

# Приложения на условия оператор

if (B)

{ P1; P2; ... PN;}

else

{ Q1; Q2; ... QM;}

# Намиране най-малкото от 3 числа

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

# Подреждане по големина на 3 числа

```
int a,b,c;
```

```
cin >> a >> b >> c;
```

```
if(a>b) swap(a,b);
```

```
if(b>c) swap(b,c);
```

```
if(a>b) swap(a,b);
```

```
cout << a << " " << b << " " << c << endl;
```

# Пълна проверка за правилността на програмата за подреждане по големина на 3 числа

Трябва да проверим всичките 6 възможности:

1) 1 2 3

2) 1 3 2

3) 2 1 3

4) 2 3 1

5) 3 1 2

6) 3 2 1

# Задача

Записани са 3 цели положителни числа:  $a$ ,  $b$  и  $c$ , такива че  $a+b=c$ . Някой изтрил едно от числата.

Напишете програма, която въвежда числата, като на мястото на изтритото число въвежда 0.

Програмата трябва да възстанови изтритото число и да изведе трите числа.

# Решение

```
int a, b, c, x, y, z;  
cin >> a >> b >> c;  
if(a==0){x=c-b; y=b; z=c;}  
if(b==0){x=a; y=c-a; z=c;}  
if(c==0){x=a; y=b; z=a+b;}  
cout << x << " " << y << " " << z << endl;
```

# Вложени оператори if

При решаване на по-сложни задачи, някои от оператори в условния оператор също може да бъдат условни оператори. Например при следния фрагмент променливата s получава стойност -1, 0 или 1, в зависимост от това дали x има отрицателна, нулева или положителна стойност:

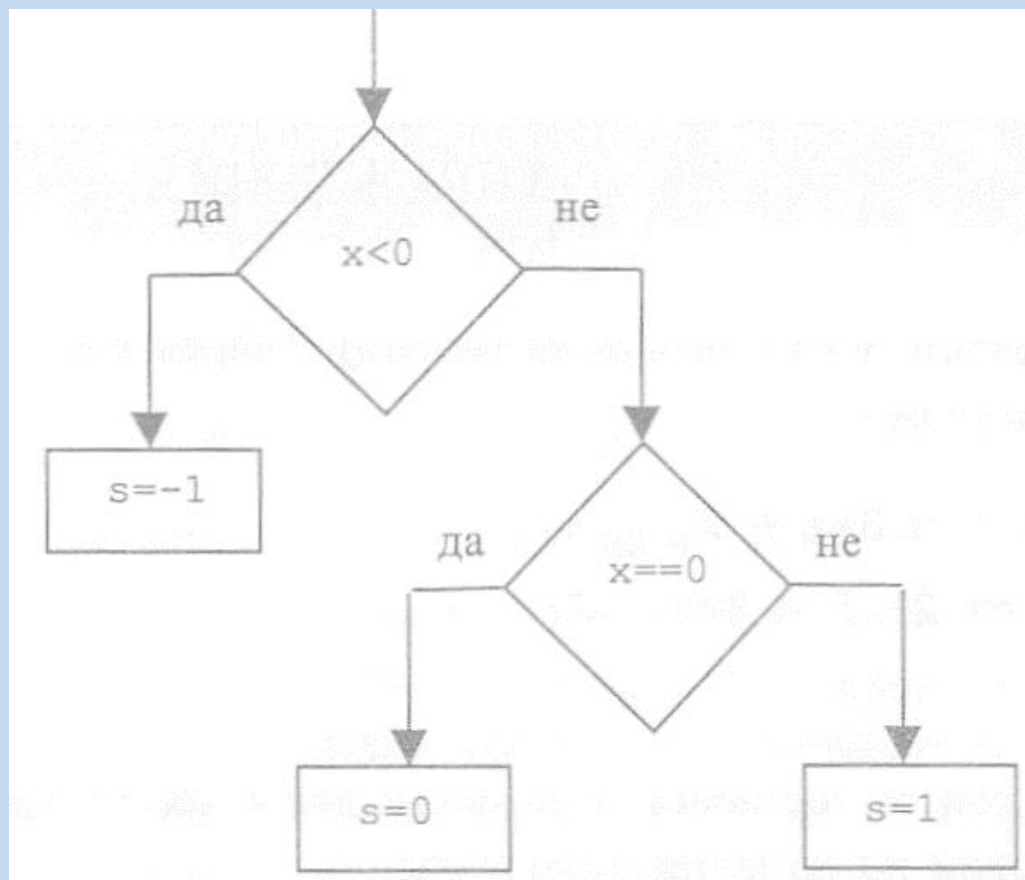
```
if(x<0) s=-1;  
else if(x==0) s=0;  
else s=1;
```

или

```
if(x<0) s=-1;  
else {if(x==0) s=0; else s=1;}
```



# Представяне чрез блок-схема



# Математически запис

$$s = \begin{cases} -1, & \text{ако } x < 0 \\ 0, & \text{ако } x = 0 \\ 1, & \text{ако } x > 0 \end{cases}$$

# Съставни логически условия

Ако  $A$  и  $B$  са условия то:

**( $A$ ) и ( $B$ )** е ново условие, което е вярно тогава и само тогава, когато и двете условия  $A$  и  $B$  са верни. Означава се с  $(A) \& \& (B)$

**( $A$ ) или ( $B$ )** е ново условие, което е вярно тогава и само тогава, когато поне едно от условията  $A$  или  $B$  е вярно. Означава се с  $(A) || (B)$

**не (A)** е ново условие, което е вярно тогава и само тогава, когато A не е вярно. Означава се с  $\neg(A)$

# Примери

Ако за числовите променливи  $a$  и  $b$  е вярно условието  $a < b$ , тогава:

Съставното условие  $(a \leq x) \&\& (x \leq b)$  е вярно точно за тези стойности на  $x$ , които принадлежат на интервала  $[a, b]$

Съставното условие  $(x < a) \vee (b < x)$  е вярно точно за тези стойности на  $x$ , които лежат извън интервала  $[a, b]$

Съставното условие  $\neg(a < x)$  е вярно точно за тези стойности на  $x$ , за които е изпълнено, че  $x \leq a$ .

Задача: за числовите променливи a, b и c, напишете програма, която отпечатва тази от стойностите им, която е най-голяма

```
if( a>=b)&&(a>=c) cout << a;  
else if(b>=a)&&(b>=c) cout << b;  
else if(c>=a)&&(c>=b) cout << c;
```

Задача: Да се напише програма, която проверява, дали правоъгълник с дължини на страните  $a$  и  $b$ , може да се постави вътре в друг правоъгълник с дължини на страните  $c$  и  $d$ , така страните на двата правоъгълника да са съответно успоредни.

За да бъде възможно поставянето трябва да е изпълнено, че числата в поне една от двойките  $(a, b)$  или  $(b, a)$  са съответно по-малки от числата в двойката  $(c, d)$



Това означава, че трябва да е изпълнено поне едно от следните две условия:

$(a < c)$  и  $(b < d)$

или

$(b < c)$  и  $(a < d)$

т.е. логическото условие се записва така:

$((a < c) \&\& (b < d)) \mid \mid ((b < c) \&\& (a < d))$

```
int a,b,c,d;
```

```
cin >> a >> b >> c >> d;
```

```
if(((a<c)&&(b<d)) || ((b<c)&&(a<d)))
```

```
    cout << "Yes"
```

```
else cout << "No"
```

Забележка: Логическият израз

$((a < c) \&\& (b < d)) \mid \mid ((b < c) \&\& (a < d))$

може да се запише и така:

$a < b \&\& b < d \mid \mid b < c \&\& a < d$

поради правилата за приоритет на операциите

$<, \&\&, \mid \mid$

Аналогично,  $(a * b) + (c * d)$  може да се запише

$a * b + c * d$

Препоръчваме да се използват скоби

# Логическите изрази имат стойност

Булева стойност 0 или 1 (false или true)

```
bool x=((a<c)&&(b<d))||((b<c)&&(a<d));
```

```
bool y=a<c&&b<d||b<c&&a<d;
```

```
cout << x << y;
```

```
cout << (a<c&&b<d||b<c&&a<d);
```

```
int z = (int) x;
```

# Задачи за дати

Проверка, дали годината е високосна

```
int g; cin >> g;  
if(((g%4==0)&&(g%100!=0)) || (g%400==0))  
    cout << "Yes";  
else cout << "No";
```

Например високосни са 2000 и 2016 г., но 1900 г. не е.