



Универзитет „Св. Кирил и Методиј“ во Скопје
ФАКУЛТЕТ ЗА ИНФОРМАТИЧКИ НАУКИ И
КОМПЈУТЕРСКО ИНЖЕНЕРСТВО

ФАКУЛТЕТ ЗА ИНФОРМАТИЧКИ НАУКИ И КОМПЈУТЕРСКО ИНЖЕНЕРСТВО

Вовед

дел 1 (вовед)

Структурно програмирање

ФИНКИ 2014

Содржина

- Бројни системи и претставување на броевите во компјутер; Опсег на броеви
- Податочни типови (видови)
- Приказ на податоци на компјутерскиот екран
- Внесување податоци од тастатура
- Операции – прв дел
 - Аритметички операции

Позициони бројни системи

Пример: 9275,34

$$N = a_n a_{n-1} \dots a_3 a_2 a_1 a_0 a_{-1} a_{-2} a_{-3} \dots a_m$$

Каде што a_i , $n \geq i \geq m$ се цифри на бројот

Така, за $n=3$, $m=-2$, $a_3=9$, $a_2=2$, $a_1=7$, $a_0=5$, $a_{-1}=3$ и $a_{-2}=4$

Во декаден броен систем

Општа форма на претставување на бројот е:

$$N = a_n b^n + a_{n-1} b^{n-1} + \dots + a_3 b^3 + a_2 b^2 + a_1 b^1 + a_0 b^0 + a_{-1} b^{-1} + a_{-2} b^{-2} + a_{-3} b^{-3} + \dots + a_{-m} b_{-m}$$

Бројни системи

Декаден - со основа 10 и со цифри: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

Бинарен - со основа 2 и со цифри: 0, 1.

Октален - со основа 8 и со цифри: 0, 1, 2, 3, 4, 5, 6, 7.

Хексадекаден - со основа 16 и со цифри:
0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C,
D, E, F

$$\text{Пример: } 315_{10} = 1000111011_2$$

Бит со најголема важност ↓

↓ бит со најмала важност

100111011

Специјално

■ Бинарен – октален – хексадекаден

бинарен

100 111 011.111 010 100

↕ ↕ ↕ ↕ ↕ ↕

октален

4 7 3 . 7 2 4

бинарен

0001 0011 1011.1110 1010 0000

↕ ↕ ↕ ↕ ↕ ↕

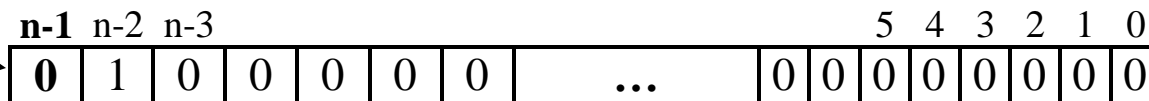
хексадекаден

1 3 В . Е А 0

Претставување на броевите во компјутер, ЦЕЛИ (1)

- Цели броеви
 - без предзнак – неозначени (unsigned) (0 до $max-1$)
 - со предзнак – означени (signed) ($-(max-1)$ до max)
- Кај броеви **без предзнак** битот со најголема важност е **дел од** бројот.
- Кај броеви **со предзнак** битот со најголема важност се користи за претстава на **знакот на** бројот.
 - Тој е: **0** – за позитивните броеви и нулата
 1 – за негативните броеви

бит со најголема важност



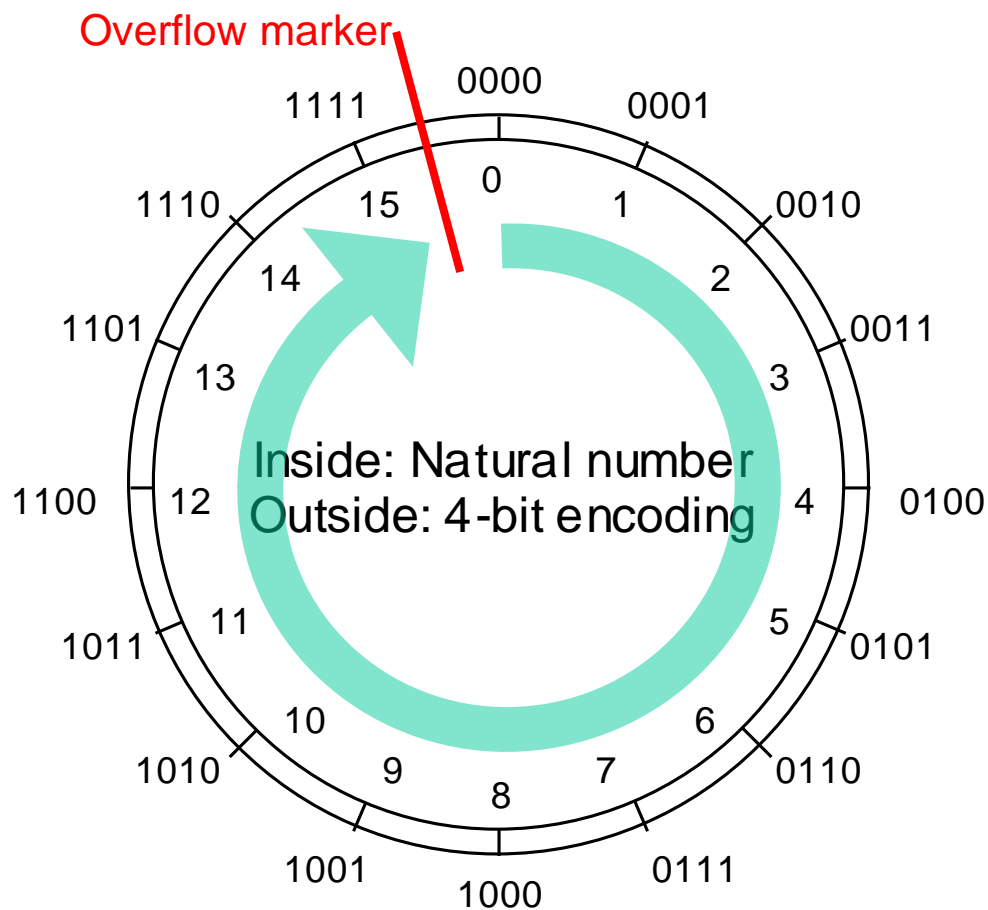
Битот со најголема важност **НЕ Е** директна претстава на предзнакот!
Не важи: $+ \rightarrow 0$; $- \rightarrow 1$

НЕОЗНАЧЕНИ

■ $n=4$, 0 до 15

0000 +1=	1000 +1=
0001 +1=	1001 +1=
0010 +1=	1010 +1=
0011 +1=	1011 +1=
0100 +1=	1100 +1=
0101 +1=	1101 +1=
0110 +1=	1110 +1=
0111 +1=	1111 +1=

0000



Опсег на ОЗНАЧЕНИ броеви

Опсегот на означените броеви претставени со n -битен збор е

$$-2^{n-1} \leq x \leq 2^{n-1}-1$$

има вкупно $2^{n-1} + 1$ (со нулата) + $2^{n-1} - 1 = 2^n$ броеви.

	од	до
$n=5$	$-2^{n-1} = -2^4 = -16$	$2^{n-1}-1 = 2^4-1 = 15$
$n=8$	$-2^{n-1} = -2^7 = -128$	$2^{n-1}-1 = 2^7-1 = 127$
$n=16$	$-2^{n-1} = -2^{15} = -32768$	$2^{n-1}-1 = 2^{15}-1 = 32767$
$n=32$	$-2^{n-1} = -2^{31} = -2147483648$	$2^{n-1}-1 = 2^{31}-1 = 2147483647$

Како се добиваат броевите од опсегот?

- Со n битови, броевите се наоѓаат во опсегот $[-2^{n-1}, 2^{n-1} - 1]$
- Негативниот дел од опсегот се добива со
 - ИНВЕРТИРАЊЕ
 - Додавање 1

... ДВОЕН КОМПЛЕМЕНТ

Пр. $n=3$, $[-4, 3]$

- 0 0 0 0
- 0 0 1 1
- 0 1 0 2
- 0 1 1 3
- 1 0 0 -4 (11 + 1 = 100 (4))
- 1 0 1 -3 (10 + 1 = 11 (3))
- 1 1 0 -2 (01 + 1 = 10 (2))
- 1 1 1 -1 (00 + 1 = 01 (1))

Кај овие **предзначени** броеви
битот со најголема важност е:
0 за позитивен број
1 за негативен број.

Двоен комплемент (1)

$$\neg \begin{array}{c} 0001 \\ \downarrow \downarrow \downarrow \downarrow \\ 1110 + 1 \\ \downarrow \\ 1111 \end{array} \quad \begin{array}{l} = +1_{10} \\ \\ \\ = -1_{10} \end{array}$$

$$\neg \begin{array}{c} 1111 \\ \downarrow \downarrow \downarrow \downarrow \\ 0000 + 1 \\ \downarrow \\ 0001 \end{array} \quad \begin{array}{l} = -1_{10} \\ \\ \\ = +1_{10} \end{array}$$

$$-3 = 101 \rightarrow \neg 010 \xrightarrow{+1} \textcolor{red}{0}\underline{11} = 3 \quad -(-3)=3$$

$$-4 = 100 \rightarrow \neg 011 \xrightarrow{+1} \textcolor{red}{1}\underline{00} = \textcolor{red}{-4} \quad -(-4)=?$$

Двоен комплемент (2)

- Менување на знак на број може да се изведе и на следниов начин

1. Поаѓајќи до лево, пронајди ја првата единица
2. Инвертирај ги сите битови лево од неа.

- Примери:

□ $00101001 = +41$

□ $00101100 = +44$

□ 00101001

□ 00101100

□ $11010111 = -41$

□ $11010100 = -44$

- Собирањето/одземањето на броеви претставени со двоен комплемент не бара никаква дополнителна обработка ако операндите имаат различни предзнаци: знакот на резултатот се добива автоматски

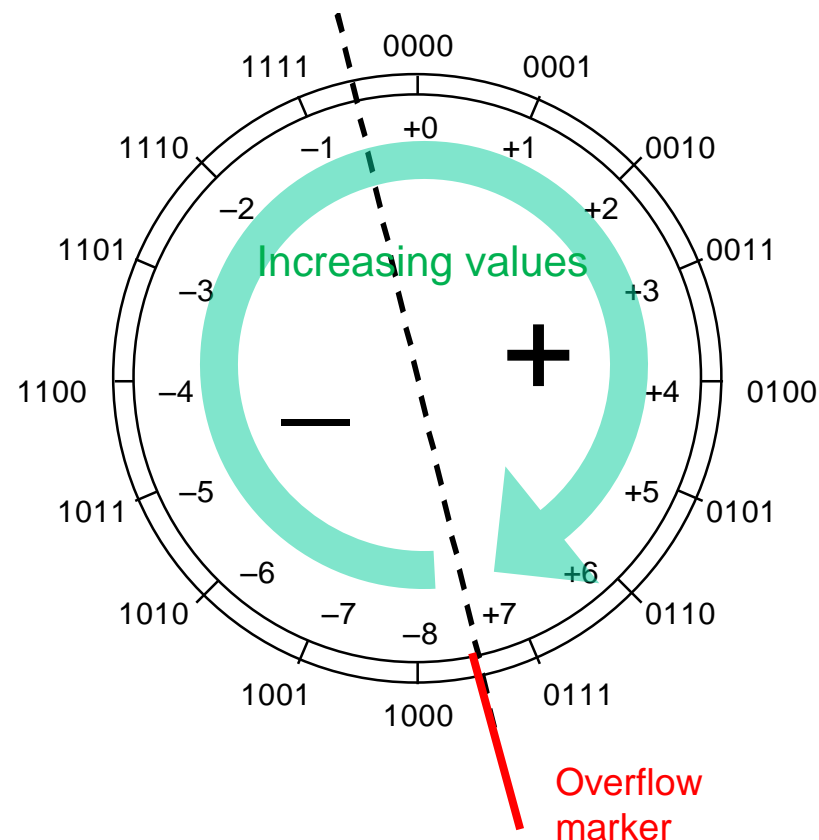
Преполнување (overflow)

За $n=4$, -8 до 7 ($-2^{n-1} \leq x \leq 2^{n-1}-1$)

При аритметички операции со
броеви ограничени во некој опсег
доаѓа до

**Преполнување, претекување,
прелевање (overflow),**

резултатот е НАДВОР од опсегот!!



Уште еднаш, за $n=5$

Означени цели 16-битни броеви во опсегот -16 до 15

декаден број	бинарен број	декаден број	бинарен број
	16 – бита		16 – бита
0	00...00000	-16	11...10000
1	00...00001	-15	11...10001
2	00...00010	-14	11...10010
3	00...00011	-13	11...10011
4	00...00100	-12	11...10100
5	00...00101	-11	11...10101
6	00...00110	-10	11...10110
7	00...00111	-9	11...10111
8	00...01000	-8	11...11000
9	00...01001	-7	11...11001
10	00...01010	-6	11...11010
11	00...01011	-5	11...11011
12	00...01100	-4	11...11100
13	00...01101	-3	11...11101
14	00...01110	-2	11...11110
15	00...01111	-1	11...11111

Како да се претстават информациите во компјутерот?

- Бит(**bit**, binary digit),
 - Вредности 0 или 1.
- Пр. Бројот 10010111 е долг 8 бита (бајт, **byte**).
- Битовите се основна единица на складирана информација во дигиталните информациски системи.

ПРЕТСТАВУВАЊЕ НА ПОДАТОЦИТЕ

- Податоците во компјутерите се претставуваат со битовите **0** и **1**.
- Основна причина
 - Цифрите 0 и 1 во електрониката се претставуваат со две вредности на напон.
 - Со цифрите 0 и 1 може да се претстави секој податок.

Текстуални податоци

- ASCII код (American Standard code for Information Interchange)
Пр. & е 0100110
- BCD (Binary Coded Decimal)
- EBCDIC (Extended BCD Interchange Code)
- Unicode

Контролни знаци

Интерпункциски и цифри

Големи букви и спец. знаци

Мали букви и специјални знаци

7-битен 128 знаци,

8-битен 256 знаци

	000	001	010	011	100	101	110	111
0000	NUL	DLC	SP	0	@	P	`	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	"	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	\$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	u	v
0111	BEL	ETB	'	7	G	W	u	v
1000	BS	CAN	(8	H	X	h	x
1001	HT	EM)	9	I	Y	i	y
1010	LT	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[k	{
1100	FF	FS	,	<	L	/	l	/
1101	CR	GS	-	=	M]	m	}
1110	SO	RS	.	>	N	^	n	
1111	SI	VS	/	?	O	_	o	DEL

Поголеми единици

- 1 byte = 8 bits
- 1 kilobyte (K / KB) = 2^{10} bytes = 1,024 bytes
- 1 megabyte (M / MB) = 2^{20} bytes = 1,048,576 bytes
- 1 gigabyte (G / GB) = 2^{30} bytes = 1,073,741,824 bytes
- 1 terabyte (T / TB) = 2^{40} bytes = 1,099,511,627,776 bytes
- 1 petabyte (P / PB) = 2^{50} bytes = 1,125,899,906,842,624 bytes
- 1 exabyte (E / EB) = 2^{60} bytes = 1,152,921,504,606,846,976 bytes
- (70) zetta, (80) yotta

Содржина

- Бројни системи и претставување на броевите во компјутер; Опсег на броеви
- **Податочни типови (видови)**
- Приказ на податоци на компјутерскиот екран
- Внесување податоци од тастатура
- Операции – прв дел
 - Аритметички операции

Структура на програма во C

```

1.  /* zaglavie – ime na programata, avtor,
    verzija */
2.  /* INCLUDE sekcija – sodrzi #include izrazi */
3.  /* definiranje konstanti i podatocni vidovi –
4.  sodrzi definicii na vidovi i #define */
5.  /* GLOBALNI promenlivi */
6.  /* definicija na funkcii */
7.  int main()
8.  {
9.  /* deklaracii na promenlivi */
10. /* izvrsni izrazi */
11.     return 0;
12. }
```

Податочни типови (видови) (1)

■ Податочниот тип определува

- ☐ како вредностите за даден податок се сместуваат во меморијата,
- ☐ множеството вредности за податокот, и
- ☐ операциите што може да се извршат со или над неговите вредности.

■ Поделба

- ☐ Стандардни и изведени (корисничко дефинирани, композитни)
 - Стандардни: int, char, float, ...
 - Изведени: набројувачки, полиња, записи итн.
- ☐ Едноставни и сложени

■ Операции:

- ☐ Аритметички, логички и релациски

Податочни типови (2)

■ целобројни вредности

- ☐ int, long, short, unsigned, ...
- ☐ Вредности на податочниот вид
- ☐ Ограничување на множеството вредности

■ реални вредности

- ☐ float, double, ...
- ☐ Вредности на податочниот вид
- ☐ Ограничување на множеството вредности
 - Големина на броевите и точност на броевите.

■ знакови вредности

- ☐ елементи на ASCII кодната шема
- ☐ во програмите се користат со 'а', '!'

Константи (1)

- ✓ Означуваат податоци што во текот на извршување на програмата не ја менуваат вредноста
- ✓ За секоја константа во програмскиот јазик C се одредува податочниот вид
- ✓ Нумерички константи

➤ целобројни константи

- ❑ децимални константи - 1, 0, 2, 123
 - дозволени цифри, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
- ❑ октални константи – 012, 076
 - дозволени цифри, 0, 1, 2, 3, 4, 5, 6, 7
- ❑ хексадецимални константи – 0x23, 0x7B3F
 - дозволени цифри, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F
- ❑ константи од видот long 890L.

Константи (2)

- ✓ Нумерички константи

- реални константи

- ❑ децимална и експоненцијална нотација
 - ❑ броевите што содржат (.) или (e) претставуваат константи од видот double
 - ❑ да се дефинира float константа потребно е да се додаде F на крајот од бројот 3.5F, 1e-7F,
 - ❑ ВНИМАНИЕ: 12.0L е long double, но 12L е long int

- ✓ Знакови константи

- 'a', '\n', '%'

- ✓ Текстуални низи

- "ana", "", "\n", "a"

Променливи (1)

- Имињата на променливите соодветствуваат на локација во компјутерската меморија
- Секоја променлива има име (идентификатор), вид, големина и вредност
- Секогаш кога нова вредност ќе биде сместена во контејнерот (променливата), претходната вредност **се брише**
- Читањето вредност од променливата **не ја менува** вредноста

- Визуелен приказ

25

godiniJana

Променливи (2)

✓ Декларирање

- го кажува на компјутерот името и типот на променливата

✓ Формат

```
tipPromenliva ImePromenliva;  
tipPromenliva Ime1, Ime2, Ime3;  
tipPromenliva Ime=Vrednost;  
tipPromenliva Ime1=vrednost1, Ime2=vrednost2;
```

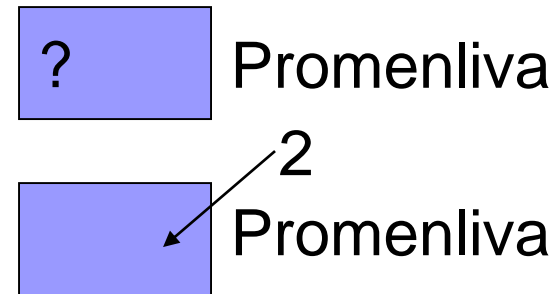
✓ Правила:

- секогаш се прави на почетокот на функција или
- на почеток на кој и да е блок од изрази или
- надвор од функција
- и тоа секогаш пред да биде употребена

Задавање вредност на променлива

```
int Promenliva;
```

```
Promenliva = 2;
```



■ формат

```
Promenliva = vrednost;
```

Податочни типови - пример

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

```
int main() {
```

```
    int suma;
```

```
    float pari;
```

```
    char bukva;
```

```
    double pi;
```

```
    suma = 10;        /* zadadi celobrojna vrednost od vidot int*/
```

```
    pari = 2.21;      /* zadadi realna vrednost od vidot float*/
```

```
    bukva= 'A';       /* zadadi znak */
```

```
    pi = 2.01E6;      /* zadadi realna vrednost od vidot double*/
```

```
    printf("Vrednost na suma = %d\n", suma );
```

```
    printf("Vrednost na pari = %f\n", pari );
```

```
    printf("Vrednost na bukva = %c\n", bukva );
```

```
    printf("Vrednost na pi = %e\n", pi );
```

```
    return 0;
```

```
}
```

Излез:

```
Vrednost na suma = 10
```

```
Vrednost na pari = 2.210000
```

```
Vrednost na bukva = A
```

```
Vrednost na pi = 2.010000e+06
```

Содржина

- Бројни системи и претставување на броевите во компјутер; Опсег на броеви
- Податочни типови (видови)
- **Приказ на податоци на компјутерскиот екран**
- **Внесување податоци од тастатура**
- **Операции – прв дел**
 - Аритметички операции

Приказ на податоци на компјутерскиот екран

■ формат:

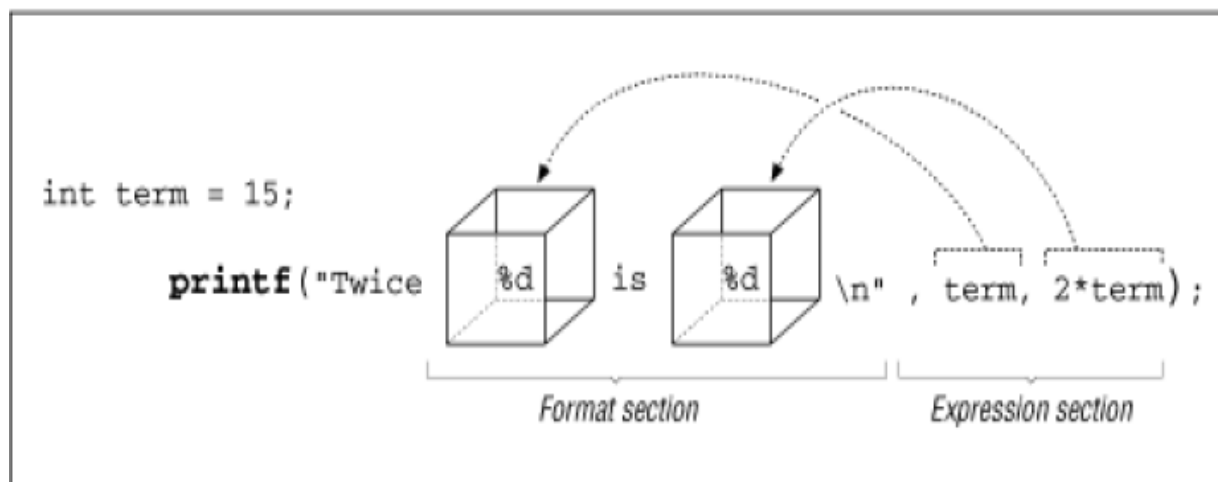
```
printf(format, izraz1, izraz2, ...);
```

- format – текстуална низа
- izraz1, izraz2, ... се индивидуални вредности што треба да се прикажат

пример:

```
#include <stdio.h>

int main() {
    int term;
    term = 3 * 5;
    printf("Twice %d is %d\n", term, 2*term);
    printf("Three times %d is %d\n", term,
        3*term);
    return (0);
}
```



Примери со проблеми!

```
#include <stdio.h>
int main() {
    int sum; /* sum */
    sum = 2 + 2;
    printf("The sum is %d\n");
    return (0);
}
```

```
#include <stdio.h>
int main() {
    float quotient; /* quotient*/
    quotient = 7.0 / 22.0;
    printf("the quotient is %d\n", quotient);
    return (0);
}
```


Внесување на податоци од тастатура

- формат `scanf(format, &prom1, &prom2, ...);`
 - `format` – текстуална низа што ги содржи информациите за редоследот и распоредот на податоците што се внесуваат
 - ознаките за внесување на податоци
 - `%c` податокот е знак
 - `%d` податокот е цел број
 - `%f` податокот е реален број
 - `%h` податокот е цел број од видот `short int`
 - `%o` податокот е цел број во октална нотација
 - `%i` податокот е од видот `unsigned decimal integer`
 - `%x` податокот е цел број во хексадецимална нотација
 - `&` означува дека вредноста на променливите `prom1, prom2, ...` треба да се измени со сместување на вредноста што се внесува од тастатура
 - `prom1, prom2, ...` ги означуваат променливите во кои се внесуваат податоците

Внесување податоци од тастатура

■ пример

```
#include <stdio.h>
int main() {
    int a, b; /*kreira dve celobrojni promenlivi*/
    printf("vnesi dva broja: ");
    scanf("%d%d", &a, &b); /*smestuva celobrojni
                           vrednosti vo promenlivite a i b*/
    printf("%d - %d = %d\n ", a, b, a-b);
    return (0);
}
```

Аритметички операции

■ Аритметички оператори

■ Изрази

■ Правила

- ☐ int op int
- ☐ float op float
- ☐ int op float
- ☐ char op char

С операции	Оператор	Аритметички Израз	Израз во С
собирање	+	$f + 7$	<code>f + 7</code>
одземање	-	$p - c$	<code>p - c</code>
множење	*	bm	<code>b * m</code>
делење	/	x / y	<code>x / y</code>
модул	%	$r \bmod s$	<code>r % s</code>

■ Правила за приоритет

Содржина

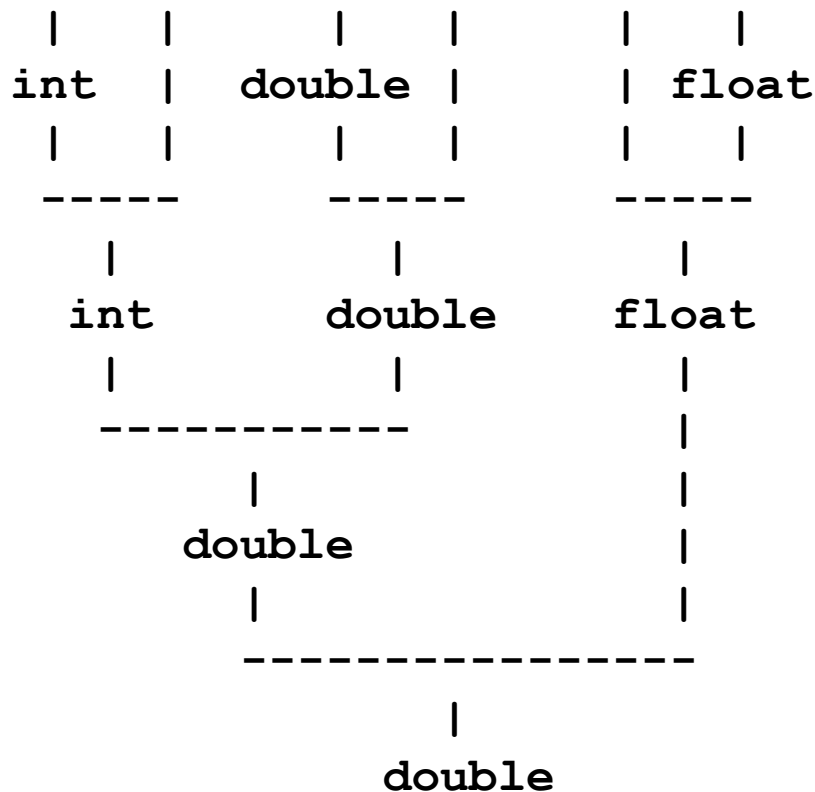
- Бројни системи и претставување на броевите во компјутер; Опсег на броеви
- Податочни типови (видови)
- Приказ на податоци на компјутерскиот екран
- Внесување податоци од тастатура
- **Операции – прв дел**
 - Аритметички операции

Аритметички операции

Оператор(и)	Операции	Приоритет
()	загради	Се пресметува на почеток. Ако заградите се вгнездени, изразот во највнатрешните загради се пресметува прв. Ако постојат повеќе загради на исто ниво во изразот тие се пресметуваат одлево надесно.
*, /, или %	множење, делење, модул	Се пресметува втора. Ако во изразот има повеќе операции од овој вид тие се пресметуваат одлево надесно.
+ или -	собирање одземање	Се пресметува последен. Ако во изразот има повеќе операции од овој вид тие се пресметуваат одлево надесно.

Кој тип на податок се добива на крај?

```
char ch; int i; float f; double d;
rezultat = (ch / i) + (f * d) - (f + i);
```



За носење дома

- Нешто за позиционите бројни системи
 - http://en.wikipedia.org/wiki/Positional_notation
 - http://en.wikipedia.org/wiki/Two%27s_complement
- Како се претставуваат броевите во компјутерот
 - Опсегот (рангот) на броевите во C
 - Зошто се јавуваат грешки при операции над броеви во даден опсег
- Типови (видови) податоци
- Приказ на податоци на компјутерскиот екран
- Внесување податоци од тастатура
- Операции (прв дел)
 - Аритметички операции

Прашања?