

## Упражнение 03

### Вградени типове данни, дефиниране на променливи, условни изрази

#### Вградени типове данни

Променливите и константите са блокове памет, които имат някакво име и съдържат някакви данни. Данните се съхраняват, за да могат да се използват в последващи изчисления.

Тип данни	Размер	Граници <sup>1</sup>	
		signed	unsigned
char (символ)	1B	от $-2^7$ до $2^7 - 1$	от 0 до $2^8 - 1$
short (цяло число)	2B	от $-2^{15}$ до $2^{15} - 1$	от 0 до $2^{16} - 1$
int (цяло число)	4B	от $-2^{31}$ до $2^{31} - 1$	от 0 до $2^{32} - 1$
float (реално число)	4B	от $-3.4 \times 10^{38}$ до $3.4 \times 10^{38}$	
double (реално число)	8B	от $-1.7 \times 10^{308}$ до $1.7 \times 10^{308}$	
bool (булева стойност)	1B	true, false	

#### Деклариране на променлива, инициализация и оператор за присвояване

Деклариране на променлива се състои от две части:

- тип на данните, които ще бъдат съхранявани в променливата;
- името, с което тези данни могат да бъдат достъпени.

```
int number;
```

На променливата може да се присвои (даде) стойност в момента на нейното деклариране. Тогава променливата е **инициализирана** с начална стойност.

#### Дефиниране на константи

Дефиницията на една константа се състои от:

- ключовата дума **const**;
- типа на данните, които ще се съхраняват в константата;
- името, с което може да бъде достъпена тази стойност;
- стойността, с която се свързва константата.

```
const double RATE = 1.95583;
```

За разлика от променливите, съдържанието на константите не може да бъде променено.

#### Оператор за присвояване

За да се даде стойност на променливата (или константата) при първоначалната инициализация или да се промени стойността на променливата в хода на програмата се използва операторът **=**. Той се нарича *оператор за присвояване*.

```
int number; // неинициализирана променлива

int number = 123; // инициализирана променлива

// използва се операторът за присвояване,
// за да се промени стойността на променливата
number = 23;
```

<sup>1</sup> Посочените интервали от допустими стойности на вградените типове данни могат да се различават, в зависимост от компилатора...

Операторът за присвояване е бинарен, инфиксен. Изпълнява се от дясно наляво.

```
a = b = c = 0;
```

Оценката на израза `c = 0` е стойността 0, след това тази стойност се присвоява на `b` и накрая и на променливата `a`.

Каква е оценката на следния израз?

```
number = number - 1;

// може да се замени с оператора --, едноаргументен
number --;
```

## Кратък преглед на операциите в C++

Категория	Операции
Аритметични	<code>+, -, *, /, %</code> <code>++, --</code>
Логически	<code>&amp;&amp;,   , !</code>
За сравнение	<code>&gt;, &gt;=, &lt;, &lt;=, !=, ==</code>
Аритметични + присвояване	<code>+=, -=, *=, /=,</code>

**Внимание!** Много често срещана грешка е използването на оператор за присвояване `=` вместо оператор за сравнение `==`.

Операциите, включващи присвояване е дясноасоциативна.

Операциите `++, --` са унарни. Могат да бъдат както префиксни, така и постфиксни.

### Целочислено делене (/) и делене с остатък (%)

Операцията `/` е бинарна, инфиксна и лявоасоциативна. Когато и двата аргумента на деленето са цели числа, делето също е *целочислено*.

Ако поне един от операндите на деленето е реално число, операцията връща реален резултат.

```
float A = 4 / 5;
float B = 4. / 5;
float C = 4 / 5.;
float D = 4. / 5.;

cout << "A( " << A << " ) "; // A( 0 )
cout << "B( " << B << " ) "; // B( 0.8 )
cout << "C( " << C << " ) "; // C( 0.8 )
cout << "D( " << D << " ) "; // D( 0.8 )
```

Операцията `%` може да се прилага **единствено** върху **целочислени** аргументи.

```
cout << "4 % 5 = " << 4 % 5 << endl; // 4 % 5 = 0
```

## Условни изрази, разклонени програми

Изразите, в които се използват аритметични операции са **числови**, защото имат стойност число. Изразите, в които се включват операции за сравнение, както и логически операции имат стойност *истина* или *лъжа*. Наричат се още *булеви изрази*.

Пример: Как можем да запишем условието, което проверява дали дадено цяло число е четно?

```
number % 2 == 0
```

## Оценка на изразите $A \ \&\& \ B$ и $A \ || \ B$

Когато се оценява изразът  $A \ \&\& \ B$ , изразът  $B$  се оценява само, когато  $A$  е истина.

Когато се оценява изразът  $A \ || \ B$ , изразът  $B$  се оценява само, когато  $A$  е лъжа.

## Конструкция `if`

Накратко да се разгледа конструкцията...

Синтаксисът на конструкцията е следният:

```
if( <условие> )
    <оператор1>
[else
    <оператор2>]
```

<условие> е булев израз, който се оценява. Ако стойността му е истина, се изпълнява <оператор<sub>1</sub>>. В противен случай, ако съществува блок `else`, се изпълнява <оператор<sub>2</sub>>. Конструкцията `else` не е задължителна.

Ако трябва да бъде изпълнен повече от един оператор в тялото на `if`-а, операциите трябва да бъдат оградени във фигурни скоби `{ }`.

## Задачи

### Задача 1

Да се напише програма, която намира средно квадратичното на три реални числа.

```
double a, b, c;
double saverage = ( a * a + b * b + c * c ) / 3;
```

### Задача 2

Да се напише програма, която намира разстоянието между две точки по техните координати.

```
double x1, y1, x2, y2;
double distance = sqrt((x1-x2)*(x1-x2) + (y1-y2)*(y1-y2));
```

### Задача 3

Да се напише програма, която намира колко от въведените три цели числа са положителни.

```
int countPositive = 0;
if (a > 0)
{
    ++countPositive;
}
```

### Задача 4

Да се напише програма, която намира най-голямото от три цели числа.

Временна променлива `max`.

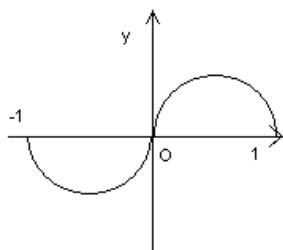
### Задача 5

Да се запише булев израз, който да има стойност истина, ако посоченото условие е вярно и стойност лъжа в противен случай.

- поне едно от числата  $a$ ,  $b$  и  $c$  е положително;
- цялото число се дели на 4 или на 7;
- уравнението  $a \cdot x^2 + b \cdot x + c = 0$ ,  $a \neq 0$  няма реални корени.
- $x$  принадлежи на интервала  $[0; 1]$ ;

- e)  $x$  е извън интервала  $[0; 1]$ ;
- f) цифрата 7 влиза в записа на трицифрено число;
- g) поне две от цифрите на трицифрено число са равни помежду си;
- h) средната цифра на трицифреното число  $a$  е нечетна и е най-голяма или най-малка измежду цифрите на това число, напр. 154 и 212 удовлетворяват условието, докато 234 - не;

### Задача 6



Да се напише програма, която по зададени координати на точка  $(x; y)$ , определя дали тя принадлежи поне на един от полукръговете, включително контура им.

**Уравнение на окръжност:**  $(x - x_0)^2 + (y - y_0)^2 = r^2$

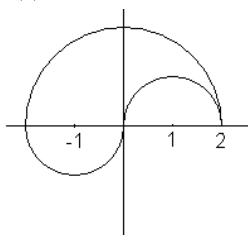
**Уравнение на права:**  $ax + by = c$

Центърът на лявата окръжност е точката  $(-0.5; 0)$ , а радиусът е 0.5. Следователно уравнението на окръжността може да се запише по следния начин:

```
onLeftCircle = ((x + 0.5)*(x + 0.5) + y*y == 0.25);
// без да се включва контура
insideLeftCircle = ((x + 0.5)*(x + 0.5) + y*y < 0.25);
onTheLine = (y == 0);
// без контур, под линията
belowTheLine = (y < 0);

insideLeftSemiCircle = ((x + 0.5)*(x + 0.5) + y*y <= 0.25) && (y <= 0);
```

### Задача 7



Да се напише програма, която по зададени координати на точка  $(x; y)$ , определя дали тя принадлежи във вътрешността и контура на фигурата.