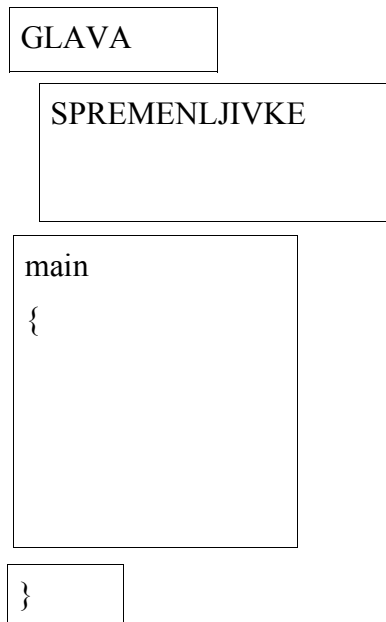


## Notranja razvejitev – strukturiranost programa

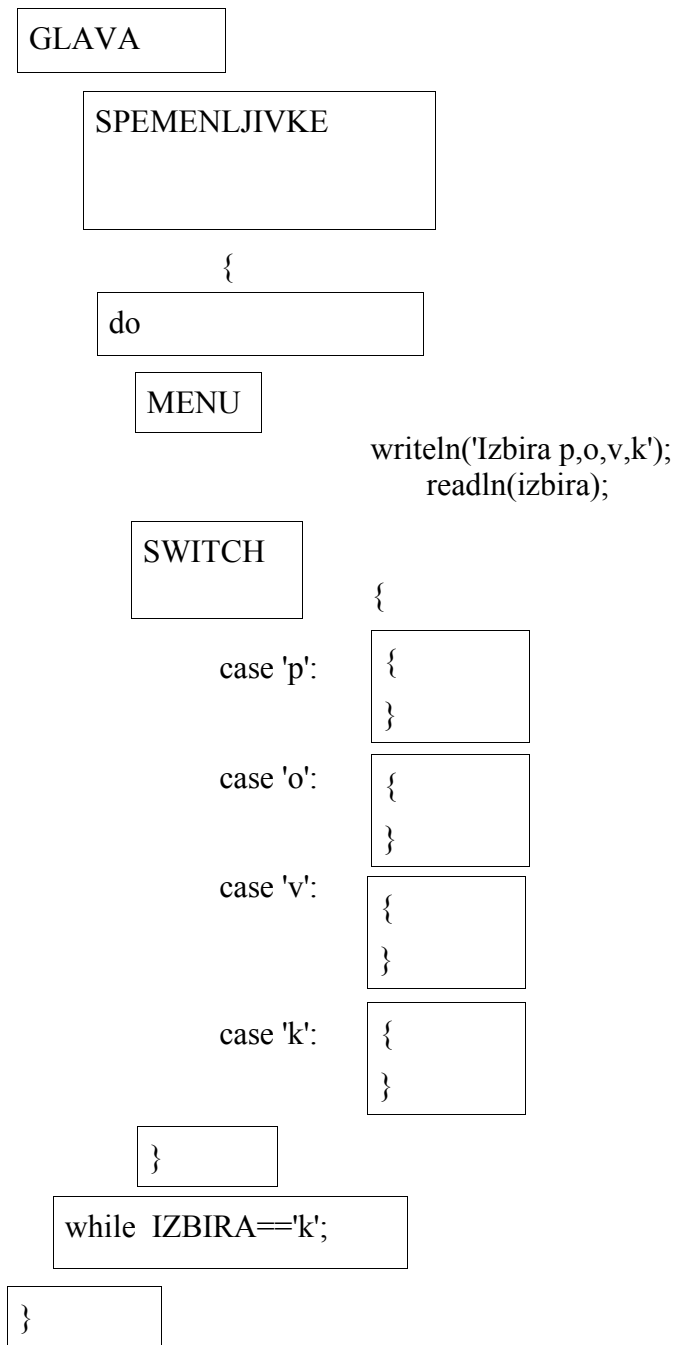
## Osnovni program



## Strukturirani program

vsebuje:

- \* menu
- \* repeat until zanko
- \* case



Posebnost strukturiranih stavkov je, da so ponovljivi, variable – spremenljivke so globalne.

## PREGLED PODATKOVNIH TIPOV V C-ju

OSNOVNI	STRUKTURIRANI
int – CELA ŠTEVILA	NIZ
float – REALNA ŠTEVILA	VEKTOR, POLJE, TABELA, 3D
bool – LOGIČEN TIP – TRUE/FALSE	ZAPIS
char – ZNAKOVNI TIP - ČRKE	DATOTEKA
	KAZALEC

To sje večina standardnih tipov podatkov, ki jih programski jezik C pozna. Poznamo še LASTNE TIPE PODATKOV, katere pa programer sam definira znotraj programa.

### int – CELOŠTEVILČNI PODATKOVNI TIP PODATKOV

1. Spada med osnovne podatkovne tipe.
2. Predstavlja interval celih števil

IME	OBMOČJE	PORABA BYTE (bit)
short	-32,768 -> +32,767	2 (16)
unsigned short	0 -> +65,53	2 (16)
unsigned int	0 -> +4,294,967,295	4 (32)
int	-2,147,483,648 -> +2,147,483,647	4 (32)
long	-2,147,483,648 -> +2,147,483,647	4 (32)

3. Deklaracija spremenljivk v C-ju:

variable(s) – spremenljivka(e)

int a;

V Cju pomeni int a; da a je ime spremenljivka, ki nosi vrednost integer in naj za a velja, da je samo celo število. A je lahko 1250, ne pa 12,50.

4. int a;  
int b; - NESMISELNO – PREVEČ NA DOLGO! Oz. pride do izraza pri več spremenljivkah  
int c; drugačne vsebine (primer rojstni dan + števci)

krajše zapišemo:

int a,b,c;

Izpis v drugih številskih sistemih:

a=10;

!!!! binarno ni mogoče izpisati (potrebno izdelati lastno funkcijo)

printf("%o\n",a); // izpis osmiško

printf("%x\n",a); // izpis šestnajstiško majhne črke

printf("%X\n",a); // izpis šestnajstiško velike črke

## Uporaba maske pri izpisu (formatiranje izpisa)

Izpis lahko tudi formatiramo, to pomeni, da izpis poravnamo na levo ali na desno stran. Rezerviramo lahko tudi določeno število znakov za izpis števila.

Izpišemo spremenljivko x poravnano na levo in rezerviramo 7 mest za izpis

```
printf("|%-7d\n",x); // | 1          |
```

Izpišemo spremenljivko x poravnano na desno in rezerviramo 7 mest za izpis

```
printf("|%7d\n",x); // |          1 |
```

Izpišemo spremenljivko x poravnano na desno in izpišemo ničle pred številom

```
printf("|%07d\n",x); // | 0000001 |
```

## Celoštevilčne računske operacije

Simboli so v C za izraz računskih operacij:

- + seštevanje
- - odštevanje
- \* množenje
- % ostanek celoštevilčnega deljenja
- / deljenje
- ++a; prištej spremenljivki 1, potem uporabi
- a++; uporabi a in nato spremenljivki prištej 1
- --a; odštej spremenljivki 1, potem uporabi
- a--; uporabi a in nato spremenljivki odštej 1
- i = j++;                i = j; j = j + 1;
- i = ++j;                j = j + 1; i = j;
- i = j--;                i = j; j = j + 1;
- i = --j;                j = j - 1; i = j;
- i += j;                i = i + j;
- i -= j;                i = i - j;
- i \*= j;                i = i \* j;
- i /= j;                i = i / j;
- i %= j;                i = i % j;

ZGLED:

10 / 4 -> 2 – število

19 % 6 -> 1 – ostanek

```
#include <stdio.h>
int main()
{
    int prvost, drugost, vsota;
    printf("Vnesi prvo stevilo: ");
    scanf("%d", &prvost);
    printf("Vnesi drugo stevilo: ");
    scanf("%d", &drugost);
    vsota = (3 / prvost) * drugost;
    printf("rezultat je %d\n", vsota);
}
```

RAZLAGA:

#include <stdio.h>; funkcije za delo z zaslonom.

printf, scanf; - ukaza za izpis na zaslon in branje iz tipkovnice

znak = pomeni naj pridobi vrednost; Torej neka spremenljivka dobi vrednost!

## Konstante v limits.h

signed char

SCHAR\_MIN -128

SCHAR\_MAX 127

unsigned char

UCHAR\_MAX 255

char

CHAR\_MIN -128

CHAR\_MAX 127

signed short

SHRT\_MIN -32768

SHRT\_MAX 32767

signed int

INT\_MIN -32768

INT\_MAX 32767

unsigned int

UINT\_MAX -1

long

LONG\_MIN -2147483648

LONG\_MAX 2147483647

unsigned long

ULONG\_MAX -1

## PODATKOVNI TIP float

Realna števila so tista števila, ki so zapisana s decimalno vejico (kot decimalna vejica velja samo pika .).

Izrazi: +, -, \*, /

Maska zapisa: %5,2f

f – spremenljivka tipa float

5 – rezervirana mesta za izpis

2 – izpis na dve decimalni številki natančno

Deklaracija:

f float;

Razširitveni tipi float:

Tip	Interval	Velikost v bytih
float	3.4E +/- 38	4 (enojna natančnost)
double	1.7E +/- 308	8 (dvojna natančnost) – najmanj tolik velik kot float
long double	1.7E +/- 308	8 (povečana natančnost) – najmanj tolik velik kot double

f float or double decimal point but no exponent (%f – 1.200000,

e or E float or double decimal point and exponent (%e – 1.200000e+000, %E – 1.200000E+000)

printf("%09.2f\n",a); - 000001.30 – če je a 1.3

Primer:

```
#include <stdio.h>
int main(void)
{
    float prvost, drugost, vsota;
    printf("Vnesi prvo stevilo: ");
    scanf("%f", &prvost); // 3
    printf("Vnesi drugo stevilo: ");
    scanf("%f", &drugost); // 4
    vsota = (3 / prvost) * drugost;
    printf("rezultat je %10.2f\n", vsota); // rezultat je _ _ _ _4.00
    return 0;
}
```

## Matematične operacije za delo nad števili (int in float)

**int abs (i)** absolutna vrednost števila i – printf("%d\n",abs(-1)); // 1  
**long labs (k)** absolutna vrednost long števila k – printf("%d\n",labs(-1)); // 1  
**double fabs (x)** absolutna vrednost float števila x – printf("%0.2f\n",fabs(1.3)); // 1.30  
**double fmod (x, y)** vrne ostanek deljenja x / y – printf("%0.2f\n",fmod(5,3)); // 2.00  
**double modf (x, pd)** razdeli število na celi in realni del  
#include <math.h>  
#include <stdio.h>

```
int main(void)
{
    double fraction, integer;
    double number = 100000.567;

    fraction = modf(number, &integer);
    printf("Stevilo %f ima celi del %f in %f\n", number, integer, fraction);
    return 0;
}
```

**double ldexp (x, i)** Vrne (x\* (2 na i) )

**double frexp (x, pi)** Vrne x = mantissa \* 2<sup>exponent</sup>

**double floor (x)** Vrne največji integer, ki se gre v x -

printf("%f\n",floor(5.1)); // 5.0                      printf("%f\n",floor(-5.1)); // 6.0

```

printf("%f\n",floor(5.6)); // 5.0
printf("%f\n",floor(-5.6)); // -6.0
double ceil (x)      Vrne najmanjši integer,ki ni manjši od x –
printf("%f\n",ceil(5.1)); // 6.0
printf("%f\n",ceil(-5.1)); // -5.0
printf("%f\n",ceil(5.6)); // 6.0
printf("%f\n",ceil(-5.5)); // -5.0
double sqrt (x)      Vrne koren števila x – printf("%f\n",sqrt(25)); // 5
double pow (x, y)     Potenca  $x^y$  – printf("%f\n",pow(2,3)); // 2*2*2 = 8
double sin (x)        Vrne sinus x
double cos(x)         Vrne cosinus x
double tan(x)         Vrne tangens x
double asin (x)       Vrne arcus sinus x
double acos (x)       Vrne arcus cosinus x
double atan (x)       Vrne arcus tangens x
double atan2 (y,x)    Vrne arcus tangens y/x. Uporabi predznaka argumentov za določitev kvadranta
double exp (x)        Vrne eksponentno vrednost x
printf("%f\n",exp(6)); // 403.428793
printf("%f\n",exp(0.5)); // 1.648721
double log (x)        Vrne naravni logaritem x
double log10 (x)      Vrne logaritem na osnovi 10 spremenljivke x

```

potrebujemo <time.h>

```

rand()              Generira naključne vrednosti
srand()            Inicializira naključni generator števil
srand((unsigned) time(NULL));

```

```
rand() % 10; // vrednosti od 0 – 9
```

```
rand() % 10 + 1; // vrednosti od 1 - 10
```

## Primeri

```

/* abs example */
#include <stdio.h>
#include <math.h>

int main ()
{
    int n,m;
    n=abs(23);
    m=abs(-11);
    printf ("n=%d\n",n); // n=23
    printf ("m=%d\n",m); // m=11
    return 0;
}

/* labs example */
#include <stdio.h>
#include <math.h>

int main ()
{
    int n,m;
    n=labs(65537);
    m=labs(-100000);
    printf ("n=%d\n",n); // n=65537
    printf ("m=%d\n",m); // m=100000
    return 0;
}

/* fabs example */
#include <stdio.h>
#include <math.h>

int main ()
{
    printf ("Abs 3.1416 is %lf\n", fabs (3.1416) ); // Abs 3.1416 is 3.141600
    printf ("Abs -10.6 is %lf\n", fabs (-10.6) ); // Abs -10.6 is 10.600000
}

```

```

    return 0;
}

/* fmod example */
#include <stdio.h>
#include <math.h>

int main ()
{
    printf ("fmod of 5.3 / 2 is %lf\n", fmod (5.3,2) ); // fmod of 5.3 / 2 is 1.300000
    printf ("fmod of 18.5 / 4.2 is %lf\n", fmod (18.5,4.2) ); // fmod of 18.5 / 4.2 is 1.700000
    return 0;
}

/* modf example */
#include <stdio.h>
#include <math.h>

int main ()
{
    double param, fractpart, intpart;

    param = 3.14159265;
    fractpart = modf (param , &intpart);
    printf ("%lf=%lf+%lf \n", param, intpart, fractpart); // 3.141593=3.000000+0.141593
    return 0;
}

/* floor primer */
#include <stdio.h>
#include <math.h>

int main ()
{
    printf ("floor of 2.3 is %.11f\n", floor (2.3) );
    printf ("floor of 3.8 is %.11f\n", floor (3.8) );
    printf ("floor of -2.3 is %.11f\n", floor (-2.3) );
    printf ("floor of -3.8 is %.11f\n", floor (-3.8) );
    return 0;
}

/* ceil primer */
#include <stdio.h>
#include <math.h>

int main ()
{
    printf ("ceil of 2.3 is %.11f\n", ceil (2.3) );
    printf ("ceil of 3.8 is %.11f\n", ceil (3.8) );
    printf ("ceil of -2.3 is %.11f\n", ceil (-2.3) );
    printf ("ceil of -3.8 is %.11f\n", ceil (-3.8) );
    return 0;
}

/* log example */
#include <stdio.h>
#include <math.h>

int main ()
{
    double param, result;
    param = 1000.0;
    result = log10 (param);
    printf ("log10(%lf) = %lf\n", param, result ); // log10(1000.000000) = 3.000000
    return 0;
}

/* log example */
#include <stdio.h>
#include <math.h>

```

```

int main ()
{
    double param, result;
    param = 5.5;
    result = log (param);
    printf ("ln(%lf) = %lf\n", param, result ); // ln(5.500000) = 1.704748
    return 0;
}

/* ceil example */
#include <stdio.h>
#include <math.h>

int main ()
{
    printf ("ceil of 2.3 is %.1lf\n", ceil (2.3) ); // ceil of 2.3 is 3.0
    printf ("ceil of 3.8 is %.1lf\n", ceil (3.8) ); // ceil of 3.8 is 4.0
    printf ("ceil of -2.3 is %.1lf\n", ceil (-2.3) ); // ceil of -2.3 is -2.0
    printf ("ceil of -3.8 is %.1lf\n", ceil (-3.8) ); // ceil of -3.8 is -3.0
    return 0;
}

```

## Znakovni tip podatkov – char

Vsak znak na tipkovnici ima svoj znak. Ti znaki so razvrščeni v ASCII tabelo. ASCII koda črke A je 065, a je 097.

Podatek tipa char – znak je podatkovni tip, ki določa vsebino ASCII tabele. V ASCII tabeli je razvrščeno 256 znakov.

Deklaracija:

char a;

int main(void)

```

{
    char a;
    a = 'b';
    return 0;
}

```

Branje in izpisovanje znakov

```

#include <stdio.h>
int main()
{
    char p;
    printf("Vnesi znak: ");
    scanf("%c", &p);
    printf("Vnesu si %c\n", p);
    return 0;
}

```

```

#include <stdio.h>
int main()
{
    char p;
    printf("Vnesi znak: ");
    p = getchar();
    printf("%c\n", p);
    return 0;
}

```

Standardne funkcije in procedure za delo z znaki (knjižnica ctype.h):



int toupper(int a);	- iz malih pretvori v velike črke
int tolower(int a);	- iz velikih pretvori v male črke
int isalnum(int c);	
int isalpha(int c);	- 0 če je številka ali poseben znak, 8 če je majhna črka, 4 če je velika črka
int iscntrl(int c);	
int isdigit(int c);	- 0 ko ni številka, 2 ko je številka
int isgraph(int c);	
int islower(int c);	- 0 ko ni majhna črka, 8 ko je majhna črka
int isprint(int c);	
int ispunct(int c);	
int isspace(int c);	- 0 ni presledek, 1 je presledek
int isupper(int c);	- 0 mala črka, 4 velika črka
int isxdigit(int c);	

Zgledi!!!

## Konstante

***Če je vrednost v programu konstantna jo deklariramo kot konstanto!***

```
#define PI 3.14
```

```
#define N 5
```

```
#define TEKST "To je moj konstanten tekst"
```

Naloge:

1. Izdelaj program, ki izračuna vsoto dveh celih števil.
2. Izdelaj program, ki izračuna razliko dveh celih števil.
3. Izdelaj program, ki izračuna produkt dveh celih števil.
4. Izdelaj program, ki izračuna količnik dveh celih števil.
5. Izdelaj program, ki izračuna vsoto, razliko, produkt in količnik dveh realnih števil.
6. Izdelaj program, ki izpiše ASCII kodo pritisnjene tipke.
7. Izdelaj program, ki izpiše ASCII kodo predhodnjika in naslednjika pritisnjene tipke.
8. Izdelaj program, ki izpiše celotno ASCII tabelo
9. Izdelaj program, ki izpiše celotno ASCII kodo in pripadajoč znak.