Данни и оператори

Прости данни и оператор за разклоняване на пресмятанията (оператор if)

Константи и променливи

```
int a;
double b;
char c;
```

Литерални константи - низ

Низ в оператора cout – загражда се в (двойни) кавички "...."

Пример:

```
cout << "Hello";

cout << "Hello" << endl;

cout << a << " " << b << endl;

cout << "a" << " " << a << endl;
```

Литерални константи - числа

Примери:

```
int a = 5;
double b = 3.14;
doble b1 = 1.23e3;
long long int a = 5;
long long int a = 1234567890123456L;
```

Знаков тип (character)

char c;

Литерал за знак чрез единични кавички:

char c = 'z';

ASCII таблица

32		48	0
33	!	49	1
34	"	50	2
35	#	51	3
36	$	52	4
37	%	53	5
38	&	54	6
39	'	55	7
40	(	56	8
41	)	57	9
42	*	58	:
43	+	59	;
44	,	60	<
45	-	61	=
46	.	62	>
47	/	63	?

64 @	80 P	96 `	112 p
65 A	81 Q	97 a	113 q
66 B	82 R	98 b	114 r
67 C	83 S	99 c	115 s
68 D	84 T	100 d	116 t
69 E	85 U	101 e	117 u
70 F	86 V	102 f	118 v
71 G	87 W	103 g	119 w
72 H	88 X	104 h	120 x
73 I	89 Y	105 i	121 y
74 J	90 Z	106 j	122 z
75 K	91 [	107 k	123 {
76 L	92 \	108 l	124 |
77 M	93]	109 m	125 }
78 N	94 ^	110 n	126 ~
79 O	95 _	111 o	127

Номер на знак

```
char a='A';
 int n = (int)a;
 // int n = a; // същото, но в стария C стил
 cout << n << endl; // отпечатва 65
// Обратно превръщане
 char b;
 b = (char) 65;
 cout << b << endl; // отпечатва знак A
```

Превръщане на цифра в число

```
char c='7';
cout << (int)c << endl; // отпечатва 55
cout << (int)c - (int)'0' << endl; // отпечатва 7
int i = (int)c-(int)'0';
```

Въвеждане на знаци

```
char c1, c2, c3;
cin >> c1 >> c2 >> c3;
cout << c1 << endl;
cout << c2 << endl;
cout << c3 << endl;
```

Входът може да бъде:

```
xyz
```

или

```
x   y   z
```

или

```
x
y
z
```

Комбинирано въвеждане

```
int a,b; char c;
cin >> a >> c >> b;
cout << a << endl;
cout << c << endl;
cout << b << endl;
```

Входът може да бъде:

```
12+34
```

или

```
12   +   34
```

И за двата входа, изходът е:

```
12
+
34
```

Разклоняване на пресмятанията

Условен оператор.

Пример за намиране на по-голямото (максимума) от две числа:

```
int a,b,r;
cin >> a >> b;
if(a>b){r=a;}
else{r=b;}
cout << r << endl;
```

Пример за намиране на максимума и на минимума на две числа:

```
int a,b,min, max;
cin >> a >> b;
if(a>b){max=a;min=b;}
else{max=b;min=a;}
cout << max << " " << min << endl;
```

Общ вид на условен оператор

if (B)

{ P1; P2; ... PN;}

else

{Q1; Q2; ... QM;}

където B е *условие,*

P1; P2; ... PN; Q1; Q2; ... QM; са оператори

{ P1; P2; ... PN;} се нарича съставен оператор или *блок*

Условието в оператора if

Просто условие: две числа, свързани със знак за отношение: ==, <, >, <=, >=, !=

Например:

```
if(a==b) cout << "a е равно на b";
if(a!=b)cout << "a е различно от b";
```

Някои от клаузите в оператора if може да са празни или да се състоят от един оператор. Например:

```
if(a>b){  } else {x=a;}
```

или `if(a>b) {x=a;} else { }`

Във втория случай може да не пишем клаузата else, т.е. да запише само:

```
if(a>b) {x=a;}
```

Когато в блока има само един оператор, не е задължително да записваме скоби {}, т.е. може да запишем `if(a>b)x=a;else y=b;`

Размяна на стойностите на две променливи

Понякога трудно се възприема, че последователността

a=b; b=a;

няма да осъществи размяната

Правилният подход е с въвеждане на допълнителна променлива:

int t=a; a=b; b=t;

Размяна на стойностите на две променливи чрез функция от STL

```
#include <algorithm> // понякога е необходимо

int a=2; int b=3;
swap(a,b);
cout << a << " " << b << endl;


char a='x'; char b='y';
swap(a,b);
cout << a << " " << b << endl;
```