbroj brojeva %d i %d je %d, razlika %d, a umnozak %d",a,b,s,r,u);
getch();
}
```

```

2./*ucitava troznamenkasti broj i ispisuje njegove znamenke*/
#include<stdio.h>
#include<conio.h>
int xyz,x,y,z;
main()
{
clrscr();
printf("\n upisi troznamenkasti broj: ");
scanf("%d",&xyz);
x=xyz/100;
y=(xyz/10)%10;
z=xyz%10;
printf("\n znamenke broja %d su %d, %d, %d", xyz,x,y,z);
getch();
}

```

```

3.
#include<stdio.h>
#include<conio.h>
float a,b,c,d,e,f,x,y,z;
main()
{
clrscr();
printf("\n upisi koeficijente sistema dvije jednadzbe s dvije nepoznanice:");
printf("\n a:");scanf("%f",&a);
printf(" b:");scanf("%f",&b);
printf(" e:");scanf("%f",&e);
printf(" c:");scanf("%f",&c);
printf(" d:");scanf("%f",&d);
printf(" f:");scanf("%f",&f);
z=(f-c*e)/(a*d-b*c);
x=b*z;
y=a*z;
printf("\n rjesenja danog sistema su: x=%.2f, y=%.2f",x,y);
getch();
}

```

9. Jednostavni programi-vježba

Zadaci:

1. Napiši program koji će za zadani polumjer kruga ispisati njegov opseg i površinu. (zad 3, str 7.)

Uputa: $o = 2r\pi$, $P = r^2\pi$

2. Zadane su dvije točke u pravokutnom koordinatnom sustavu. Ispisati njihovu udaljenost. (zbirka zadataka, str. 6, zad 2.)

Uputa: koristiti formulu za udaljenost dvije točke u koordinatnom sustavu:

Funkcija za drugi korjen u C-u je `sqrt()`. Funkcija za kvadrat nekog broja ne postoji. Za korištenje funkcije `sqrt()` potrebno je dodati

$$d(A,B) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

`#include<math.h>`.

3. Zadan je kompleksan broj. Izračunaj njegov modul. (zad6, str. 10)

Uputa: $z = a + bi$, $|z| = \sqrt{a^2 + b^2}$

4. Zadana su dva kompleksna broja. Ispiši učitane brojeve, njihovu sumu i razliku.

Uputa: $z_1 = a + bi$, $z_2 = c + di$, $z_1 \pm z_2 = (a + c) \pm i(b + d)$

Rješenja:

1.

```
#include<stdio.h>
#include<conio.h>
int r;
float o,p,pi=3.14159;
main()
{
clrscr();
printf("\nupisi polumjer kruga:",&r);
scanf("%d",&r);
o=2*r*pi;
p=r*r*pi;
printf("opseg danog kruga je %.2f, a površina %.2f",o,p);
getch();
}
```

2.

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
int x1,x2,y1,y2;
float d;
main()
{
clrscr();
printf("\nupisi koordinate tocke A:");
scanf("%d,%d",&x1,&y1);
printf("\nupisi koordinate tocke B:");
scanf("%d,%d",&x2,&y2);
d=sqrt((x2-x1)*(x2-x1)+(y2-y1)*(y2-y1));
printf("udaljenost između danih točaka je %.2f",d);
getch();
}
```

3./*ucitava kompleksan broj i ispisuje njegov modul*/

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
void main()
{
float a,b,m;
clrscr();
printf("Upisi jedan kompleksni broj: \n");
scanf("%f,%f",&a,&b);
m=sqrt(a*a+b*b);
printf("Modul upisanog kompleksnog broja je %.2f ",m);
getch();
}
```


4./*ucitava dva kompleksna broja i ispisuje njihovu sumu i razliku*/

```
#include<stdio.h>
#include<conio.h>
main()
{
float a1,b1,a2,b2,rz,rr,iz,ir;
clrscr();
printf(" Upisi prvi kompleksni broj: ");
scanf("%f,%f",&a1,&b1);
printf(" Upisi drugi kompleksni broj:");
scanf("%f,%f", &a2,&b2);
rz=a1+a2;
rr=a1-a2;
iz=b1+b2;
ir=b1-b2;
printf("\n Upisali ste brojeve %.2f + %.2fi i %.2f + %.2fi \n",a1,b1,a2,b2);
printf("\n Njihova suma je %.2f + %.2fi",rz,iz);
printf("\n Razlika je %.2f + %.2fi",rr,ir);
getch();
}
```

10. Relacijski i logički operatori

Relacijski operatori:

manje od	<
veće od	>
jednako	==
manje ili jednako	<=
veće ili jednako	>=
različito	!=

Relacijski operatori imaju niži prioritet od aritmetičkih operatora.

Npr. u izrazu $a > x + 1$ najprije će se obaviti operacija zbrajanja, a zatim uspoređivanja.

Logički operatori:

logički i (AND)	&&
logički ili (OR)	
negacija (NOT)	!

Najveći prioritet ima operator !, a ako se u izrazu nađu logički i relacijski operatori prioritet je slijedeći:

!
>, >=, <, <=
==, !=
&&
| |

Rezultati relacijskih i logičkih izraza mogu biti samo 0 ili 1 (true ili false)

Zadaci za vježbu:

Napiši u programskom jeziku C:

a) $(a \text{ AND } b) < (c \text{ OR } d)$ Rj: $(a \&\&b) < (c \ | \ | \ d)$

b) $(x \text{ OR } y) = z$ Rj: $x \ | \ | \ y = z$

c) $a \neq b$ Rj: $a != b$

11. Jednostruko uvjetno grananje

Opći oblik za jednostruko uvjetno grananje programa je:

```
if (uvjet)
{
    blok naredbi;
}
```

Ako je uvjet ispunjen izvršava se blok naredbi, koji se može sastojati od jedne ili više naredbi. Početak bloka u C-u se označava sa {, a kraj sa }. Ako se blok naredbi sastoji od jedne naredbe, oznake za početak i kraj bloka nisu neophodne.

Primjer:

```
/*ucitava broj i ispisuje da li je veci, manji ili jednak nuli*/
#include <stdio.h>
#include <conio.h>
int a;
main()
{
    clrscr();
    printf("ucitaj jedan broj: ");
    scanf("%d", &a);
    if (a>0) printf("ucitani broj je veci od nule");
    if (a<0) printf("ucitani broj je manji od nule");
    if (a==0) printf("ucitani broj je nula");
    getch();
}
```

Zadatak: Učitaj broj x . Izračunaj $f(x)$ ako je $f(x) = \begin{cases} x^2, & x < 0 \\ 0, & x = 0 \\ \sqrt{x}, & x > 0 \end{cases}$.

12. Vježba IF

Zadaci:

1. Zadaj prirodni broj i i ispitaj da li je paran ili neparan.
2. Napiši program koji će od tri učitana broja ispisati najveći.
3. Učitaj brojeve a i b , a zatim izračunaj $x = \begin{cases} a + 2b, & a < b \\ 2a - 3b, & a \geq b \end{cases}$.

Rješenja:

```
1./*ucitava broj i ispisuje da li je paran ili neparan*/
#include <conio.h>
#include <stdio.h>
main()
{
    int a;
    clrscr();
    printf("Upisi jedan broj \n");
    scanf("%d", &a);
    if (a%2!=0) printf ("\n Broj %d je neparan", a);
    if (a%2==0) printf ("\n Broj %d je paran", a);
    getch();
}
```

```

2./*ucitava tri broja i ispisuje najveći*/
#include <stdio.h>
#include <conio.h>
main()
{
int max,a,b,c;
clrscr();
printf("\n Upisi tri broja: ");
scanf("%d,%d,%d",&a,&b,&c);
max=a;
if (b>max) max=b;
if (c>max) max=c;
printf ("\n Najveći od upisanih brojeva %d, %d, %d je %d",a,b,c,max);
getch();
}

```

3.

```

#include <stdio.h>
#include <conio.h>
int a,b,x;
main()
{
clrscr();
printf("upisi dva broja: ");
scanf("%d,%d",&a,&b);
if (a<b) x=a+2*b;
if (a>=b) x=2*a-3*b;
printf("x=%d",x);
getch();
}

```

13. Višestruki IF

Opći oblici naredbi višestrukih uvjetnih grananja programa su:

```

1) if (uvjet)
{
    blok naredbi1;
}
else
{
    blok naredbi2;
}

```

tj.

```

2) if (uvjet1)
{
    blok naredbi1;
}
else if (uvjet2)
{
    blok naredbi2;
}
...
else(uvjet n)
{
    blok naredbi n;
}

```

Za prvi oblik naredbe if-else vrijedi: ako je uvjet ispunjen izvršava se blok naredbi1, a ako nije blok naredbi2.

Drugi oblik naredbe if-else if-else omogućuje ispitivanje niza uvjeta. Ako je bilo koji uvjet ispunjen izvršava se blok naredbi pridružen tom uvjetu, a ako nije ispunjen niti jedan uvjet izvršava se blok naredbi n. Broj uvjeta nije ograničen.

Primjer 1:

```
/*ucitava broj i ispisuje da li je paran ili neparan*/
#include <conio.h>
#include <stdio.h>
main()
{
int a;
clrscr();
printf("Upisi jedan broj \n");
scanf("%d",&a);
if (a%2!=0)
    printf ("\n Broj %d je neparan",a);
else
    printf ("\n Broj %d je paran",a);
getch();
}
```

Primjer 2:

```
/*ucitava broj i ispisuje da li je veci ili manji od nule*/
#include <stdio.h>
#include <conio.h>
void main()
{
int a;
clrscr();
printf("Upisi jedan broj \n");
scanf("%d",&a);
if (a>0)
    printf("\n Broj %d je veci od nule",a);
else if (a<0)
    printf("\n Broj %d je manji od nule",a);
else
    printf("\n Broj %d je nula",a); {
    }
getch();
}
```

Zadatak: Učitaj broj. Ako je veći od nule umanjiti ga za jedan, a ako je manji od nule uvećati ga za 1.

Rj:

```
a)/*negativan broj uvecaj za 1, a pozitivan umanji za 1*/
#include <stdio.h>
#include <conio.h>
int a;
main()
{
clrscr();
printf("\n upisi jedan broj: ");
scanf("%d",&a);
if (a<0)
a++;
else
a--;
printf ("%d",a);
getch();
}
```

```

b)/*negativan broj uvecaj za 1, a pozitivan umanji za 1*/
#include <stdio.h>
#include <conio.h>
int a;
main()
{
clrscr();
printf("\n upisi jedan broj: ");
scanf("%d",&a);
if (a<0)
a=a+1;
else
a=a-1;
printf ("%d",a);
getch();
}

```

14. Vježba IF

Zadaci:

1. Napiši program koji će za zadane stranice trokuta izračunati površinu. (zad2, str. 12)

$$Uputa: P = \sqrt{s(s-a)(s-b)(s-c)}, \quad s = \frac{a+b+c}{2}$$

u trokutu zbroj bilo koje dvije duljine stranica mora biti veći od treće. Isto tako duljine stranica moraju biti veće od nule.

2. Učitaj koeficijente kvadratne jednadžbe i ispiši rješenja x1 i x2. (zad7, str 16.)

Uputa: $ax^2 + bx + c = 0$, $a, b, c \in R, a \neq 0$. Ovisno o diskriminanti $D = b^2 + 4ac$, imamo slijedeća rješenja: $D > 0$, dva različita realna rješenja, $D = 0$, dva jednaka realna rješenja, $D < 0$, dva konjugirano kompleksna rješenja

Rješenja:

```

1.
#include <stdio.h>
#include <conio.h>
#include <math.h>
float a,b,c,o,p,s;
main()
{
clrscr();
printf("upisi duljine stranica trokuta a, b, c: ");
scanf("%f,%f,%f",&a,&b,&c);
if (a+b>c && a+c>b && b+c>a && a>0 && b>0 && c>0)
{
s=(a+b+c)/2;
o=a+b+c;
p=sqrt(s*(s-a)*(s-b)*(s-c));
printf("opseg zadanog trokuta je %f, a površina %.2f",o,p);
}
else
printf("zadani brojevi ne cine stranice trokuta");
getch();
}

```

2.

```
/*ucitava koeficijente kvadratne jednadzbe i ispisuje rjesenja*/
#include <stdio.h>
#include <conio.h>
#include <math.h>
float a,b,c,d,x1,x2;
main()
{
clrscr();
printf("ucitaj koeficijente a, b i c: ");
scanf("%f,%f,%f",&a,&b,&c);
if (a!=0)
{
d=b*b-4*a*c;
if (d>0)
{
d=sqrt(d);
x1=(-b+d)/(2*a);
x2=(-b-d)/(2*a);
printf("x1=%.2f, x2=%.2f",x1,x2);
}
else if (d==0)
{
x1=(-b/(2*a));
printf("x1=x2=%.2f",x1);
}
else
{
x1=-b/(2*a);
d=-d;
x2=sqrt(d)/(2*a);
printf("x1=%.2f+%.2fi, x2=%.2f-%.2fi",x1,x2,x1,x2);
}
}
else
{
printf("a mora biti razlicit od nule, jednadzba nije kvadratna");
}
getch();
}
```

15. Uvjetni operator

Kao kraći zapis jednostavnijih if naredbi može se upotrijebiti uvjetni operator ?:

(izraz1)?izraz2:izraz3;

gdje je izraz1 uvjetni izraz, a izraz2 i izraz3 aritmetički izrazi ili funkcije.

Navedena konstrukcija koristi se za zamjenu niza naredbi

```
if(izraz1)
    izraz2;
else
    izraz3;
```

Primjer:

```
/*ucitati dva broja i ispisati veci (uvjetni operator)*/
#include <stdio.h>
#include <conio.h>
void main()
{
    int a,b;
    clrscr();
    printf("\n Upisi dva broja: ");
    scanf("%d,%d",&a,&b);
    (a<b)? printf ("\n Veci je %d",b) : printf ("\n Veci je %d",a);
    getch();
}
```

Zadaci:

1. Napiši program koji će od većeg broja oduzeti manji. (zad 2, str 18)
2. Napiši program koji će za zadani broj ispisati njegovu apsolutnu vrijednost (bez upotrebe funkcije abs). (zad 4, str 19)
3. sgn funkcija: za broj veći od nule ispisuje 1 za nulu ispisuje 0, a za broj manji od 0 -1. (zad 5, str 19)

Rješenja:

1. a)

```
/* od veceg broja oduzima manji */
#include <stdio.h>
#include <conio.h>
void main()
{
    int a,b,rez;
    clrscr();
    printf("\n Upisi dva broja: ");
    scanf("%d,%d",&a,&b);
    (a<b)? printf("\n %d - %d = %d",b,a,b-a):printf("\n %d - %d = %d",a,b,a-b);
    getch();
}
```

b)

```
/* od veceg broja oduzima manji */
#include <stdio.h>
#include <conio.h>
void main()
{
    int a,b,rez;
    clrscr();
    printf("\n Upisi dva broja: ");
    scanf("%d,%d",&a,&b);
    rez = (a<b)? b-a:a-b;
    printf("\n  %d",rez);
    getch();
}
```

2.

```
/*ispis apsolutne vrijednosti zadanog broja*/
#include <stdio.h>
#include <conio.h>
void main()
{
int a,b;
clrscr();

printf("\n Upisi jedan broj: ");
scanf("%d",&a);
b= (a<0)? -a : a;
printf("\n Apsolutna vrijednost ucitanog broja je %d",b);
getch();
}
```

3.

```
/* sgn funkcija */
# include <conio.h>
# include <stdio.h>
void main()
{
int a,b;
clrscr();
printf("\n Upisi jedan broj: ");
scanf("%d",&a);
b= (a>0)? 1: (a<0)? -1 : 0;
printf("\n predznak broja je %d",b);
getch();
}
```


16. SWITCH-CASE naredba

koristi se kod višestrukog grananja programa. Omogućava izbor jednog između više mogućih puteva daljnjeg izvođenja programa, što ovisi o vrijednosti varijable ili izraza. Opća struktura naredbe glasi:

```
switch (izraz)
{
    case konstanta 1:
        blok naredbi 1;
        break;
    case konstanta 2:
        blok naredbi 2;
        break;
        ...
    case konstanta n-1:
        blok naredbi n-1;
        break;
    default:
        blok naredbi n;
        break;
}
```

Tok izvršavanja switch-case naredbe je slijedeći: prvo se izračuna vrijednost izraza, a zatim se uspoređuje sa konstantama (1 do n-1). Ako je izračunati izraz jednak nekoj od konstanti, izvršava se blok naredbi pridružen toj konstanti, a ako rezultat izraza nije jednak niti jednoj od navedenih konstanti, izvršava se blok naredbi pridružen opciji default. Izvršavanje bloka naredbi pridruženog određenoj konstanti prekida se naredbom break. Ako blok naredbi u case ne završava sa break, nastavlja se izvršavati blok slijedećeg case. Opcija default se može izostaviti. Tada se, u slučaju da vrijednost izraza nije jednaka ni jednoj konstanti, izvršavanje programa nastavlja od prve naredbe koja slijedi iza switch-case strukture.

Primjer: Učitaj broj i u ovisnosti o ostatku pri dijeljenju s brojem 7 ispiši dan u tjednu.

```
#include <stdio.h>
#include <conio.h>
#include <math.h>
int x;
float fx;
main()
{
    clrscr();
    printf("ucitaj jedan broj:");scanf("%d",&x);
    switch(x%7)
    {
        case 0:
            printf("\n ponedjeljak");
            break;
        case 1:
            printf("\n utorak");
            break;
        case 2:
            printf("\n srijeda");
            break;
        case 3:
            printf("\n cetvrtak");
            break;
        case 4:
            printf("\n petak");
            break;
        case 5:
            printf("\n subota");
            break;
        default:
            printf("\n nedjelja");
            break;
    }
    getch();
}
```

```
}
```

Zadatak: Učitati dva broja. Ovisno o izabranoj operaciji ispisati sumu, umnožak ili količnik učitanih brojeva.

Rješenje:

```
#include<stdio.h>
#include<conio.h>
main()
{
    float a,b,x;int i;
    clrscr();
    printf(" zadaj dva broja a i b: ");
    scanf("%f,%f", &a, &b);
    printf("izbor:");
    printf(" \n 1: zbroj ucitanih brojeva \n 2: umnozak \n 3: kolicnik");
    printf(" \n izaberi zeljenu operaciju: ");
    scanf ("%d", &i);
    switch(i)
    {
        case 1:
            x=a+b;
            printf ("\n x=%.2f+%.2f=%.2f", a,b,x);
            break;
        case 2:
            x=a*b;
            printf ("\n x=%.2f*%.2f=%.2f", a,b,x);
            break;
        case 3:
            if (b==0)
                printf("\n dijeljenje s nulom nije dozvoljeno");
            else
            {
                x=a/b;
                printf ("\n x=%.2f/%.2f=%.2f", a,b,x);
            }
            break;
        default:
            printf("\n Ucitani broj mora biti izmedu 1 i 3");
            break;
    }
    getch();
}
```

17. Vježba CASE

Zadaci:

1. Koristeći case naredbu i jednu numeričku cjelobrojnu varijablu ispiši mjesec u godini.
3. Zadaj dva kompleksna broja. Ovisno o izboru, izračunaj i ispiši rezultat zbrajanja, oduzimanja, množenja ili dijeljenja zadanih brojeva.

Uputa: $z_1 = a + bi, z_2 = c + di, z_1 \pm z_2 = (a + c) + i(b + d), z_1 z_2 = ac - bd + i(bc + ad),$

$$\frac{z_1}{z_2} = \frac{ac + bd}{c^2 + d^2} + i \frac{bc - ad}{c^2 + d^2} .$$

Rješenja:

```
1.
#include <stdio.h>
#include <conio.h>
int x;float fx;
main()
{
clrscr();printf("ucitaj jedan broj:");scanf("%d",&x);
switch(x%12)
{
case 0:printf("\n sijecanj");break;
case 1: printf("\n veljaca");break;
case 2: printf("\n ozujak");break;
case 3: printf("\n travanj");break;
case 4: printf("\n svibanj");break;
case 5: printf("\n lipanj");break;
case 6: printf("\n srpanj");break;
case 7: printf("\n kolovoz");break;
case 8: printf("\n rujan");break;
case 9: printf("\n listopad");break;
case 10:printf("\n studeni");break;
default:printf("\n prosinac");break;
}
getch();
}
```

```

3.
#include<stdio.h>
#include<conio.h>
main()
{
float a,b,c,d,rz,rr,ru,rk,iz,ir,iu,ik; int i;
clrscr();
printf(" Upisi prvi kompleksni broj:"); scanf("%f,%f",&a,&b);
printf(" Upisi drugi kompleksni broj:");
scanf("%f,%f", &c,&d);
printf("\n izbor:");
printf("\n1)zbroj ucitanih brojeva \n2)razlika \n3)umnozak \n4)kolicnik");
printf("\n odaberi zeljeni broj: "); scanf("%d",&i);
switch (i){
case 1:
rz=a+c; iz=b+d;
printf("\n Njihova suma je %.2f + %.2fi",rz,iz);
break;
case 2:
rr=a-c; ir=b-d;
printf("\n Razlika je %.2f + %.2fi",rr,ir);
break;
case 3:
ru=a*c-b*d; iu=b*c+a*d;
printf("\n Umnozak je %.2f + %.2fi",ru,iu);
break;
case 4:
if (c==0 && d==0)
printf("dijeljenje s nulom");
else
{
rk=(a*c+b*d)/(c*c+d*d); ik=(b*c-a*d)/(c*c+d*d);
printf("\n Kolicnik je %.2f + i %.2fi", rk,ik);
}
break;
default:
printf("upisali ste pogresan broj");
break;}
getch();
}

```

18. GOTO naredba

Naredba goto zahtjeva simboličku oznaku linije na koju se program grana nakon izvršene naredbe. Simbolička oznaka linije u C-u je identifikator sa znakom ..

Treba imati u vidu da velik broj goto naredbi smanjuje čitljivost programa, pa tako i otežava eventualne naknadne ispravke. Međutim, ako je očito da će korištenjem naredbe goto program postati pregledniji, kraći i razumljiviji, nema razloga da se goto ne upotrijebi.

Primjer: Ispis brojeva od 100 do 200 uz pomoć naredbe goto.

```

/*ispisuje brojeve od 100 do 200*/
#include <stdio.h>
#include <conio.h>
int a=100;
main()
{
clrscr();
poc:printf("%d,",a);
a++;
if (a<=200)
goto poc;
getch();
}

```

Zadaci:

1. Napiši program koji će za zadani prirodni broj ispisati umnožak njegovih znamenki (zad 2, str 38).
2. Napiši program koji će za učitani broj ispisati njegove proste faktore. Uputa:12=1*2*2*3

Rješenja:

```
1. /*ucitava broj i ispisuje umnozak njegovih znamenki*/
#include <stdio.h>
#include <conio.h>
int a,z,u=1;
main()
{
clrscr();
printf("ucitaj jedan prirodan broj: ");
scanf("%d",&a);
poc:z=a%10;
u=u*z;
a=a/10;
if (a!=0)
goto poc;
printf("umnozak znamenki ucitanog broja je %d",u);
getch();
}
```

```
2./*rastavlja ucitani broj na proste faktore*/
#include <stdio.h>
#include <conio.h>
int a,f=2;
main()
{
clrscr();
printf("ucitaj prirodni broj: ");scanf("%d",&a);
printf("%d=1",a);
poc: if (a!=1)
{
if (a%f==0)
{
printf("*%d",f);
a=a/f;goto poc;
}
else
{
f=f+1;goto poc;
}
}
getch();
}
```

19. Priprema za pismeni**Zadaci:**

1. if: Napiši program koji će od tri broja ispisati najmanji.
2. case: Zadane su stranice pravokutnika. Ovisno o izboru izračunaj opseg, površinu ili duljinu dijagonale zadanog pravokutnika.

Uputa: $o = 2(a + b)$, $P = ab$, $d = \sqrt{a^2 + b^2}$.

3. goto: Omogućiti da se učitavaju realni brojevi sve dok se ne učita 0. Ispiši:
 - a) koliko je učitanih elemenata
 - b) koliko je među učitanim brojevima pozitivnih, koliko negativnih
 - c) aritmetičku sredinu učitanih brojeva (zbirka snjez, zad.27, str.31)

Rješenja:

```
1./*ucitava tri broja i ispisuje najmanji*/
#include <stdio.h>
#include <conio.h>
main()
{
int min,a,b,c;
clrscr();
printf("\n Upisi tri broja: ");
scanf("%d,%d,%d",&a,&b,&c);
min=a;
if (b<min) min=b;
if (c<min) min=c;
printf ("\n Najveci od upisanih brojeva %d, %d, %d je %d",a,b,c,min);
getch();
}
```

2.

```
#include <stdio.h>
#include <conio.h>
#include <math.h>
main()
{
int i;float a,b,p,o,d;
clrscr();
printf("\n upisi duljine stranica a i b: ");
scanf("%f,%f", &a, &b);
if (a>0 && b>0)
{
printf("\n zelis li izracunati:\n 1)opseg \n 2)povrsinu \n 3)duljinu
dijagonale");
printf("\n izaberi zeljeni broj: ");
scanf("%d",&i);
switch (i)
{
case 1:
o=2*(a+b);
printf("\n Opseg pravokutnika iznosi %.2f",o);
break;
case 2:
p=a*b;
printf("\n Povrsina pravokutnika iznosi %.2f",p);
break;
case 3:
d=sqrt(a*a+b*b);
printf("\n duljina dijagonale je %.2f",d);
break;
default:
printf ("izabrali ste pogresan broj");
break;
}
}
else
printf ("niste upisali stranice pravokutnika");
getch();
}
```

3. /*ucitava realne brojeve sve dok se ne ucita nula, ispisuje koliko je ucitanih brojeva, koliko pozitivnih, koliko negativnih, kolika je aritmeticka sredina ucitanih elemenata*/

```
#include <stdio.h>
#include <conio.h>
float s, a, br, poz, neg, as;
main()
{
  clrscr();
  poc: printf("\nucitaj broj");
  scanf("%f", &a);
  if (a!=0)
  {
    br=br+1;
    s=s+a;
    if (a>0)
      poz=poz+1;
    else
      neg=neg+1;
    goto poc;
  }
  as=s/br;
  printf("ucitano je %.0f brojeva, od toga %.0f poz., a %.0f neg. ",br,poz,neg);
  printf("\n Aritmeticka sredina ucitanih brojeva je %.5f",as);
  getch();
}
```

22. Programske petlje, for petlja

Programske petlje u programskom jeziku C su: FOR, WHILE (s ispunjavanjem uvjeta na početku petlje) i DO-WHILE (s ispunjavanjem uvjeta na kraju petlje)

FOR petlja:

```
for (izraz1;izraz2;izraz3)
{
  blok naredbi;
}
```

gdje je izraz1 početna vrijednost kontrolne varijable, izraz2 uvjet koji kontrolna varijabla mora zadovoljiti da bi se izvršio blok naredbi, a izraz3 je dio u kojem se mijenja stanje kontrolne varijable.

Primjer:

```
/*ispisuje prvih 10 prirodnih brojeva*/
#include <stdio.h>
#include <conio.h>
int i;
main()
{
  clrscr();
  for(i=1;i<=10;i++)
    printf("%d ",i);
  getch();
}
```

Zadatak 1: Ispiši prvih 20 prirodnih brojeva u obrnutom redosljedu (od 20 do 1)

Rješenje:

```
/*ispisuje prvih 20 prirodnih brojeva u obrnutom redosljedu*/
#include <stdio.h>
#include <conio.h>
int i;
main()
{
clrscr();
for(i=20;i>=1;i--)
printf("%d ",i);
getch();
}
```

Zadatak 2: Ispiši prvih 10 prirodnih brojeva, svaki broj u novi red.

Rješenje:

```
/*ispisuje prvih 10 prirodnih brojeva, svaki broj u novi red*/
#include <stdio.h>
#include <conio.h>
int i;
main()
{
clrscr();
for(i=1;i<=10;i=i+1)
printf("%d \n",i);
getch();
}
```

23. Vježba for

Zadaci:

1. Napiši program koji će zbrojiti sve cijele brojeve između 200 i 300 koji su dijeljivi sa 7. Koliko ima takvih brojeva?
2. Zadaj n brojeva. Izračunaj njihovu aritmetičku sredinu.
3. Napiši program koji će učitati n brojeva i ispisati najveći od njih.
4. Upiši težine za n učenika. Ispiši prosječnu težinu i koliko ima učenika čija je težina manja od 65 kg.

Rješenja:

```
1. /*suma brojeva između 200 i 300 dijeljivih sa 7*/
#include <conio.h>
#include <stdio.h>
main()
{
int s,i,br;
clrscr();
s=0;br=0;
for(i=200;i<=300;i++)
if (i % 7 == 0)
{
s=s+i;
br=br++;
}
printf("\n suma brojeva dijeljivih sa 7 je %d, a ima ih %d",s,br);
getch();
}
```


2. /*Aritmeticka sredina n brojeva*/

```
#include <conio.h>
#include <stdio.h>
main()
{
    int i,br;
    float a,s,ars;
    clrscr();
    s=0;
    printf("\n Koliko imas brojeva? ");
    scanf("%d",&br);
    for(i=1;i<=br;i++)
    {
        printf("\n Zadaj %d. broj: ",i);
        scanf("%f",&a);
        s+=a;
    }
    ars=s/br;
    printf("\n Aritmeticka sredina zadanih brojeva je %.2f",ars);
    getch();
}
```

3. /*najveci od n brojeva*/

```
#include <stdio.h>
#include<conio.h>
void main()
{
    int i,n;
    float br,max;
    clrscr();
    printf("\n Koliko imas brojeva? ");scanf("%d",&n);
    printf("\n Zadaj 1. broj: ");scanf("%f",&br);
    max=br;
    for(i=2;i<=n;i++)
    {
        printf("\n Zadaj %d. broj: ",i);
        scanf("%f",&br);
        if (br>max) max=br;
    }
    printf("\n Najveci od upisanih brojeva je %.2f", max);
    getch();
}
```

4. /*prosjecne tezine za n ucenika*/

```
#include <conio.h>
#include <stdio.h>
main()
{
    int i,br=0,n;
    float a,s,ars;
    clrscr();s=0;
    printf("\n Koliko imas ucenika? ");scanf("%d",&n);
    for(i=1;i<=n;i++)
    {
        printf("\n Zadaj tezinu %d. ucenika: ",i); scanf("%f",&a);
        s+=a;
        if (a<65) br=br++;
    }
    ars=s/n;
    printf("\n Prosjecna tezina ovih ucenika je %.2f, a laksih od 65 kila ima %d",ars,br);
    getch();
}
```

24. Programska petlja while

Opći oblik programske petlje while glasi:

```
while (uvjet)
{
    blok naredbi;
}
tj.
dok je (ispunjen uvjet)
{
    radi...
}
```

Dakle, blok naredbi u petlji se ponavlja sve dok je uvjet istinit. Kad uvjet postane lažan program se grana na prvu naredbu iza while. Uvjet ponavljanja petlje ispituje se na početku petlje, što znači da blok naredbi ne mora nužno biti izvršen niti jednom.

Primjer:

```
/*ispisuje prvih 10 prirodnih brojeva*/
#include <stdio.h>
#include <conio.h>
int i;
main()
{
    clrscr();
    i=1;
    while (i<=10)
    {
        printf("%d ", i);
        i++;
    }
    getch();
}
```

Napomena: U ovakvim situacijama prikladnija je petlja for.

25. Vježba while

Zadaci:

1. Izračunaj zbroj neparnih brojeva od 1 do 100. (2500)
2. Učitaj n i ispiši n!.
3. Učitaj n i izračunaj $s = 1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \dots \pm \frac{1}{n}$.

Rješenja:

1.

```
a)/*ispisuje sumu neparnih brojeva između 1 i 100, koristeći while*/
#include <stdio.h>
#include <conio.h>
main()
{
    int i,s;
    clrscr();
    i=1;s=0;
    while (i<=100)
    {
        if(i%2 !=0)
            s+=i;
        i=i++;
    }
    printf("\n suma je %d",s);
    getch();
}
```

```

b) /*na drugi nacin suma neparnih brojeva manjih od 100*/
#include <stdio.h>
#include <conio.h>
main()
{
int i,s;
clrscr();
i=1;
s=0;
while(i<=100)
{
s+=i;
i=i+2;
}
printf("\n suma je %d",s);
getch();
}

```

```

2. /*za ucitani n ispisuje n!*/
#include<conio.h>
#include<stdio.h>
void main()
{
int i,f,n;
clrscr();
i=1;
f=1;
printf("\n Upisi jedan broj: ");
scanf("%d",&n);
while(i<=n)
{
f=f*i;
i=i++;
}
printf("\n %d! = %d",n,f);
getch();
}

```

```

3. /*za ucitani n ispisuje s=1-1/2+1/3-...+-1/n*/
#include<conio.h>
#include<stdio.h>
main()
{
int p,i,n;
float s;
clrscr();
i=1;
s=0;
p=1;
printf("\n Upisi jedan broj: ");
scanf("%d",&n);
while(i<=n)
{
s+=(float)p/i;
i++;
p=-p;
}
printf("\n suma je %.3f",s);
getch();
}

```

26. Programska petlja do-while

predstavlja grupu programskih petlji kod kojih se uvjet ponavljanja ispituje na kraju bloka naredbi petlje, što znači da se blok naredbi mora izvesti barem jednom.

Opći oblik petlje do-while glasi:

```
do
{
    blok naredbi;
}while (uvjet);
```

Ovakve petlje koriste se u slučajevima kada se blok naredbi mora izvršiti barem jednom, no češće se koriste petlje for i while.

Primjer:

```
/*ispisuje prvih 10 prirodnih brojeva*/
#include <stdio.h>
#include <conio.h>
int i;
main()
{
    clrscr();
    i=1;
    do
    {
        printf("%d ",i);
        i++;
    }while (i<=10);
    getch();
}
```

Zadaci:

1. Napiši program koji će zbrojiti prvih 10 prirodnih brojeva.
2. Pomoću do-while petlje zbroji troznamenkaste brojeve djeljive sa 17. Koliko ima takvih brojeva?

Rješenja:

```
1./*ispisuje i zbraja prvih 10 prirodnih brojeva*/
#include <stdio.h>
#include <conio.h>
int i,s=0;
main()
{
    clrscr();
    i=1;
    do
    {
        printf("%d ",i);
        i++;
        s=s+i;
    }while (i<=10);
    getch();
}
```

```

2./*ispisuje sumu troznamenkastih brojeva djeljivih sa 17 i njihov broj,
koristeci while*/
#include <stdio.h>
#include <conio.h>
main()
{
int i=100,s=0,br=0;
clrscr();
do
{
if(i%17==0)
{
s+=i;
br++;
}
i=i++;
} while (i<=999);
printf("\n suma je %d, a ima ih %d",s,br);
getch();
}

```

27. Vježba petlji

Zadaci:

1. Koristeći for petlju za učitani n izračunaj: $s = 1 + 2^2 + 3^3 + \dots + n^n$.
2. Pomoću while petlje izračunaj zbroj svakog trećeg od prvih n prirodnih brojeva.
3. Koristeći do-while, za učitani n izračunaj sumu

$$s = 1 - \frac{1}{3} + \frac{1}{5} - \dots \pm \frac{1}{n}$$

Rješenja:

```

1./*za ucitani n izracunava 1+2^2+3^3+...+n^n*/
#include <stdio.h>
#include <conio.h>
#include <math.h>
void main()
{
int i=0, s=0, n,j,pot;
clrscr();
scanf("%d",&n);
for (i=1;i<=n;i++)
{
pot=i;
for(j=2;j<=i;j++)
pot=pot*i;
printf("\n %d",pot);
s=s+pot;
}
printf ("\n suma je %d",s);
getch();
}

```

```

2. /*ispisuje sumu svakog 3. od n ucitanih brojeva, pomocu while*/
#include<conio.h>
#include<stdio.h>
main()
{
int i,s,n;
clrscr();
i=3;
s=0;
printf("\n Upisi jedan broj: ");
scanf("%d",&n);
while(i<=n)
{
s=s+i;
i=i+3;
}
printf("\n Suma svakog treceg, od %d brojeva, je %d",n,s);
getch();
}

```

3.

```

#include <stdio.h>
#include <conio.h>
main()
{
int i=1,n;
float p=1.0,s=0.0;
clrscr();
printf("ucitaj broj n: ");
scanf("%d",&n);
do
{
s=s+(p*1/float(i));
i=i+2;
p=-p;
} while (i<=n);
printf("\n s=%.3f",s);
getch();
}

```

31. Jednodimenzionalna polja - učitavanje i ispis

Skup podataka istog tipa koji nosi zajedničko ime naziva se polje. Član polja jednoznačno je određen svojim položajem u polju koji se iskazuje indeksom. Indeksi polja u C-u kreću se od 0.

npr: $a = (a_1, a_2, \dots, a_n)$

deklaracija polja: **tip_polja ime_polja[dimenzija];**

gdje je **tip_polja** jedan od osnovnih tipova podataka u C-u (int, float, double, char), **ime_polja** identifikator, a **dimenzija** broj članova polja.

Primjer: učitavanje i ispis elemenata u jednodimenzionalnom polju.

```
/*ucitava n elemenata u polje i ispisuje ih*/
#include <conio.h>
#include <stdio.h>
void main()
{
    int i,n,a[20];
    clrscr();printf("upisi broj elemenata u polju: ");
    scanf("%d",&n);
    if (n<=20)
    {
        printf("\n upisi elemente polja:\n");
        for (i=0;i<n;i++)
        {
            printf("a[%d] ",i);scanf("%d",&a[i]);
        }
        printf("\n ucitani elementi su: \n");
        for (i=0;i<n;i++)
            printf("\n %d. el. je %d",i,a[i]);
    }
    else
        printf("\n upisali ste prevelik broj");
    getch();
}
```

32. Pretraživanje jednodimenzionalnog polja

Zadaci:

1. Učitaj polje od najviše 10 elemenata. Pronađi i ispiši najveći element.
2. Učitaj u polje 10 elemenata, a zatim upiši jedan broj. Ispiši mjesta u polju na kojima se učitan broj nalazi.
3. Učitaj n elemenata u polje, $n < 10$. Ispiši najveći i najmanji učitan broj, te njihovu aritmetičku sredinu.
4. Učitaj uspjeh za n učenika, $n < 40$. Ispiši koliko ima odličnih, koliko vrlo dobrih, dobrih, dovoljnih i nedovoljnih učenika. Zatim ispiši prosječnu ocjenu i broj učenika iznad prosjeka.
5. Učitaj n elemenata u jedno polje. Ispiši sumu parnih i sumu neparnih brojeva posebno, te sumu elemenata na parnim i sumu elemenata na neparnim mjestima (ne indeksima) u polju.

Rješenja:

1. (pola3.cpp)

```
#include <conio.h>
#include <stdio.h>
void main()
{
    int i,n,a[10],max;
    clrscr();
    printf("upisi broj elemenata u polju: ");
    scanf("%d",&n);
    if (n<=10)
    {
        printf("\n upisi elemente polja:\n");
        for (i=0;i<n;i++)
        {
            printf("%d: ",i+1);
            scanf("%d",&a[i]);
        }
        max=a[0];
        for (i=1;i<n;i++)
            if (a[i]>max) max=a[i];
        printf ("\n najveći od upisanih brojeva je %d",max);
    }
    else
        printf("\n upisali ste prevelik broj");
    getch();
}
```

2. (polja2.cpp)

```
/*za n ucitanih brojeva u polju ispisuje na kojem mjestu je trazeni broj*/
#include <conio.h>
#include <math.h>
#include <stdio.h>
void main()
{
    int i,n,a[10],t,b=0;
    clrscr();
    printf("upisi broj elemenata u polju: ");
    scanf("%d",&n);
    if (n<=10)
    {
        printf("\n upisi elemente polja:\n");
        for (i=0;i<n;i++)
        {
            printf("%d: ",i+1);
            scanf("%d",&a[i]);
        }
        printf("\n upisi broj koji trazis: ");
        scanf("%d",&t);
        for (i=0;i<n;i++)
        {
            if (a[i]==t)
            {
                printf("\n trazeni broj se nalazi na %d. mjestu u polju",i+1);
                b=1;
            }
        }
        if (b==0)
            printf("\n trazenog broja nema u polju");
    }
    else
        printf("\n upisali ste prevelik broj");
    getch();
}
```


3. (polja5.cpp)

```
#include <conio.h>
#include <stdio.h>
int n,a[10],min,max,s=0,i;
float as;
void main(void)
{
    clrscr();
    printf("\n upisi broj elemenata u polju: "); scanf("%d",&n);
    if (n<10)
    {
        /*ucitavanje elemenata polja*/
        for (i=0;i<n;i++)
        {
            printf("%d: ",i+1);scanf("%d",&a[i]);
        }
        max=a[1]; min=a[1];
        /*pretrazivanje polja*/
        for (i=0;i<n;i++)
        {
            s=s+a[i];
            if (a[i]>max) max=a[i];
            if (a[i]<min) min=a[i];
        }
        as=(float)s/n;
        printf("najveci element polja je %d, najmanji %d, a ar. sred.
%.2f",max,min,as);
    }
    else
        printf("\n upisali ste prevelik broj");
    getch();
}
```

4. (polja6.cpp)

```
#include <conio.h>
#include <stdio.h>
int a[40],s, n, od=0,vdob=0,dob=0,dov=0,nedov=0,ip=0,i;
float as;
void main(void)
{
    clrscr();
    printf("upisi broj ucenika: ");
    scanf("%d",&n);
    if (n<=40)
    {
        for (i=0;i<n;i++)
        {
            printf("upisi uspjeh %d. ucenika",i+1);
            scanf("%d",&a[i]);
            if (a[i]==1) nedov=nedov++;
            if (a[i]==2) dov=dov++;
            if (a[i]==3) dob=dob++;
            if (a[i]==4) vdob=vdob++;
            if (a[i]==5) od=od+1;
            s=s+a[i];
        }
        as=(float)s/n;
        for (i=0;i<n;i++)
            if (a[i]>as) ip=ip++;
        printf("\n odlicnih je %d, vrlo dobrih %d, dobrih %d, dovoljnih %d, a
nedovoljnih %d",od,vdob,dob,dov,nedov);
        printf("\n prosjecna ocjena razreda je %.2f, a broj ucenika iznad prosjeka je
%d",as,ip);
    }
    else
        printf ("upisali ste prevelik broj");
    getch();
}
```

33. Zbrajanje jednodimenzionalnih polja i skalarni produkt

Zadaci:

1. Napiši program koji spaja dva jednodimenzionalna polja.

Uputa: $a = (a_1, a_2, \dots, a_n), b = (b_1, b_2, \dots, b_m) \Rightarrow c = (a_1, \dots, a_n, b_1, \dots, b_m)$

2. Učitaj jedno polje od n elemenata i jedan broj (posebno). Ispiši skalarni produkt učitanoj broja sa poljem.

Uputa: $\alpha a = (\alpha a_1, \alpha a_2, \dots, \alpha a_n)$

3. Napiši program koji zbraja dva učitana polja jednake veličine.

Uputa: $a + b = (a_1, a_2, \dots, a_n) + (b_1, \dots, b_n) = (a_1 + b_1, \dots, a_n + b_n)$

Rješenja:

```
1.
#include <conio.h>
#include <stdio.h>
void main(void)
{
    int i,n,m,q,a[20],b[20],c[20];
    clrscr();
    printf("upisi broj elemenata prvom polju: ");
    scanf("%d",&n);
    printf("upisi broj elemenata drugom polju: ");
    scanf("%d",&m);
    if (n<=20 && m<=20)
    {
        printf("\n upisi elemente prvog polja:\n");
        for (i=0;i<n;i++)
        {
            printf("a[%d] ",i+1);
            scanf("%d",&a[i]);
        }
        printf("\n upisi elemente drugog polja:\n");
        for (i=0;i<m;i++)
        {
            printf("a[%d] ",i+1);
            scanf("%d",&b[i]);
        }
        q=m+n;
        for(i=0;i<q;i++)
        {
            if (i<n) c[i]=a[i]; else c[i]=b[i-n];
            printf("%d ",c[i]);
        }
    }
    else
        printf("\n upisali ste prevelik broj");
    getch();
}
```

```

2.
#include <conio.h>
#include <stdio.h>
void main()
{
    int i,n,m,a[20],b[20];
    clrscr();
    printf("upisi broj elemenata u polju: ");
    scanf("%d",&n);
    printf("upisi broj s kojim zelis mnoziti polje: ");
    scanf("%d",&m);
    if (n<=20)
    {
        printf("\n upisi elemente polja:\n");
        for (i=0;i<n;i++)
        {
            printf("a[%d] ",i+1);
            scanf("%d",&a[i]);
        }
        for(i=0;i<n;i++)
        {
            b[i]=m*a[i];
            printf("%d ",b[i]);
        }
    }
    else
        printf("\n upisali ste prevelik broj");
    getch();
}

```

```

3.
#include <conio.h>
#include <stdio.h>
void main()
{
    int i,n,a[20],b[20],c[20];
    clrscr();
    printf("upisi broj elemenata u poljima: ");
    scanf("%d",&n);
    if (n<=20)
    {
        printf("\n upisi elemente prvog polja:\n");
        for (i=0;i<n;i++)
        {
            printf("a[%d] ",i+1);
            scanf("%d",&a[i]);
        }
        printf("\n upisi elemente drugog polja:\n");
        for (i=0;i<n;i++)
        {
            printf("a[%d] ",i+1);
            scanf("%d",&b[i]);
        }
        for(i=0;i<n;i++)
        {
            c[i]=a[i]+b[i];
            printf("%d ",c[i]);
        }
    }
    else
        printf("\n upisali ste prevelik broj");
    getch();
}

```

34. Vježba polja

Zadaci:

1. Učitati jednodimenzionalno polje a na osnovu njega formirati drugo, tako da bude $a_0 = b_0, b_1 = a_0 + b_0, \dots, b_n = a_1 + a_2 + \dots + a_n$.
2. Napiši program koji će učitati dva jednodimenzionalna polja od najviše 100 elemenata i na osnovi njih formirati treće polje tako da je i -ti element trećeg polja jednak manjem od i -tih elemenata učitanih polja. Ako su i -ti elementi prvog i drugog polja jednaki, i -ti element novodobivenog polja treba poprimiti vrijednost 13.

Rješenja:

1.

```
#include <conio.h>
#include <stdio.h>
void main()
{
    int i,n,s=0,a[20],b[20];
    clrscr();
    printf("upisi broj elemenata u polju: ");
    scanf("%d",&n);
    if (n<=20)
    {
        printf("\n upisi elemente polja:\n");
        for (i=0;i<n;i++)
        {
            printf("a[%d] ",i+1); scanf("%d",&a[i]);
            s+=a[i];b[i]=s;
        }
        for (i=0;i<n;i++)
            printf("b[%d]=%d\n",i+1,b[i]);
    }
    else
        printf("\n upisali ste prevelik broj");
    getch();
}
```

2.

```
#include <conio.h>
#include <stdio.h>
void main()
{
    int i,n,a[100],b[100],c[100];
    clrscr();
    printf("upisi broj elemenata u poljima: "); scanf("%d",&n);
    if (n<=100)
    {
        printf("\n upisi elemente prvog polja:\n");
        for (i=0;i<n;i++)
        {
            printf("a[%d] ",i+1); scanf("%d",&a[i]);
        }
        printf("\n upisi elemente drugog polja:\n");
        for (i=0;i<n;i++)
        {
            printf("a[%d] ",i+1); scanf("%d",&b[i]);
        }
        for(i=0;i<n;i++)
        {
            if (a[i]<b[i]) c[i]=a[i];
            else if(a[i]>b[i]) c[i]=b[i];
            else c[i]=13;
            printf("%d ",c[i]);
        }
    }
    else printf("\n upisali ste prevelik broj");
    getch();
}
```

35. Sortiranje

Zadatak: Učitati niz od n elementa. Ispisati učitane elemente poredane po veličini, od najmanjeg prema najvećem, a zatim od najvećeg prema najmanjem.

Uputa: Za sortiranje elemenata polja postoji nekoliko algoritama. Objasniti ćemo jedan od jednostavnijih, iako nije baš efikasan. U prvom koraku, na prvo mjesto dovodimo najmanji broj, u drugom koraku, na drugo mjesto dovodi se drugi po redu broj. Postupak se nastavlja sve dok se ne dođe do potpuno sređenog niza.

Primjer:

A[1]	A[2]	A[3]	A[4]	A[5]
5	4	2	3	1

1. korak

A[1]	A[2]	A[3]	A[4]	A[5]
5	4	2	3	1

2. korak

A[1]	A[2]	A[3]	A[4]	A[5]
4	5	2	3	1

3. korak

A[1]	A[2]	A[3]	A[4]	A[5]
2	5	4	3	1

4. korak

A[1]	A[2]	A[3]	A[4]	A[5]
2	5	4	3	1

Rezultat nakon prvog prolaska kroz polje:

A[1]	A[2]	A[3]	A[4]	A[5]
1	5	4	3	2

U drugom prolasku kroz polje ponavljamo isti postupak sa drugim elementom polja:

A[1]	A[2]	A[3]	A[4]	A[5]
1	5	4	3	2

1. korak

A[1]	A[2]	A[3]	A[4]	A[5]
1	5	4	3	2

2. korak

A[1]	A[2]	A[3]	A[4]	A[5]
1	4	5	3	2

3. korak

A[1]	A[2]	A[3]	A[4]	A[5]
1	3	5	4	2

Rezultat nakon 2. prolaska kroz polje:

A[1]	A[2]	A[3]	A[4]	A[5]
1	2	5	4	3

U trećem prolasku kroz polje uzimamo treći element i uspoređujemo ga sa 4. i 5. Zatim, u 4. prolasku kroz polje uzimamo 4. element i uspoređujemo ga sa 5. Po potrebi zamjenimo. Na kraju ovog postupka početni niz izgleda ovako:

A[1]	A[2]	A[3]	A[4]	A[5]
1	2	3	4	5

Dakle, algoritam za sortiranje izgleda ovako:

```

za i=1 do n-1 radi
  za j=i+1 do n radi
    ako je a[i]>a[j] tada zamjeni a[i] i a[j]

```

Rješenje:

```

#include <conio.h>
#include <stdio.h>
int n,a[10],i,j,pom;
void main(void)
{
  clrscr();
  printf("\n upisi broj elemenata u polju: ");
  scanf("%d",&n);
  if (n<10)
  {
    /*ucitavanje elemenata polja*/
    for (i=0;i<n;i++)
    {
      printf("%d: ",i+1);
      scanf("%d",&a[i]);
    }
    /*sortiranje*/
    for (i=0;i<n-1;i++)
      for (j=i+1;j<n;j++)
        if (a[j]<a[i])
        {
          pom=a[i];
          a[i]=a[j];
          a[j]=pom;
        }
    printf("sortirano polje je:\n");
    for (i=0;i<n;i++)
      printf("%d ",a[i]);
  }
  else printf("upisali ste previse elemenata");
  getch();
}

```

36. Sortiranje polja - vježba

Zadatak:

Upisati bodove za 3 zadatka na natjecanju iz matematike za svakog od n ($n < 30$) učenika. Ispisati rang listu - redni broj natjecatelja i broj ostvarenih bodova.

37. Rad sa stringovima

Niz znakova (string) nije ugrađen kao osnovni podatkovni tip programskog jezika C. Jedan od načina definicije niza znakova je preko jednodimenzionalnog polja znakova (`char str[10]` je niz znakova duljine 10). Kod definicije duljine niza znakova treba imati na umu da se on sastoji od jednodimenzionalnog polja znakova ograničenog nul znakom (`\0`).

Funkcije za rad sa stringom:

gets()-učitava niz znakova sa standardnog ulaza

strlen()-vraća duljinu zadanog niza znakova

Primjer: Učitati string i ispisati njegovu duljinu

Rješenje:

```
a)
#include<conio.h>
#include<stdio.h>
char a[80];
int i=0;
void main(void)
{
clrscr();
printf("unesi string: ");
gets(a);
while (a[i+1]!='\0') i++;
printf("%d", i+1);
getch();
}
```

```
b)
#include<conio.h>
#include<stdio.h>
#include<string.h>
char a[80];
int n;
void main(void)
{
clrscr();
printf("unesi string: ");
gets(a);
n=strlen(a);
printf("%d", n);
getch();
}
```

Zadaci:

1. Napiši program koji će učitano rečenicu ispisati s desna na lijevo.
2. Napiši program koji će za učitano riječ provjeriti da li je palindrom.
3. Napiši program koji će ispisati koliko se puta učitani znak pojavljuje u učitanoj stringu.

Rješenja:

1.

```
/*string3.cpp
ucitava string i ispisuje ga unatraske*/
#include<conio.h>
#include<stdio.h>
#include<string.h>
char a[80];
int n,i;
void main(void)
{
clrscr();
printf("unesi string: ");
gets(a);
n=strlen(a)-1;
/*ispisuje string unatraske*/
for(i=n;i>=0;i--) printf("%c",a[i]);
getch();
}
```

2.

```
/*string4.cpp
ucitava rijec i ispisuje da li je palindrom*/
#include<conio.h>
#include<stdio.h>
#include<string.h>
char a[80];
int n,i,k=0;
void main(void)
{
clrscr();
printf("unesi string: ");
gets(a);
n=strlen(a)-1;
for(i=0;i<=n;i++)
    if (a[i]!=a[n-i])k=1;
if (k==0)printf("ucitana rijec je palindrom");
else printf ("ucitana rijec nije palindrom");
getch();
}
```


3.a)

```
/*string5.cpp
broji pojavljivanje ucitanog znaka u stringu
ne radi za ucitano veliko slovo kao znak, ako se string ucita malim slovima*/
#include<conio.h>
#include<stdio.h>
#include<string.h>
#include<ctype.h>
char a[80],b;
int n,i,br=0;
void main(void)
{
clrscr();
printf("unesi string: ");
gets(a);
n=strlen(a)-1;
printf("upisi slovo koje trazis: ");
scanf("%c",&b);
for(i=0;i<=n;i++)
    if (a[i]==b || a[i]==toupper(b)) br++;
if (br!=0) printf("znak %c se u ucitanom stringu pojavljuje %d puta",b,br);
else printf("znaka %c nema u stringu",b);
getch();
}
```

b)

```
/*string5a.cpp
upisuje string i broji trazeni znak*/
#include<conio.h>
#include<stdio.h>
#include<string.h>
#include<ctype.h>
char a[80],b,c;
int n,i,br=0;
void main(void)
{
clrscr();
printf("unesi string: ");
gets(a);
n=strlen(a)-1;
printf("upisi znak koji trazis: ");
scanf("%c",&b);
for(i=0;i<=n;i++)
{
a[i]=toupper(a[i]);
c=toupper(b);
if (a[i]==c) br++;
}
if (br!=0) printf("znak %c se u ucitanom stringu pojavljuje %d puta",b,br);
else printf("trazenog znaka nema u stringu");
getch();
}
```

Napomena: toupper() je funkcija koja pretvara mala u velika slova. Za nju je potrebno uključiti jedinicu ctype.h.

38. Vježba stringova

Zadaci:

1. Napiši program koji će ispisati koliko riječi ima učitana rečenica.
2. Napiši program koji će u zadanom stringu ispisati broj najdužeg niza uzastopno jednakih znakova. Ispiši taj niz.
3. Napiši program koji će riječi zadane rečenice ispisati jednu ispod druge.
4. Učitaj rečenicu i ispiši da li je palindrom.

Rješenja:

```
1.
/*string6.cpp*/
/*ispisuje broj rijeci u ucitanoj recenici*/
#include<conio.h>
#include<stdio.h>
#include<string.h>
char a[80];
int n,i,k=0,br=0,j;
void main(void)
{
clrscr();
printf("unesi string: ");
gets(a);
n=strlen(a)-1;
/*broji razmake*/
for(i=0;i<n;i++)
    if (a[i]==' ' && a[i+1]!=' ') br++;
printf("ucitana recenica ima %d rijeci",br+1);
getch();
}
```

```
2.
/*string8.cpp*/
/*ispisuje najveći podniz jednakih znakova*/
#include<conio.h>
#include<stdio.h>
#include<string.h>
char a[80];
int n,i,j,k,l=1,znam,max=0;
void main(void)
{
clrscr();
printf("unesi string: ");
gets(a);
n=strlen(a)-1;
for(i=0;i<=n;i++)
{
    if (a[i]== a[i+1])
    {
        l++;
        if (l>max)
        {
            max=l;
            znam=a[i];
        }
    }
    else l=1;
}
printf("najduzi trazeni podniz ima duljinu %d i glasi: ",max);
for (i=0;i<max;i++) printf ("%c",znam);
getch();
}
```

3.

```
/*string7.cpp*/
/*ispisuje rijeci u slijedeci red*/
#include<conio.h>
#include<stdio.h>
#include<string.h>
char a[80];
int n,i,j,k,l;
void main(void)
{
  clrscr();
  printf("unesi string: ");
  gets(a);
  n=strlen(a)-1;
  l=0;
  for(i=0;i<=n;i++)
  {
    if (a[i]==' ' || a[i]==',' || i==n)
    {
      for (k=l;k<=i;k++) printf("%c",a[k]);
      if (a[i]==',' && a[i+1]==' ') l=i+2;
      else
      {
        l=i+1;
        printf("\n");
      }
    }
  }
  getch();
}
```

4.

```
/*string4a.cpp*/
/*provjerava da li je ucitani string palindrom*/
#include<conio.h>
#include<stdio.h>
#include<string.h>
char a[80];
int n,i,k=0,br=0,j;
void main(void)
{
  clrscr();
  printf("unesi string: ");
  gets(a);
  n=strlen(a)-1;
  /*broji razmake*/
  for(i=0;i<=n;i++)
    if (a[i]==' ') br++;
  /*izbacuje razmake*/
  for(i=0;i<=br-1;i++)
    for(j=0;j<=n;j++)
    {
      if(a[j]==' ')
      {
        a[j]=a[j+1];
        a[j+1]=' ';
      }
    }
  /*provjerava da li je ucitani string palindrom*/
  n=n-br;
  for(i=0;i<=n;i++)
    if (a[i]!=a[n-i])k=1;
  if (k==0)printf("ucitani string je palindrom");
  else printf ("ucitani string nije palindrom");
  getch();
}
```

39. Ponavljanje jednodimenzionalnih polja

1. Učitati niz od n cijelih brojeva ($n < 20$). Ispisati najveći negativni učitani broj.
2. Učitati cijeli broj pa ga ispisati u oktalnom brojevnom sustavu.
3. Učitati niz od n cijelih brojeva ($n < 20$). Ispisati poziciju najvećeg elementa po apsolutnoj vrijednosti.
4. Učitati niz od n cijelih brojeva ($n < 20$). Ispisati sumu svih dvoznamenkastih učitanih brojeva i njihov broj.
5. Učitati cijeli broj i ispisati ga u heksadecimanom brojevnom sustavu.
6. Učitati rečenicu od najviše 80 znakova i ispisati broj suglasnika.
7. Učitati niz od n cijelih brojeva. Ispisati najveći negativni broj i njegovu poziciju.
8. Učitati rečenicu od najviše 80 znakova. Ispisati rečenicu bez razmaka.
9. Učitaj niz od najviše 20 elemenata (broj elemenata se učitava). Ispiši zbroj svih neparnih učitanih elemenata i njihov broj.
10. Učitaj niz od najviše 10 elemenata i poredaj ih po veličini, od najvećeg prema najmanjem.

41. Dvodimenzionalna polja - učitavanje i ispis

Višedimenzionalno polje može se opisati kao skup jednodimenzionalnih polja. Da bi se pristupilo određenom elementu potrebno je zadati poziciju elementa u polju. Za razliku od jednodimenzionalnih polja, kod dvodimenzionalnih polja treba zadati i red i stupac u kojem se element nalazi.

U matematici, dvodimenzionalna polja nazivaju se matrice. Za matricu koja ima m redova i n stupaca kažemo da je matrica tipa $m \times n$. Ako matrica ima isti broj redova i stupaca (npr. n) zove se kvadratna matrica (reda n).

$$a = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ & & \vdots & \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}$$

S obzirom da bilo koji element dvodimenzionalnog polja određuje broj reda i broj stupca, za učitavanje i ispis jednog takvog polja potrebne su dvije petlje, jedna za red, a druga za stupac.

Primjer: Učitavanje i ispis dvodimenzionalnog polja od n redova i m stupaca. Ispis je u obliku matrice (tablice).

```
#include<conio.h>
#include<stdio.h>
int a[10][10],i,j,n,m;
void main(void)
{
clrscr();
printf("upisi broj redova matrice: ");scanf("%d",&n);
printf("upisi broj stupaca matrice: ");scanf("%d",&m);
printf("upisi elemente polja:\n");
/*ucitavanje elemenata polja*/
for(i=0;i<n;i++)
for(j=0;j<m;j++)
{
printf("a[%d][%d]: ",i,j);
scanf("%d",&a[i][j]);
}
/*ispis u obliku tablice*/
printf("ucitali ste elemente: \n");
for(i=0;i<n;i++)
{
for(j=0;j<m;j++)
printf("%d ",a[i][j]);
printf("\n"); /*zbog prelaska u novi red*/
}
getch();
}
```

42. Zbrajanje i oduzimanje dvodimenzionalnog polja

Zadatak: Učitaj dvije matrice od n redova i m stupaca. Ispiši njihov zbroj i razliku.

Napomena: Matrice se zbrajaju tako da se zbroje elementi s istim indeksima. Mogu se zbrojiti samo matrice s istim brojem redova i stupaca.

$$c = a + b = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} + \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1n} \\ b_{21} & b_{22} & \dots & b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{m1} & b_{m2} & \dots & b_{mn} \end{bmatrix} =$$

$$\begin{bmatrix} a_{11} + b_{11} & a_{12} + b_{12} & \dots & a_{1n} + b_{1n} \\ a_{21} + b_{21} & a_{22} + b_{22} & \dots & a_{2n} + b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} + b_{m1} & a_{m2} + b_{m2} & \dots & a_{mn} + b_{mn} \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & \dots & c_{1n} \\ c_{21} & c_{22} & \dots & c_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{m1} & c_{m2} & \dots & c_{mn} \end{bmatrix}$$

Rješenje:

```
#include<conio.h>
#include<stdio.h>
int a[10][10],b[10][10],c[10][10],d[10][10],i,j,n,m;
void main(void)
{
clrscr();
printf("upisi broj redova matrica: ");scanf("%d",&n);
printf("upisi broj stupaca matrica: ");scanf("%d",&m);
printf("upisi elemente prve matrice:\n");
for(i=0;i<n;i++)
for(j=0;j<m;j++)
{
printf("a[%d][%d]: ",i,j);
scanf("%d",&a[i][j]); }
printf("upisi elemente druge matrice: \n");
for(i=0;i<n;i++)
for(j=0;j<m;j++)
{
printf("%d b[%d][%d]:",i,j);
scanf("%d",&b[i][j]); }
for(i=0;i<n;i++)
for(j=0;j<m;j++)
{
c[i][j]=a[i][j]+b[i][j];
d[i][j]=a[i][j]-b[i][j];}
clrscr();
printf("\n zbroj ucitanih matrica je: \n");
for(i=0;i<n;i++)
{
for(j=0;j<m;j++)
printf("%d ",c[i][j]);
printf("\n");}
printf("\n razlika ucitanih matrica je: \n");
for(i=0;i<n;i++)
{
for(j=0;j<m;j++)
printf("%d ",d[i][j]);
printf("\n"); }
getch();
}
```

43. Množenje matrica

Zadatak: Učitati dvije matrice i pomnožiti ih.

Napomena: Množiti se mogu jedino dvije matrice kod kojih je broj redova druge matrice jednak broju stupaca prve matrice.

Ako je A matrica tipa (n,m), a B tipa (m,k) onda će C=A*B biti tipa (n,k).

$$c = a \cdot b = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1m} \\ a_{21} & a_{22} & \dots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nm} \end{bmatrix} \cdot \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1k} \\ b_{21} & b_{22} & \dots & b_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ b_{m1} & b_{m2} & \dots & b_{mk} \end{bmatrix} =$$

$$\begin{bmatrix} a_{11}b_{11} + a_{12}b_{21} + \dots + a_{1m}b_{m1} & \dots & a_{11}b_{1k} + a_{12}b_{2k} + \dots + a_{1m}b_{mk} \\ a_{21}b_{11} + a_{22}b_{21} + \dots + a_{2m}b_{m1} & \dots & a_{21}b_{1k} + a_{22}b_{2k} + \dots + a_{2m}b_{mk} \\ \vdots & \vdots & \vdots \\ a_{n1}b_{11} + a_{n2}b_{21} + \dots + a_{nm}b_{m1} & \dots & a_{n1}b_{1k} + a_{n2}b_{2k} + \dots + a_{nm}b_{mk} \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & \dots & c_{1n} \\ c_{21} & c_{22} & \dots & c_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{m1} & c_{m2} & \dots & c_{mn} \end{bmatrix}$$

Rješenje:

```
#include<conio.h>
#include<stdio.h>
int a[10][10],b[10][10],c[10][10],d[10][10],i,j,n,
    m,k,l;
void main(void) {
    clrscr();
    printf("upisi broj redova prve matrice: ");
    scanf("%d",&n);
    printf("upisi broj stupaca prve matrice: ");
    scanf("%d",&m);
    printf("upisi broj stupaca druge matrice: ");
    scanf("%d",&k);
    printf("upisi elemente prve matrice:\n");
    for(i=0;i<n;i++)
        for(j=0;j<m;j++)
        {
            printf("a[%d][%d]: ",i,j); scanf("%d",&a[i][j]);
        }
    printf("upisi elemente druge matrice: \n");
    for(i=0;i<m;i++)
        for(j=0;j<k;j++)
        {
            printf("b[%d][%d]:",i,j);scanf("%d",&b[i][j]);
        }
    for(i=0;i<n;i++)
        for(j=0;j<k;j++)
        {
            c[i][j]=0;
            for(l=0;l<m;l++)
                c[i][j]=c[i][j]+a[i][l]*b[l][j];
        }
    printf("\n umnozак ucitanih matrica je: \n");
    for(i=0;i<n;i++)
        {
            for(j=0;j<k;j++)
                printf("%d ",c[i][j]);
            printf("\n");
        }
    getch();
}
```

44. Pretraživanje dvodimenzionalnog polja (dijagonale)

Zadatak: Učitati jednu kvadratnu matricu reda n: Ispisati matricu u obliku tablice, te elemente na glavnoj i sporednoj dijagonali.

Napomena: Elementi glavne dijagonale su oni elementi matrice kojima je jednak indeks reda i stupca. Za elemente sporedne dijagonale vrijedi da je suma indeksa reda i stupca jednaka $n+1$, odnosno da se indeks reda povećava, a indeks stupca smanjuje za 1.

Rješenje:

```
#include<conio.h>
#include<stdio.h>
int a[10][10],b[10],c[10],i,j,n;
void main(void)
{
clrscr();
printf("upisi red matrice: ");scanf("%d",&n);
printf("upisi elemente matrice:\n");
for(i=0;i<n;i++)
for(j=0;j<n;j++)
{
printf("a[%d][%d]: ",i,j);scanf("%d",&a[i][j]);
}
for(i=0;i<n;i++)
{
b[i]=a[i][i];
c[i]=a[i][n-i-1];
}
for(i=0;i<n;i++)
{
for(j=0;j<n;j++)
printf("%d ",a[i][j]);
printf("\n");
}
printf("\n elementi na glavnoj dijagonali su: ");
for(i=0;i<n;i++) printf("%d ",b[i]);
printf("\n elementi sporedne dijagonale su: ");
for(i=0;i<n;i++) printf("%d ",c[i]);
getch();
}
```


45. Traženje najmanjeg i najvećeg elementa redova i stupaca matrice

Zadatak: Učitaj jednu kvadratnu matricu reda n. Ispiši najveći i najmanji element u svakom redu i stupcu. Dobivene elemente smjestiti u jednodimenzionalna polja.

Rješenje:

```
#include<conio.h>
#include<stdio.h>
int a[10][10],maxr[10],maxs[10],minr[10],mins[10],i,j,n;
void main(void)
{
clrscr();
printf ("upisi red matrice: ");scanf("%d",&n);
printf("upisi elemente matrice:\n");
for(i=0;i<n;i++)
for(j=0;j<n;j++) {
printf("a[%d][%d]: ",i,j); scanf("%d",&a[i][j]); }
for(i=0;i<n;i++) {
minr[i]=a[i][0]; maxr[i]=a[i][0];
mins[i]=a[0][i]; maxs[i]=a[0][i];
for (j=1;j<n;j++) {
if(a[i][j]<minr[i])minr[i]=a[i][j];
if(a[i][j]>maxr[i])maxr[i]=a[i][j];
if(a[j][i]<mins[i])mins[i]=a[j][i];
if(a[j][i]>maxs[i])maxs[i]=a[j][i]; } }
for(i=0;i<n;i++) {
for(j=0;j<n;j++)
printf("%d ",a[i][j]);
printf("\n"); }
for(i=0;i<n;i++)
printf("\n najveci u redu %d je %d, a najmanji %d ",i+1,maxr[i],minr[i]);
for(i=0;i<n;i++)
printf("\n najveci u stupcu %d je %d, a najmanji %d ",i+1,maxs[i],mins[i]);
getch();
}
```