





## P2: Treća nedelja

### Uvod u programski jezik C

## Uvodni zadatak – Zdravo svete!

- Sastaviti na jeziku C program koji ispisuje poruku na glavnom izlazu.

- Rešenje:

```
/* pozdrav.c - Ispisivanje pozdrava */  
#include <stdio.h>  ukljućivanje funkcije za ispis  
  
void main () {  početak bloka  
  
    printf ("Zdravo svete!\n");  poziv funkcije za ispis  
  
}  kraj bloka
```

## Uvodni zadatak – Opšte napomene

- Programi na jeziku C se sastoje od funkcija
- Svaki program na jeziku C mora sadržati funkciju `main()` koja se automatski poziva prilikom pokretanja programa
- Svaka naredba na jeziku C se završava znakom `;` (tačka-zarez)
- Ukoliko se u programu koriste bibliotečke funkcije, kao što je funkcija za ispis, na početku programa mora postojati direktiva `#include` sa odgovarajućim parametrom, što predstavlja uputstvo prevodiocu da pročita odgovarajuće opise bibliotečkih funkcija

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

## Zadatak 1 – Zbir brojeva (1.2)

- Sastaviti na jeziku C program koji pročita dva cela broja sa glavnog ulaza računara, izračuna njihov zbir i ispiše rezultat na glavnom izlazu računara.

- Rešenje:

```
/* zbirl.c - Izracunavanje zbira dva cela broja. */  
#include <stdio.h>  
  
void main () {  
    int a, b, c;  
    printf ("Unesite dva cela broja: ");  
    scanf ("%d%d", &a, &b);  
    c = a + b;  
    printf ("Zbir unetih brojeva: %d\n", c);  
}
```

## Zadatak 1 – Promenljive

### • Deklaracije promenljivih

- dozvoljene samo na početku bloka
- identifikatori mogu počinjati malim slovom, velikim slovom ili znakom `_`, a mogu sadržati pored navedenih znakova i cifre
- poželjno je da ime odslikava upotrebu promenljive
- navodi se najpre tip, a zatim imena promenljivih, odvojena zarezom  
`int suma;`  
`float x,y;`
- Jezik C pravi razliku između malih i velikih slova (eng. *case-sensitive*), za razliku od jezika Pascal!

### • Često korišćeni prosti tipovi

- **char, int, long** – celobrojni tipovi podataka
- **float, double** – realni tipovi podataka, jednostruke i dvostruke preciznosti u skladu sa ANSI/IEEE 754 standardom za realnu aritmetiku
- **ne postoji logički tip**, celobrojna vrednost 0 predstavlja logičku neistinu, a sve ostale vrednosti istinu

## Zadatak 1 – Ispis (nekompletan)

### • Funkcija printf (print formatted)

`printf(format, izraz, ..., izraz);`

### • Štampa znakovni niz prema zadatom formatu

- format se zadaje unutar znakova `" "`, a precizan opis, za ispis pojedinačnog izraza, počinje znakom `%`

### • Podrazumevani format je doslovno prenošenje navedenih znakova

- `printf("Zdravo!");` ispisuje **Zdravo!** na ekranu

### • Specijalni simboli:

- `\n` novi red
- `\t` tabulacija (ostavljanje više praznina radi poravnavanja)
- `\\` znak `\`
- `\"` znak `"`
- `%%` znak `%`

## Zadatak 1 – Ispis (2)

### • Ispis vrednosti promenljivih:

- `%c` slovo (simbol, odnosno karakter)
- `%d` ceo broj u decimalnom brojnom sistemu
- `%o` ceo broj u oktalanom brojnom sistemu
- `%x` ceo broj u heksadecimalnom brojnom sistemu
- `%f` realan broj
- `%s` znakovni niz

### • Specifično formatiranje

- `%nd` ispisuje broj na najmanje *n* mesta (višak popunjava razmacima)
- `%.nf` ispisuje realan broj u formatu x.y (najviše *n* decimalnih mesta)

### • Primer

```
int a = 1;
printf("a = %d", a);
```

 ispisuje **a = 1** na ekranu

## Zadatak 1 – Unos

### • Funkcija scanf (scan formatted)

`scanf(format, &podatak, ..., &podatak);`

### • Učitava podatak sa glavnog ulaza u skladu sa zadatim formatom za pojedinačni podatak

### • Podatak predstavlja promenljivu u koju će biti smešten rezultat učitavanja

- operator **&** se koristi za uzimanje adrese promenljive u koju se smešta rezultat učitavanja

### • Format sličan kao kod printf funkcije, i obavezno je navesti ga za svaki podatak

### • Primer

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

## Zadatak 2 – Sintaksa

- Sledeći program za određivanje rešenja linearne jednačine  $Ax+B=0$ , za  $A \neq 0$ , napisan na programskom jeziku C, sadrži više grešaka. Ispraviti sve greške!

```
main
Float _x,a,b
{
    scanf('%f %f',a,b);
    _x=-b/a;
    printf('%f',_X)
}
```

## Zadatak 2 – Pokušaj prevođenja

- Ukoliko pokušamo da prevedemo dati program, dobijamo sledeće poruke o grešci\*:

```
Compiling...
c5.c
c5.c(5) : error C2061: syntax error : identifier 'Float'
c5.c(5) : error C2059: syntax error : ';'
c5.c(7) : error C2143: syntax error : missing ';' before '{'
c5.c(7) : error C2449: found '{' at file scope (missing
        function header?)
c5.c(8) : error C2015: too many characters in constant
c5.c(14) : error C2059: syntax error : '}'
```

\* za prevođenje je korišćen prevodilac Visual Studio.net

## Zadatak 2 – Rešenje

```
main
Float _x,a,b
{
    scanf('%f %f',a,b);
    _x=-b/a;
    printf('%f',_X)
}
```

```
#include <stdio.h>
main()
{
    float _x,a,b;

    scanf("%f %f",&a,&b);
    _x=-b/a;
    printf("%f",_x);
}
```

**Greške:**  
**1)direktiva #include**  
**2)main**  
**3)Float**  
    **(mala/velika slova)**  
**4)format**  
    **kod scanf/printf**  
**5)adresa promenljive**

## Operatori

- **Operatori – radnje koje se izvršavaju nad operandima i proizvode rezultat**
  - unarni, binarni, ternarni
- **Izrazi se sastavljaju od operandi i operatora**
  - $a = b + c;$
- **Redosled izvršavanja operatora**
  - prvenstveno određen upotrebom oblika zagrada ( )
  - zagrade se mogu ugnežđavati do proizvoljne dubine
  - unutar zagrada, operatori se izvršavaju prema redosledu prioriteta
  - ako više operatora u izrazu ima isti prioritet, izvršiće se sleva na desno ili sdesna na levo, u zavisnosti od smeru grupisanja operatora na tom nivou prioriteta
- **Vrste: adresni, aritmetički, relacijski, bitski, logički, dodele vrednosti...**

## Operatori – prioritet i smer grupisanja

PRIORITET	BROJ OPERANADA	OPERATORI	SMER GRUPISANJA
15	2	[ ] ( ) . ->	→
14	1	! ~ ++ -- + - * & (tip) sizeof	←
13	2	* / %	→
12	2	+ -	→
11	2	<< >>	→
10	2	< <= > >=	→
9	2	== !=	→
8	2	&	→
7	2	^	→
6	2		→
5	2	&&	→
4	2		→
3	3	?:	←
2	2	= += -= *= /= %= &= ^=  = <<= >>=	←
1	2	,	→

## Relacioni operatori

Izraz	Objašnjenje (rezultat)
5 > 7	0
10 <= 20	1
8 == 13 > 5	8==(13>5) -> 8==1 -> 0
14 > 5 < 3	(14>5)<3 -> 1<3 -> 1
a < b < 5	(a<b)<5 -> (0/1)<5 -> 1
a+5 >= c-1.0/e	(a+5) >= (c-(1.0/e))

## Operatori dodele vrednosti

Izraz	Objašnjenje
y = a * x + b	y = ((a * x) + b)
d *= e + f	d = (d *(e+f))
d = d * e + f	d = ((d*e)+f)
a=b=c=d+5	c=d+5, b=c, a=b
a = b++ + 3*(c=d<<3)	c=d<<3, u=b, b=b+1, a=u+(3*c)
a = b++ + 3*c = d<<3	greška ! ((b++) + (3*c)) =(d<<3) ???

## Operatori – redosled računanja i bočni efekti (1)

**((a<<5)+4/b)\*(d=c-2) \* ++e**

### • Standard

- pre množenja računa se vrednost operanada u zagradama
- ne precizira kojim će se redom računati (prvo levi ili prvo desni operand) – implementaciona zavisnost
- rezultat koji zavisi od redosleda izračunavanja - loš stil pisanja izraza

### • Bočni efekti: uzgredna promena vrednosti jedne ili više promenljivih

- rezultat zavisi od redosleda izračunavanja operanada ako se bar jedna promenljiva, koja podleže uticaju nekog bočnog efekta, koristi na više od jednog mesta u izrazu
- operatori koji proizvode bočne efekte su ++, -- i svi operatori dodele vrednosti.

## Operatori – redosled računanja i bočni efekti (2)

Izraz	Objašnjenje
$a=b*c + b++$	$u=b*c, v=b, b=b+1, a=u+v$
	$v=b, b=b+1, u=b*c, a=u+v$
$a=b*c + b=d/e$	$u=b*c, v=b=d/e, a=u+v$
	$v=b=d/e, u=b*c, a=u+v$
$a[i] = ++i$	Indeksiranje je bin.op.
	$u=i, ++i, v=i, a[u]=v$
	$i++, v=i, u=i, a[u]=v$

## Operatori – redosled računanja i bočni efekti (3) PRIMER

Bočni efekti u 8 različitih izraza, za različite prevodioce. Inicijalna vrednost promenljive  $j$  je 5.

	izraz	VS 6.0	VS.NET 2003	TC 2.0	GNU GCC	ARM C
1.	$j * j++$	25	25	30	25	30
2.	$j++ * j$	25	25	30	25	30
3.	$++j * j$	36	36	36	36	36
4.	$j * ++j$	36	36	36	36	36
5.	$(j=3)*(j++)$	9	9	9	9	9
6.	$(j=3)*(++j)$	16	16	12	12	12
7.	$(j++)*(j=3)$	9	9	15	9	15
8.	$(++j)*(j=3)$	9	9	18	9	18

?

## Zadatak 3 – Operatori (C15)

- Odrediti čemu su ekvivalentni sledeći izrazi (koristiti zagrade da bi se eksplicitno odredio redosled izračunavanja)

1.  $x+=y-=m$

$$x=(x+(y=(y-m)))$$

2.  $n\%=y+m$

$$n=(n \% (y+m))$$

## Zadatak 3

3.  $m++ - --j$

Osnovni rezultat:  $m-(j-1)$ ;

Bočni efekti:  $m=m+1; j=j-1$

4.  $x=j * j++$

$$x=j * j++ ; ???$$

(implementaciono zavisno!)

5.  $++j==m!=y*2$

$$((j+1)==m)!=(y*2); j=j+1$$

## Zadatak 4 – Površina trougla (1.4)

- Sastaviti na jeziku C program za izračunavanje površine trougla u ravni ako su zadate koordinate temena. Potrebne podatke čitati sa glavnog ulaza računara, a rezultat prikazati na glavnom izlazu.

- Važe sledeće formule:

$$A(x_A, y_A), B(x_B, y_B), C(x_C, y_C)$$

$$a = \sqrt{(x_B - x_C)^2 + (y_B - y_C)^2}$$

$$s = (a + b + c) / 2$$

$$b = \sqrt{(x_C - x_A)^2 + (y_C - y_A)^2}$$

$$P = \sqrt{s(s-a)(s-b)(s-c)}$$

$$c = \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2}$$

## Zadatak 4

- Rešenje

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

void main () {

    double xA, yA, xB, yB, xC, yC, a, b, c, s, P;

    /* Temena trougla: */
    printf ("Koordinate temena trougla\n");
    printf ("- prvo teme? ");
    scanf ("%lf%lf", &xA, &yA);
    printf ("- drugo teme? ");
    scanf ("%lf%lf", &xB, &yB);
    printf ("- trece teme? ");
    scanf ("%lf%lf", &xC, &yC);
```

## Zadatak 4

```
/* Stranice trougla: */
a = sqrt (pow (xB-xC, 2) + pow (yB-yC, 2));
b = sqrt (pow (xC-xA, 2) + pow (yC-yA, 2));
c = sqrt (pow (xA-xB, 2) + pow (yA-yB, 2));

/* Površina trougla: */
s = (a + b + c) / 2;
P = sqrt (s * (s-a) * (s-b) * (s-c));
printf ("Povrsina trougla: %f\n", P);

}
```

## Zadatak 5 – if - else (2.3)

- Sastaviti na jeziku C program za rešavanje sistema od dve linearne jednačine s dve nepoznate.

$$a_1x + b_1y = c_1$$

$$a_2x + b_2y = c_2$$

- Važi:

$$D = \begin{vmatrix} a_1 & b_1 \\ a_2 & b_2 \end{vmatrix} = a_1b_2 - a_2b_1$$

$$D \neq 0: x = \frac{D_x}{D}, y = \frac{D_y}{D}$$

$$D_x = \begin{vmatrix} c_1 & b_1 \\ c_2 & b_2 \end{vmatrix} = c_1b_2 - c_2b_1$$

$$D = D_x = D_y = 0$$

$$D_y = \begin{vmatrix} a_1 & c_1 \\ a_2 & c_2 \end{vmatrix} = a_1c_2 - a_2c_1$$

rešenje neodređeno,  
inače protivrečno

## Zadatak 5

### • Rešenje

```
#include <stdio.h>

void main () {
    double a1, b1, c1, a2, b2, c2, D, Dx, Dy, x, y;

    printf ("Koeficijenti prve jednacine? ");
    scanf ("%lf%lf%lf", &a1, &b1, &c1);
    printf ("Koeficijenti druge jednacine? ");
    scanf ("%lf%lf%lf", &a2, &b2, &c2);

    D  = a1 * b2 - a2 * b1;
    Dx = c1 * b2 - c2 * b1;
    Dy = a1 * c2 - a2 * c1;
```

## Zadatak 5

```
if (D != 0) {

    x = Dx / D;
    y = Dy / D;
    printf ("x= %10.2f\n", x);
    printf ("y= %10.2f\n", y);

} else
    if (Dx==0 && Dy==0)
        printf ("Sistem je neodredjen.\n");
    else
        printf ("Sistem je protivrecan.\n");

}
```

## Zadatak 6 – for (2.6a,c)

- Sastaviti na jeziku C programe za izračunavanje vrednosti sledećih izraza:

$$a) s = 1 + 2 + 3 + \dots + n = \sum_{i=1}^n i$$

$$c) s = 1! + 2! + 3! + \dots + n! = \sum_{i=1}^n i! = \sum_{i=1}^n \prod_{j=1}^i j$$

## Ciklusi - for

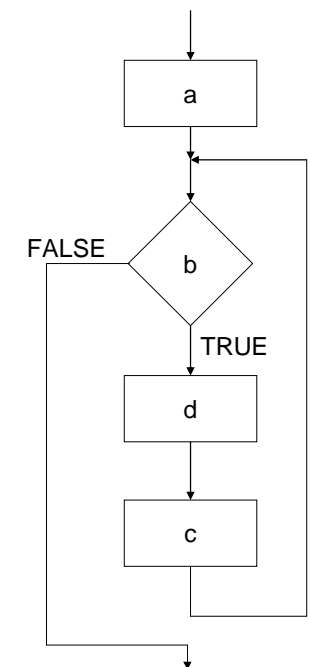
Opšti oblik:

**for( a; b; c ) d;**

Polja **a**, **b**, **c** i **d** su opcionala.  
Ako se ne zada **b** podrazumeva se da je uslov uvek ispunjen.

Beskonačan ciklus:

**for( ; ; ) { ... }**



## Zadatak 6

- **Rešenje a)**

```
#include <stdio.h>
void main () {
    int n, s, i;
    printf ("n? "); scanf ("%d", &n);
    for (s=0, i=1; i<=n; s+=i++);
    printf ("s= %d\n", s);
}
```

- **Rešenje c)**

```
#include <stdio.h>
void main () {
    int n, i; long s, f;
    printf ("n? "); scanf ("%d", &n)
    for (s=0, f=i=1; i<=n; i++) {
        f *= i;
        s += f;
    }
    printf ("s= %ld\n", s);
}
```

## Primer

- **Sastaviti program na programskom jeziku C za ispisivanje tablice ASCII kodova za sve štampanje znakove.**

### Rešenje:

- **ASCII kod sa brojem 32 je blanko simbol ili razmak. Prva 32 ASCII koda (od 0 do 31) i svi ASCII kodovi veći ili jednaki 127 predstavljaju specijalne simbole koji se koriste u različite svrhe i najčešće nemaju neko jasno slovo značenje.**
- **ASCII kodovi između 32 i 126 su štampanje znaci i ima ih ukupno 95.**
- **Znaci se prikazuju po kolonama. U jednoj koloni se prikazuje po 19 znakova, a u jednom redu se prikazuje po 5 znakova (5x19=95).**

## Primer – rešenje

```
#include<stdio.h>
main()
{
    char c;
    int i;
    printf("\tTablica ASCII kodova\n\n");
    for(c = ' '; c < ' ' + 19; c++)
    {
        for(i = 0; i < 95; i += 19)
            printf("%3d %c\t", c+i, c+i);
        putchar('\n');
    }
}
```

## Primer – rešenje

Tablica ima:  
-19 vrsta (spoljni for ciklus)  
-5 kolona (unutrašnji for ciklus)

Tablica prikazuje sve simbole  
ASCII koji imaju  
čitljivu grafičku reprezentaciju.

Deo tablice ASCII kodova

32	51	3	70	F	89	Y	108	l	
33	!	52	4	71	G	90	Z	109	m
34	"	53	5	72	H	91	[	110	n
35	#	54	6	73	I	92	\	111	o
36	\$	55	7	74	J	93	]	112	p
37	%	56	8	75	K	94	^	113	q
38	&	57	9	76	L	95	_	114	r
39	'	58	:	77	M	96	`	115	s
40	(	59	;	78	N	97	a	116	t
41	)	60	<	79	O	98	b	117	u
42	*	61	=	80	P	99	c	118	v
43	+	62	>	81	Q	100	d	119	w
44	,	63	?	82	R	101	e	120	x
45	-	64	@	83	S	102	f	121	y
46	.	65	A	84	T	103	g	122	z
47	/	66	B	85	U	104	h	123	{
48	0	67	C	86	V	105	i	124	
49	1	68	D	87	W	106	j	125	}
50	2	69	E	88	X	107	k	126	~



## Zadatak 7 – while (C25)

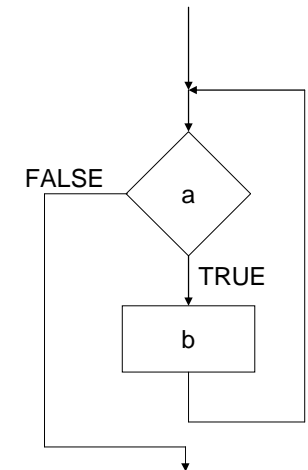
- Sastaviti program na programskom jeziku C za određivanje broja velikih slova, malih slova i cifara u tekstu koji se iz proizvoljnog broja redova učitava preko glavne ulazne jedinice. Tekst se završava znakom za kraj datoteke.
- U rešenju zadatka će biti prikazano:
  - korišćenje **while** ciklusa
  - korišćenje funkcije `getchar()`
  - korišćenje funkcija iz biblioteke `<ctype.h>`
  - korišćenje rezultata relacionih i logičkih operatora

## Ciklusi - while

Opšti oblik:

`while(a) b;`

- izraz **a** je obavezan
- instrukcija (blok) **b** je opcion
- uslov ciklusa je ispunjen ako je izraz **a** tačan (tj. vrednost  $\neq 0$ )



## Zadatak 7

- Rešenje:

```
#include <stdio.h>

void main() {
    int znak, vel_sl=0, mal_sl=0, cifara=0;

    printf("Unesite zeljeni tekst \n");
    while( ( znak = getchar() ) != EOF ) {
        vel_sl += ( znak >= 'A' && znak <= 'Z' );
        mal_sl += ( znak >= 'a' && znak <= 'z' );
        cifara += ( znak >= '0' && znak <= '9' );
    }
    printf("Velikih slova ima %d\n", vel_sl);
    printf("Malih slova ima %d\n", mal_sl);
    printf("Cifara ima %d\n", cifara);
}
```

## Funkcija getchar()

- Funkcija iz biblioteke `<stdio.h>`

```
int getchar();
```
- Čita jedan znak (karakter) sa standardnog ulaza i kao rezultat vraća kôd koji predstavlja ishod njenog izvršenja
- Zašto je rezultat int, kada se čita jedan karakter?
- Neka `sizeof(x)` označava dužinu u bajtovima na kojoj se podatak tipa *x* smešta u memoriju
- Standard jezika C propisuje:

```
sizeof(char) = 1
sizeof(char) ≤ sizeof(short) ≤ sizeof(int) ≤ sizeof(long)
```
- `getchar()` mora da vraća kôd koji izlazi iz opsega `char` da bi se označile specijalne situacije (npr. kraj datoteke)

## Zaglavlje <ctype.h>

- U zaglavlju <ctype.h> se nalaze prototipovi funkcija za ispitivanje znakova
- Često korišćene funkcije:
  - **isupper(znak)** - ispituje da li je zadovoljen uslov  
(znak>='A') && (znak<='Z')
  - **islower(znak)** - ispituje da li je zadovoljen uslov  
(znak>='a') && (znak<='z')
  - **isdigit(znak)** - ispituje da li je zadovoljen uslov  
(znak>='0') && (znak<='9')
- Sve **is\_** funkcije vraćaju 0 za logičku neistinu, a različito od 0 (ne obavezno 1) za logičku istinu
- Na jeziku C, karakteri se tretiraju kao mali celi brojevi!

## Zadatak 7

- Rešenje uz korišćenje funkcija iz <ctype.h>

```
#include <stdio.h>
#include <ctype.h>
void main() {
    int znak, vel_sl=0, mal_sl=0, cifra=0;

    printf("Unesite zeljeni tekst \n");
    while ((znak=getchar())!=EOF) {
        vel_sl += isupper(znak) != 0; /* ctype.h */
        mal_sl += islower(znak) != 0; /* ctype.h */
        cifra += isdigit(znak) != 0; /* ctype.h */
    }
    printf("Velikih slova ima %d\n", vel_sl);
    printf("Malih slova ima %d\n", mal_sl);
    printf("Cifara ima %d\n", cifra);
}
```

## Upotreba operatora

- Relacioni operatori

- a < b    a <= b
- a == b   a != b
- a >= b   a > b

- Rezultat: 0 ako relacija nije tačna, 1 ako jeste

!!! Česta greška:

a = b    Dodela vrednosti  
a == b   Poređenje na jednakost

- Logički operatori

- a && b    Logičko "i"
- a || b    Logičko "ili"

- Rezultat: 0 ako logički uslov nije tačan, 1 ako jeste

- Ne mešati logičke operatore sa bitskim:

- a & b    Bitsko "i"
- a | b    Bitsko "ili"
- Rezultat: zavisi od binarne reprezentacije a i b

## Zadatak 8 – continue (C40a)

- Šta ispisuje sledeći program?

```
#include <stdio.h>
void main(){
    int x;
    for (x = 0; x < 100; x++) {
        if ( x % 2 )
            continue;
        printf("%d\n", x);
    }
}
```

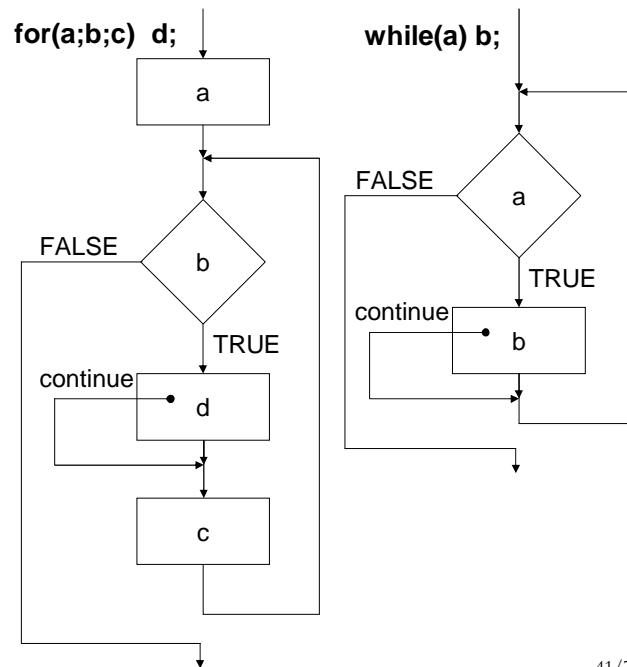
- Odgovori:

- a) Sve cele brojeve od 0 .. 99
- b) Sve parne brojeve od 0 .. 99
- c) Sve neparne brojeve od 0 .. 99

## Continue

### Naredba continue

- sme da se pojavi samo u okviru ciklusa
- prekida tekuću iteraciju ali ne napušta ciklus
- odnosi se na najbliži obuhvatajući ciklus



## Zadatak 8

### • Rešenje:

```
for (x = 0; x < 100; x++) {  
    if ( x % 2 )  
        continue;  
    printf("%d\n", x);  
}
```

- U svakoj iteraciji u kojoj je uslov  $x \% 2$  tačan (tj.  $\neq 0$ ), štampanje vrednosti  $x$  će biti preskočeno.
- $x \% 2$  će biti tačno kada  $x$  nije parno
- Prema tome, ispisivaće brojeve 0, 2, 4, ... , 98
- Odgovor: B

## Zadatak 9 – break (C40b)

### • Šta ispisuje sledeći program?

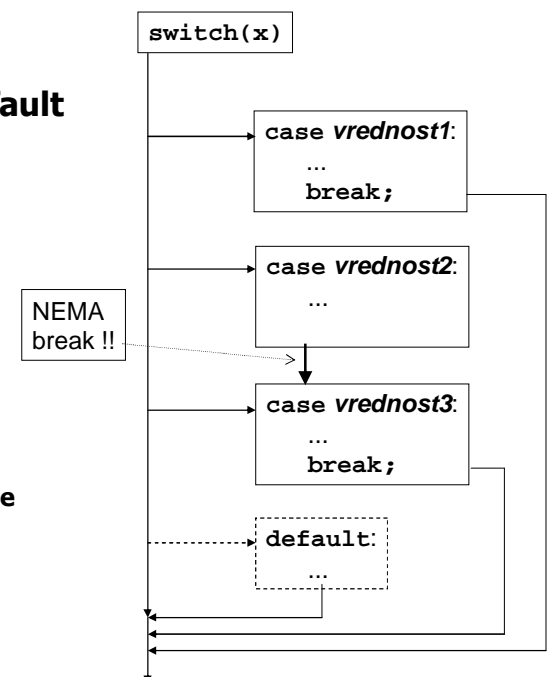
```
#include <stdio.h>  
void main() {  
    int x=0,i;  
    for (i=0;i<5;i++)  
        switch (i) {  
            case 1: x+=1;  
            case 2: x+=2; break;  
            case 4: x+=3;  
        }  
    printf("d=%d\n",x);  
}
```

- a) 5
- b) 6
- c) 9

## Switch

### switch/case/break/default

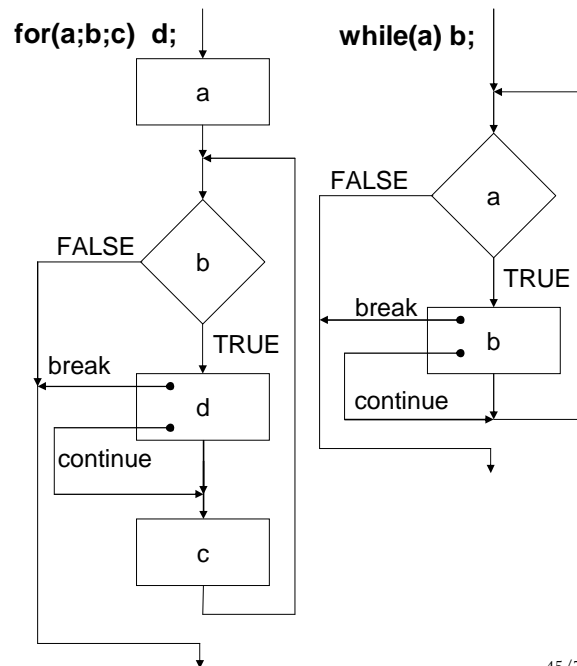
- kontrolna struktura sa višestrukim grananjem
- grananje se vrši na osnovu celobrojne vrednosti
- izvršava se grana označena odgovarajućom celobrojnou konstantom
- nakon izvršenja grane koja se ne završava sa **break** izvršava se naredna grana (ako postoji)
- grana **default** se izvršava ako ne postoji odgovarajuća grana. Grana default je **opciona**



## Break

### Naredba break

- može se pojaviti u okviru ciklusa ili u okviru switch
- u okviru ciklusa: prekida ciklus
- u okviru switch: napušta blok switch
- odnosi se na najbližu obuhvatajuću kontrolnu strukturu



## Zadatak 9 – rešenje

```
for (i=0;i<5;i++)
    switch (i)
    {
        case 1: x+=1;
        case 2: x+=2; break;
        case 4: x+=3;
    }
```

i	Kod koji se izvršava	x
0	/	0
1	x+=1; x+=2;	3
2	x+=2;	5
3	/	5
4	x+=3;	8

Tačan odgovor: N

## Zadatak 10 – datumi (2.10)

- Sastaviti na jeziku C program za određivanje narednog datuma u odnosu na zadati dan. Program treba da čita datume i da ispisuje rezultate sve dok za jednu od komponentata datuma ne pročita nulu.
- Zadatak ilustruje upotrebu:
  - while ciklusa
  - if - else kontrolne strukture
  - switch kontrolne strukture

## Zadatak 10

• Rešenje:

```
#include <stdio.h>

void main () {
    int dan, mes, god, d;

    while (1) {
        printf ("Danas? ");
        scanf ("%d%d%d", &dan, &mes, &god);
        if (dan==0 || mes==0 || god==0) break;

        /* Broj dana u tekucem mesecu: */
        switch (mes) {
            case 1: case 3: case 5: case 7: case 8: case 10: case 12:
                d = 31; break;
            case 4: case 6: case 9: case 11:
                d = 30; break;
            case 2:
                d = 28 + (god%4==0 && god%100!=0 || god%400==0); break;
        }
    }
}
```

## Zadatak 10

```
/* Obrazovanje datuma za sledeci dan: */
if (dan < d) dan++;
else {
    dan = 1;
    if (mes < 12) mes++;
    else { mes = 1; god++; }
}
printf ("Sutra= %2.2d. %2.2d. %d.\n\n",dan,mes,god);
}
```

## Zadatak 11 – nizovi (3.3)

- Sastaviti na jeziku C program koji pročita dva vektora zadate dužine sa realnim komponentama, izračuna njihov skalarni proizvod i ispisuje dobijeni rezultat. Prethodne radnje ponavlja sve dok se za dužinu vektora ne pročita nedozvoljena vrednost.
- Skalarni proizvod:

$$s = \sum_{i=0}^{n-1} a_i b_i$$

## Zadatak 11

- Rešenje:

```
#include <stdio.h>
#define DIM 50
void main () {
    double a[DIM], b[DIM], skal_pro;
    int i, n;
    while (1) {
        printf ("\nDuzina vektora (najvise %d)? ", DIM);
        scanf ("%d", &n);
        if (n <= 0 || n > DIM) break;
        printf ("Komponente vektora A? ");
        for (i=0; i<n; scanf ("%lf", &a[i++]));
        printf ("Komponente vektora B? ");
        for (i=0; i<n; scanf ("%lf", &b[i++]));
        for (skal_pro=i=0; i<n; i++)
            skal_pro += a[i] * b[i];
        printf ("Skalarni proizvod A*B= %10.3f\n", skal_pro);
    }
}
```

## Pretprocesor

- Pretprocesor

- vrši analizu izvornog koda programa pre prevodioca i sprovodi odgovarajuće tekstualne zamene
- svaku sekvencu koja odgovara postojećoj definiciji zamenjuje desnom stranom definicije
- najčešće je pretprocesor deo prevodioca

- Pretprocesorska direktiva #include

- prepisuje (uključuje) sadržaj zaglavlja koje je navedeno između znakova < > ili " " u izvorni kod
- ukoliko je ime zaglavlja navedeno unutar " ", pretprocesor će ga najpre tražiti u direktorijumu gde se nalazi izvorni kod programa, a zatim u sistemskim direktorijumima
- ako je zaglavlje navedeno unutar < >, pretprocesor će ga tražiti samo u sistemskim direktorijumima

- Pretprocesorska direktiva #define

- najčešće služi za definisanje simboličke konstante
- može da se parametrizuje
- razne druge primene (uslovno prevođenje, ... )

## Pretprocesor

- **Primeri**

```
- #define BROJ_MESECI    12
    srednja_temp /= BROJ_MESECI; ⇒ srednja_temp /= 12;
- #define EVER          ;;
    for(EVER) ⇒ beskonačan ciklus
- #define FOREVER       for(;;)
    FOREVER { ... } ⇒ beskonačan ciklus
```

- **Oprez !!**

```
- #define KVARDAT(x)      (x)*(x)
    int n = 5;
    printf("%d", KVADRAT(n++));
```

posle pretprocesora se zapravo dobija  
`printf("%d", (n++)*(n++));` ⇒ **ispisaće 30 a ne 25!**

```
- #define MAX_DUZ 100;
    int niz[MAX_DUZ];
```

posle pretprocesora se zapravo dobija  
`int niz[100];` ⇒ **greška u prevodenju!!**

## Zadatak 12 – rand i sort (3.11)

- **Sastaviti program na programskom jeziku C koji formira slučajan celobrojni niz sastavljen od jednocifrenih brojeva i izvrši uređivanje niza po neopadajućem redosledu vrednosti brojeva. Postupak ponavljati sve dok se za dužinu niza ne unese nedozvoljena vrednost.**

- **Rešenje prikazuje:**

- korišćenje funkcije `rand()` za dobijanje slučajnog broja
- algoritam za uređivanje po metodi izbora (selection sort)

## Zadatak 12

- **Rešenje:**

```
#include <stdio.h>
#include <stdlib.h>
#define DIM 50
```

```
void main() {
    int n, a[DIM], i, j, b;
```

```
    while (1) {
        printf("\n\n Duzina niza (max %d): ", DIM);
        scanf("%d", &n);
        if ( n <= 0 || n > DIM ) break;
```

```
        for (i = 0; i < n; i++)
            a[i] = rand()/((double)RAND_MAX+1.0)*10;
```

## Zadatak 12

```
printf("\nPocetni niz:\n\n");
for (i = 0; i < n; i++)
    printf("%d%c", a[i],
           (i % 30 == 29 || i == n-1) ? '\n' : ' ');

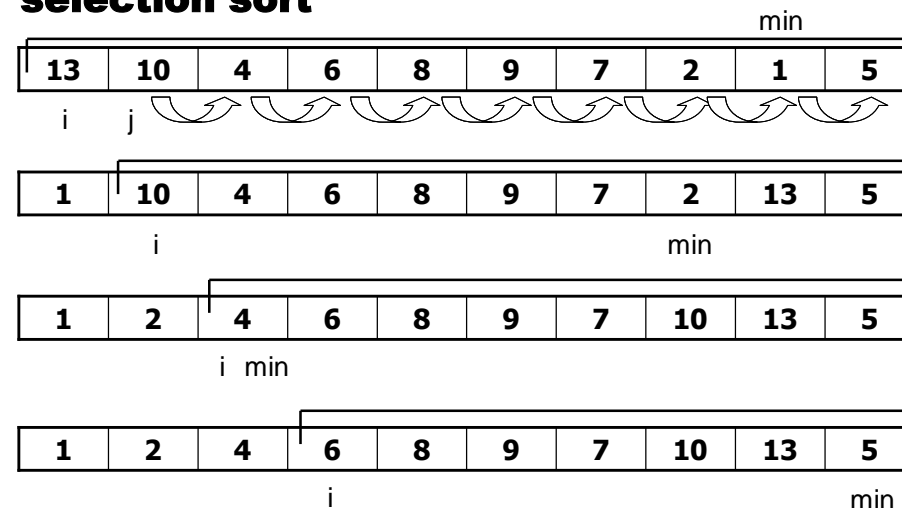
for (i = 0; i < n-1; i++)
    for (j = i+1; j < n; j++)
        if (a[i] > a[j]) b=a[i], a[i]=a[j], a[j]=b;

printf("\nSortirani niz:\n\n");
for (i = 0; i < n; i++)
    printf("%d%c",
           a[i],
           (i % 30 == 29 || i == n-1) ? '\n' : ' ');
}
```

## rand()

- Funkcija koja generiše pseudoslučajan ceo broj u opsegu od 0 do RAND\_MAX
- Nalazi se u standardnom zaglavlju <stdlib.h>
  - RAND\_MAX je konstanta iz standardnog zaglavlja <stdlib.h>
  - #define RAND\_MAX 0x7FFF
- rand() generator relativno loših karakteristika
- Postavljanje semena: srand(int)
- Generisanje opsega vrednosti:  
$$\text{rand()} / ((\text{double})\text{RAND\_MAX}) * (\text{high} - \text{low}) + \text{low}$$
- high i low predstavljaju donju i gornju granicu opsega u kome želimo da generišemo slučajan broj

## Uređivanje metodom izbora – selection sort



## Zadatak 13 – enum (C20 mod)

- Sastaviti program na programskom jeziku C koji sa standardnog ulaza učitava srednje temperature po mesecima za 12 meseci u godini i na standardnom izlazu ispiše redni broj meseca čija se srednja temperatura najmanje razlikuje od godišnjeg proseka.

## Zadatak 13

- Rešenje:

```
#include <stdio.h>
#include <math.h>
#define BROJ_MESECI 12

void main() {
    enum meseci
        {JAN=1,FEB,NAR,APR,MAJ,JUN,JUL,AVG,SEP,OKT,NOV,DEC};
    enum meseci mesec=JAN, najblizi;

    float temperature[BROJ_MESECI];
    float srednja_temp=0, razlika, najmanja_razlika;

    while (1) {
        printf("Temperatura za mesec %2d: ",mesec);
        scanf("%f",&temperature[mesec - 1]);
        srednja_temp+=temperature[mesec - 1];
        if (mesec==DEC) break;
        mesec++;
    }
}
```

## Zadatak 13

```
srednja_temp /= BROJ_MESECI;
najblizi = JAN;
najmanja_razlika = fabs(srednja_temp - temperature[0]);
for( mesec = FEB; mesec <= DEC; mesec++ ) {
    razlika = fabs(srednja_temp - temperature[mesec]);
    if( razlika < najmanja_razlika ) {
        najblizi = mesec;
        najmanja_razlika = razlika;
    }
}

printf("Mesec: %d, razlika %f\n", najblizi,
najmanja_razlika);
}
```

## Zadatak 14 – bitski operatori

- Šta ispisuje sledeći program na programskom jeziku C?

```
#include <stdio.h>
void main() {
    int a=55, b=27;
    if (a & 0xfa)
        if (b | 037)
            a=6,b=4;
        else
            a=7,b=2;
    else
        a=5,b=3;
    printf("%d", a+b);
    printf("%d", a-b);
}
```

## Operatori za rad sa bitovima

### • Bitski operatori

- primenjuju se samo nad celobrojn timer tipovima podataka
- funkcionišu na nivou bita u binarnoj reprezentaciji operanada

Operator	Operandi	Rezultat
(bitsko "ili")	A=0000 1111 <sub>(2)</sub> B=0101 0101 <sub>(2)</sub>	A B = 0101 1111 <sub>(2)</sub>
& (bitsko "i")	A=0000 1111 <sub>(2)</sub> B=0101 0101 <sub>(2)</sub>	A&B = 0000 0101 <sub>(2)</sub>
^ (ekskluzivno "ili")	A=0000 1111 <sub>(2)</sub> B=0101 0101 <sub>(2)</sub>	A^B = 0101 1010 <sub>(2)</sub>
<< (pomeranje bita) >>	A=0000 1111 <sub>(2)</sub> B=2	A<<B = 0011 1100 <sub>(2)</sub> A>>B = 0000 0011 <sub>(2)</sub>
>> (označen broj)	A=1000 1111 <sub>(2)</sub>	A>>B = 1110 0011 <sub>(2)</sub>
>> (neoznačen broj)	B=2	A>>B = 0010 0011 <sub>(2)</sub>
~ (komplementiranje)	A=0000 1111 <sub>(2)</sub>	~A = 1111 0000 <sub>(2)</sub>

## Zadatak 14

```
int a=55, b=27;
if (a & 0xfa)
    if (b | 037) a=6,b=4;
    else a=7,b=2;
else a=5,b=3;
```

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

a: 55<sub>10</sub>: 0011 0111<sub>2</sub>  
b: 27<sub>10</sub>: 0001 1011<sub>2</sub>

a & 0xfa:      0011 0111  
                 1111 1010  
                 0011 0010    različito od 0 ⇒ tačno

b | 037:          0001 1011  
                 0001 1111  
                 0001 1111    različito od 0 ⇒ tačno

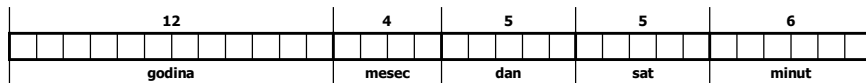
⇒ a = 6, b = 4 ⇒ a+b = 10, a-b = 2 ⇒ ispisuje 102



## Zadatak 15 – bitski operatori (1.5)

- Vreme se zadaje pomoću broja godina, meseca, dana, sata i minuta. Sastaviti na jeziku C program za pakovanje i obrnuti proces raspakivanja podataka o vremenu u jednu 32-bitnu celobrojnu promenljivu. Potrebne podatke čitati sa glavnog ulaza, a rezultate prikazati na glavnom izlazu.

minut	0 – 59	6 bitova
sat	0 – 23	5 bitova
dan	1 - 31	5 bitova
mesec	1 - 12	4 bita
godina		12 bitova
		32 bita



## Zadatak 15

- Rešenje:

```
#include <stdio.h>

void main () {
    unsigned short godina, mesec, dan, sat, minut;
    unsigned long vreme;

    printf ("Dan, mesec, godina? ");
    scanf ("%hd%hd%hd", &dan, &mesec, &godina);
    printf ("Sat, minut? ");
    scanf ("%hd%hd", &sat, &minut);

    vreme = (unsigned long) godina << 20 |
            (unsigned long) mesec << 16 |
            dan << 11 | sat << 6 | minut;
```

## Zadatak 15

```
printf("Pakovano: %lu (%lx)\n", vreme, vreme);
godina = vreme >> 20;
mesec = vreme >> 16 & 0x0f;
dan = vreme >> 11 & 0x1f;
sat = vreme >> 6 & 0x1f;
minut = vreme & 0x3f;

printf ("Raspakovano: %hd.%hd.%hd %hd:%hd\n",
        dan, mesec, godina, sat, minut);
}
```

## Zadatak 16 – min (3.4)

- Sastaviti program na jeziku C za nalaženje vrednosti najmanjeg elementa u nizu realnih brojeva. Program treba da čita i obrađuje nizove sve dok za dužinu niza ne pročita nedozvoljenu vrednost.
- Algoritam za nalaženje ekstremne vrednosti u nekoj kolekciji (nizu, listi...):
  - prvi u kolekciji se proglaši za najmanji
  - obilazi se kolekcija i za svaki element kolekcije ispituje da li ima manju vrednost od trenutno najmanjeg
  - ukoliko postoji manji od trenutno najmanjeg, on se proglašava za najmanji

## Zadatak 16

- **Rešenje:**

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

void main () {
    double a[NMAX], min;
    int n, i;
    while (1) {
        printf ("n? "); scanf ("%d", &n);
        if (n<=0 || n>NMAX) break;
        printf ("A? ");
        for (i=0; i<n;i++) scanf ("%lf", &a[i]);
        min = a[0];
        for (i=1; i<n; i++) if (a[i] < min) min = a[i];
        printf ("min= %.2f\n\n", min);
    }
}
```

## Zadatak 17 – izbacivanje elementa (3.7)

- Sastaviti na jeziku C program za izostavljanje svih elemenata zadatog niza brojeva koji imaju neku zadatak vrednost. Program treba da čita i obrađuje niz sve dok za dužinu niza ne pročita nedozvoljenu vrednost.

## Zadatak 17

- **Rešenje:**

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

void main () {
    int i, j, k, n, a[N];
    while (1) {
        printf ("n? "); scanf ("%d", &n);
        if (n<=0 || n>N) break;
        printf ("A? ");
        for (i=0; i<n;i++) scanf ("%d", &a[i]);
        printf ("k? "); scanf ("%d", &k);

        for (i=j=0; i<n; i++) if (a[i] != k) a[j++] = a[i];
        n = j;

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

## Zadatak 18 – umetanje elemenata (3.8)

- Sastaviti na jeziku C program kojim se u uređeni niz brojeva umeće novi broj tako da niz i dalje bude uređen. Program treba da čita i obrađuje niz sve dok za dužinu niza ne pročita nedozvoljenu vrednost.

## Zadatak 18

- Rešenje:

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

void main () {
    int i, n, b, a[N+1];
    while (1) {
        printf ("n? "); scanf ("%d", &n);
        if (n<=0 || n>N) break;
        printf ("A? "); for (i=0; i<n; scanf ("%d", &a[i++]));
        printf ("b? "); scanf ("%d", &b);
        for (i=n-1; i>=0 && a[i]>b; i--) a[i+1] = a[i];
        a[i+1] = b;
        n++;
        printf ("A= "); for (i=0; i<n; printf ("%d ", a[i++]));
        printf ("\n\n");
    }
}
```

## Zadatak 19 (C35)

- Koja od datih konstrukcija na programskom jeziku C predstavlja ekvivalent iskaza na proramskom jeziku Pascal:

if a > b then begin a:=1; b:=1 end

A) if (a>b) { a=1; b=1; }

B) if (a>b) a=1, b=1;

C) if (a>b) a=1; b=1;

A: tačno, potpuno odgovara segmentu

B: uslovno tačno, krajnji rezultat je isti

C: netačno, b uvek dobija vrednost 1

## Zadatak 20 (C48)

- Data su tri segmenta programa na programskom jeziku C:

I	for (i=x=y=0; ; ++i) {x++; if (i==n) break; y++; } printf("%ld %ld",x,y);
II	x=0, y=0; for (i=0; i<=n; i++) {x=x+1; if (i<n) continue; y=y+1;} printf("%ld %ld",x,y);
III	x=i=0; while (i<=n) { i++; ++x; y=x>n;} printf("%ld %ld",x,y);

- Ako su sve promenljive celi brojevi koja dva segmenta daju isti izlaz za n>0?

A) I i II B) II i III C) I i III

## Zadatak 20

- Rešenje:

I uvek ispisuje x=n+1, y=n, a II i III uvek ispisuju x=n+1, y=1 (n ne utiče na vrednost y).

- Odgovor: B