Аритметични операции, коментари, собствени функции, функции за връщане на стойности, използване на аргументи за функции

1. Аритметични операции :

- **+ събиране** //+ **Пр. 24+7 Рез. 31**
- **- изваждане** // - **Пр. 8-5 Рез. 3**
- *** умножение** // * **Пр. 4*4 Рез. 16**
- **/ деление** // / **Пр. 10/2 Рез. 5**
- **% модул /остатък при целочислено деление/** // **Пр. 10%3 Рез. 1**
- **++ увеличава с 1 /оператор за инкрементация/** /* **Пр. B=3; A=++B; -A съдържа 4, B съдържа 4 */**

--намалява с 1 / оператор за декрементация/ /* **Пр. B=3; A=B++; -Асъдържа 3, B съдържа */**

Условният оператор (?) проверява дадено условие и ако то е вярно връща стойност, ако е грешно, връща друга стойност. Структурата му е следната:

условие ? резултат1 : резултат2

Ако условието е вярно, изразът ще върне резултат1, ако не е, ще върне резултат2.

Пример:
7 5 4 3 // **връща 3, защото 7 не е равно на 5**
7==5+2?4:3 //**връща 4, защото 7 е равно на 5+2**

<u>Приоритети на операторите в низходящ ред</u>

1. () скоби
2. +, - унарни //**унарните операции се извършват над един операнд, записани преди операциите , т.е определят знака пред дадено число**
3. *, /, % унарни
4. +, - бинарни // **извършват се над два операнда записани между операндите**

Пример:
(5+6)*33=33 е различно от 5+6*3=23

Операции за сравнение

<	**по-малко**
<=	**по-малко или равно**
>	**по-голямо**
>=	**по-голямо или равно**
==	**равно**
!=	**различно /неравно/**

Като резултат от сравнението се връща логическа стойност 1 или 0 (true, false)

Пример:

```
cout<<(5>4);  истина  Изх. 1
cout<<(-2<-20); лъжа  Изх. 0
cout<<(6==6); истина Изх. 1
cout<<(4!=16); истина Изх.1
cout<<(6<=5); лъжа  Изх. 0
```

Комбинирани операции за присвояване

Операция	Пълна форма	Кратка форма
+-	a=a+20	a+=20
-=	b=b-15	b-=15
*=	c=c*a	c*=a
/=	d=d/b	d/=b
%=	e=e%2	e%=2

Пример 1 : Събиране на две числа

```
#include<iostream>
using namespace std;
int main()
{
  int a, b, c;
  cout<<"Enter two numbers to add";
  cin>>a>>b;
  c = a + b;
  cout<<"Sum of entered numbers = "<<endl;
  cout<<c;
  return 0;
}
```

Резултат:
Enter two numbers to add
3 6
Sum of entered numbers = 9

Пример 2 : Демонстрация на аритметичните операции

```
#include<iostream>
using namespace std;
int main() {

  int a = 21;
  int b = 10;
  int c ;
  c = a + b;
  cout<<"Line 1 - Value of c is %d\n"<<c ;
  c = a - b;
  cout<<"Line 2 - Value of c is %d\n"<<endl;
  c = a * b;
  cout<<"Line 3 - Value of c is "<< c <<endl;
  c = a / b;
  cout<<"Line 4 - Value of c is " <<c <<endl;
  c = a % b;
  cout<<"Line 5 - Value of c is "<<c <<endl;
  c = a++;
  cout<<"Line 6 - Value of c is "<<c <<endl;
  c = a--;
  cout<<"Line 7 - Value of c is "<<c <<endl;

return ();
}
```

Резултат:

Line 1 - Value of c is 31
Line 2 - Value of c is 11
Line 3 - Value of c is 210
Line 4 - Value of c is 2
Line 5 - Value of c is 1
Line 6 - Value of c is 21

Line 7 - Value of c is 22

Пример 3. Да се напише програма на C++, която да изчислява следният израз:

$\sqrt{x} + 3x^3$

```
#include <iostream>       //библиотека за вход- изход
#include<math.h>          //библиотека за математ. функции
using namespace std;
int main ()               //главна функция
{                         //отваряме тялото на функцията
 int x,RES;               //деклариране на променливи
 cout<<"vavedi x=";       //изкарай на монитора
 cin>>x;                  //въведи от клавиатурата
 RES=sqrt(x)+3*(pow(x,3)); //√x +3x^3
 cout<<RES;               //изкарай на монитора резултата
 return 0;                //край на функцията
}                         //затваряме тялото на функцията
```

http://www.cplusplus.com/reference/cmath/pow/
На този адрес може да видите функцията за степен на число POW

Пример 4: Да се състави програма, която въвежда от клавиатурата цяло положително трицифрено число и извежда на екран числото написано в обратен ред

Решение:

```
#include <iostream>
using namespace std;

int main ()
{
   int a,c1,c2,c3;
   cin>>a; //въвежда от клавиатурата трицифрено число a
   c1=a/100; /*цифрата на стотиците получаваме
            като делим трицифреното число на 100, примерно 345/100=3
            цяла част 3 */
```

```
c2=a/10%10; /*цифрата на десетиците получаваме, като делим
последователно 2 пъти на 10: примерно 345 , първи път 345/10=34 и втори път
34%10=3 с остатък 4 - пишем остатъка 4*/
   c3=a%10; /*c3 -цифра на единиците: 345/10=34 с остатък 5 което е
числото, което търсим*/
   cout<<c3<<c2<<c1<<endl; //числото се изписва в обратен ред

    return 0;
}
```

Същата задача може да се реши и по втори начин: цифрите на числото последователно се отделят отдясно наляво и се извеждат на екран:

```
#include <iostream>
using namespace std;

int main ()
{
   int a;
   cin>>a;
   cout<<(a%10); //245/10 остатък 5
   cout<<(a/10%10); //245/10=24, 24%10 е 20 с остатък 4
   cout<<a/100; /*245/100 е цяло число 2 /понеже числата са int
            програмата ще отреже дробната част 45*/
    return 0;
}
```

Пример 5: Да се състави програма, която въвежда от клавиатурата цяло положително четирицифрено число и извежда на екран числото написано в обратен ред

Решение:

```
#include <iostream>
using namespace std;

int main ()
{
   int a;
   cin>>a;
   cout<<(a%10); //2435/10 остатък 5
```

```
    cout<<(a/10%10);  //2435/10=243, 243%10 е 24 с остатък 3
    cout<<a/100%10; /*2435/100 е цяло число 20 с остатък 4
    cout<<a/1000; /*2435/1000 е 2,435 но понеже числото е int, програмата ще
отреже дробната част 435 и ще изведе само 2*/
     return 0;
}
```

Пример 6: Да се състави програма, която въвежда от клавиатурата цяло положителни четирицифрено число. Програмата извежда сумата и произведението на цифрите на числото

Решение:

```
#include <iostream>
using namespace std;
int main ()
{
    unsigned int a,c1,c2,c3,c4;
    cin>>a;
    c1=a/1000;
    c2=a/100%10;
    c3=a/10%10;
    c4=a%10;
    cout<<c1+c2+c3+c4<<"   ";
    cout<<c1*c2*c3*c4;
     return 0;
}
```

Пример 7: Да се състави програма, която въвежда от клавиатурата цяло положително петцифрено число.
Програмата да извежда:
а/ броя на нечетните цифри на числото
б/ сумата на четните цифри на числото

Решение а:

```
#include <iostream>
using namespace std;

int main ()
{
```

```
    unsigned long a; /*числото е long защото едно петцифрено число може да надвиши стойността на int , което е 2 байта, тоест числото което можем да въведем може да бъде примерно над 65535 */
    int c1,c2,c3,c4,c5;
    cin>>a;
    c1=a/10000;  //отделяме всички цифри на числото
    c2=a/1000%10;
    c3=a/100%10;
    c4=a/10%10;
    c5=a%10;
    cout<<c1%2+c2%2+c3%2+c4%2+c5%2<<endl; /*при деление с модул 2 всяко от горните числа  ще даде остатък 1 ако е нечетно а четните ще са с остатък 0 , следователно събираме единиците   и намираме колко са те, тоест тяхната сума, което търсим; ако търсехме нечетнните за cout щяхме да напишем: cout<<5-(c1%2+c2%2+c3%2+c4%2+c5%2); тоест от общия брой на цифрите които са 5 в случая изваждаме нечетните цифри*/
    return 0;
}
```

Решение б:

```
#include <iostream>
using namespace std;

int main ()
{
    unsigned long a;
    int c1,c2,c3,c4,c5,sum;
    cin>>a;
    c1=a/10000;  //отделяме всички цифри на числото
    c2=a/1000%10;
    c3=a/100%10;
    c4=a/10%10;
    c5=a%10;
    sum=c1+c2+c3+c4+c5;
    cout<<sum-(c1%2*c1+c2%2*c2+c3%2*c3+c4%2*c4+c5%2*c5)<<endl;
/*ако имаме числото 12345 sum =15, след което от нея изваждаме сумата на нечетните числа, която се намира като се фиксира чрез деление модул 2 при остатък  съответното нечетно число*/
    return 0;
}
```

2. Добавяне на коментари в програмата

Коментарът е бележка за вас или други програмисти, която поставяте във вашия сорс код.

Всички коментари се игнорират от програмата.

Коментара започва с /* и завършва с */

Коментарите могат да се разпростират на няколко реда.

Коментара може да се постави навякъде, освен по средата на някоя ключова дума, име на функция или променлива.

Може да използвате коментар, за да премахнете някой ред от кода

Въпреки, че още не е дефинирано в стандарта ANSI, , можете да използвате още един стил на коментиране, който се използва в C++ . Това е символа // **- за коментар на един ред. В езика C** този вид коментар е технически невалиден , но повечето компилатори го приемат

Коментарите не могат да се влагат!/коментар в коментар/

Пример 8: Създайте програма, която конвертира земните дни в юпитерски години

1 година на Юпитер /времето за което Юпитер обикаля около Слънцето/ е равна на 12 земни години, като използвате коментари, за да поясните кода на програмата ви.

```
#include<iostream>
using namespace std;
int main() {
   float Earth_days; /*earth days*/
   float Jupiter_years; /*Jupiter years*/

   cout<<"Enter Number of Earth days:";
   cin>>Earth_days; /*enter the number of earth days*/
   Jupiter_years = Earth_days/(365*12.0);  /*make calculation*/
   cout<<"Equivalent Jupiter years:"<<Jupiter_years ;  /*rezultat*/
   return ();
}
```

Резултат:

Enter Number of Earth days:45
Equivalent Jupiter years: 0.010274

Задачи за домашна работа

Зад.1
Напишете програма, която да показва резултатите от няколко целочислени деления и техните остатъци

Зад.2
Напишете програма, която да изчислява лицето на правоъгълник със следните параметри: lenght, width

Зад.3
Напишете програма, която да изчислява броя на секундите в една година

3. Писане на ваши собствени функции

До тук програмите включваха само една функция **main().**
Повечето програми от реалния свят обаче, ще съдържат много повече функции.
Общата форма на C/C++ програма, състояща се от множество функции е:

```
/*включване на хедърни файлове*/
/*прототипи на функции*/

int main(void)
{
/*……*/
}
тип на резултата f1 (списък с параметри)
{
/*……*/
}
тип на резултата f2 (списък с параметри)
{
/*……*/
}
тип на резултата fN (списък с параметри)
{
/*……*/
```

```
}
```
Разбира се имената на функциите могат да бъдат не f1,f2....fN , а други.

<u>Тип на резултата –</u> това са типа на данните, връщани от функцията
Ако дадена функция не връща стойност, тогава типът на нейния резултат трябва да е void
<u>Списък с параметри</u>- ако функцията не използва параметри, нейният списък от параметри трябва да е void
<u>Прототипът</u> на функцията декларира функцията преди тя да се използва и преди нейната дефиниция. Прототипът се състои от името на функцията, типът на резултата и списъкът с параметрите и.
Той завършва със ;
Компилаторът трябва да знае тази информация, за да може да изпълни правилно обръщенията към функцията.

Например : Дадена е простата функция:

```
void myfunc(void)
{
cout<<“This is a test.”;
}
```
Нейният прототип е
```
void myfunc(void);
```
//тоест прототипа е само декларацията на функцията без тялото и, като завършва с ; .

Единствената функция която няма нужда от прототип е main(), тъй като тя е предварително дефинирана от езика C/C++.

Когато една функция се извика, изпълнението на програмата се прехвърля към нея, и когато се достигне края на тази функция, изпълнението се връща веднага след конструкцията за извикване на функцията

Пример 9: Програма, която използва 2 функции func1 и func2
```
#include <iostream>
using namespace std;
void func1(void);
void func2(void);
int main(int argc, char** argv) {
```

```
   func2();   /*главната програма извиква първо тази функция и после
отпечатва  "3"*/
   cout<<"3";
   return ();
}
void func2(void){
   func1();   /*тази функция 2 извиква първо функция1 и после отпечатва "2"*/
   cout<<"2";
   }
void func1(void){
   cout<<"1";  /*това ще се отпечата първо*/
}
```

Задачи за домашна работа

Зад.4 Напишете програма, която съдържа поне две функции и отпечатва съобщението „ I study in PGEA and I live in Sofia”

4. Използване на функции за връщане на стойности

В C/C++ функциите могат да връщат стойност на извикващите конструкции Например стандартната библиотечна функция sqrt() /корен квадратен/ връща квадратен корен от аргумента.

Пример 10: Демонстриране на квадратен корен

```
#include <stdio.h>
#include<iostream>
using namespace std;
int main(){
   double answer;  /*връщаната ст-т е от тип double*/
  answer = sqrt(10.0);   /*присвояване*/
  cout<<answer;
  return 0;
}
```

Тази програма извиква корен квадратен функцията **sqrt() и присвоява връщаната от нея стойност на answer**

Конструкцията за присвояване в горната програма не е технически необходима, защото **sqrt() може просто да използува аргумента в cout**

Пример 11: Намиране на квадратен корен

```
#include<iostream>
using namespace std;
#include <math.h> /* необходим на  sqrt() */
int main(){
  cout<<sqrt(10.0);
  return 0;
}
```

Резултат:
3.162278

Причината за това е, че C /C++ автоматично ще извика sqrt() и ще получи връщаната от нея стойност преди да извика printf()
След това връщаната стойност става втори аргумент на printf()

Когато съставяте ваши собствени функции, връщането на стойност в изискващата функция става чрез конструкцията **return**

return стойност

стойност – стойността, която ще се връща

Пример 12: Тази програма връща стойност 10

```
#include <iostream>
using namespace std;
int func(void); /* прототип */
int main(int argc, char** argv){
   int num;
  num = func( ); // ст-та num присвоява  ст-та на прототипа, която е 10
  cout<<num;
  return 0;
}
int func(void)
{
  return 10; //функцията int func(void) връща стойност 10
```

```
}
```

Пример 13:

```
#include<iostream>
#include <math.h> /* необходима е от sqrt() */
int func(void); /* прототип */
int main(){
   int num;
  num = func( );
  cout<<num;
  return 0;
}
int func(void)
{
  return 10;
}
```

Резултат:
10

Ако дадена функция не задава изрично тип на връщаната стойност, по подразбиране тя се приема за **int .**

```
func(void)
{
  return 10;
}
```

В този случай се подразбира **int**
Когато програмата срещне конструкцията **return,** функцията незабавно приключва. Никакви конструкции след това няма да се изпълнят /функцията приключва преди фигурната скоба/
Стойността на **return** не е задължително да бъде константа.
Конструкцията **return** може да се използва и самостоятелно – без стойност на връщане: **return;..** Тази форма се използва най-често от **void функциите**

Пример 14: Програма, която изписва квадрата на число, въведено от клавиатурата.
Квадратът се изчислява чрез функцията get_sqr()

```
#include<iostream>
#include <math.h>
using namespace std;
int main(int argc, char** argv){
 int sqr;
  sqr = get_sqr( );
  cout<<"Square:"<<sqr;
  return 0;
}
int get_sqr(void)
{
  int num;
  cout<<"Enter a number: ";
  cin>>num;
  return num*num; /* square the number */
}
```

Пример 15: Този пример показва, че след конструкцията return, нищо повече няма да се изпълни

```
#include<iostream>
#include <math.h>
using namespace std;
int main(){
  func1();
  return 0;}
void func1(void)
{
  cout<<"This is printed.";
  return; /* return with no value */
  cout<<"This is never printed.";
}
```

Резултат:

This is printed.

Задачи за домашна работа

Зад.5 Напишете програма, използваща функция на име convert(), изискваща от потребителя да въведе количество долари и след това да

връща стойността им в английски лири / обменния курс да е $2.00 за 1 лира/ Отпечатайте конвертираната стойност.

Зад. 6 Какво не е наред в тази програма

```
#include<iostream>
#include <math.h>
using namespace std;
int f1(void);
int main(){
   double answer;
   answer =f1();
   cout<<answer;
   return 0;
}
int f1(void)
{
   return 100;
}
```

Зад. 7 Какво не е наред в тази функция?

```
void func(void)
{
int=i;
cout<<“Enter a number:”;
cin>>1;
return i;
}
```

5. Използване на аргументи на функции

Аргументът на една функция е стойността, която и се предава при нейното извикване.

Всяка функция в C/C++ може да има от нула до няколко аргумента. /ANSI стандарта дава възможност да са до 31 аргумента/

За да може една функция да приема аргументи, трябва да се декларират специални променливи, които да приемат стойностите на аргументите.

Те се наричат формални параметри на функцията.

Параметрите се декларират между скобите следващи името и.

Пример:

```
void sum(int x, int y);
{
cout<<(x+y);
}
```

При всяко извикване на sum(), тя ще сумира стойността предадена на x, с тази предадена на y

Пример 16:

```
#include<iostream>
#include <math.h>
using namespace std;
void sum(int x, int y);
int main(){
  sum(1, 20); /*аргументи*/
  sum(9, 6);
  sum(81, 9);
  return 0;
}
void sum(int x, int y)  /*извикване на функцията*/
{
  cout<<(x + y);
}
```

Резултат:

21 15 90

Когато се извика функцията sum(), стойността на всеки аргумент 1,20 9,6 81,9 се копират в съответния параметър x и y.

Функция, приемаща параметри, се наричат параметрична функция.

При функциите в C, аргументите винаги се разделят със запетая.

Даден аргумент на функция може да се състои от израз.

Например: `sum(10-2, 9+7);`

Пример 17: Програма използваща функцията outchar(), за да отпечатва знакове на екрана. Програмата отпечатва ABC

```
#include<iostream>
#include <math.h>
using namespace std;
void sum(int x, int y);
int main(){
  sum(1, 20); /*аргументи*/
  sum(9, 6);
  sum(81, 9);
  return 0;
}
void sum(int x, int y)  /*извикване на функцията*/
{
  cout<<(x + y);
}
void outchar(char ch);
int main(int argc, char** argv){
  outchar('A');
  outchar('B');
  outchar('C');
  return 0;
}
void outchar(char ch)
{
  cin>>ch;
}
```

Резултат:
ABC

Зад. 18 Намиране корени на квадратно уравнение

```
#include <iostream>
#include <cmath>
using namespace std;
int main()
```

```
{
   double a,b,c,d,x1,x2,x;
   cout<<"Namira korenite na uravneniq ot tipa: ax^2 +- bx +- c=0\n";
   cout<<"a=";
   cin>>a;
   cout<<"b=";
   cin>>b;
   cout<<"c=";
   cin>>c;
   d=b*b-4*a*c;
   cout<<"D="<<d<<"\n";
   if (d>0)
 {
  x1 = (-b+sqrt(d))/(2*a);
  x2 = (-b-sqrt(d))/(2*a);
 cout << "x1=" << x1 <<"\n";
 cout << "x2=" << x2 <<"\n";
}
  if (d==0)
 {
 x=-b/2*a;
 cout <<"X="<< x <<"\n";
}
   if (d<0)
 {
 cout << "Uravnenieto nqma reshenie"<<"\n";
}
 system( "pause" );
 return 0;
}
```

Зад. 19 Напишете програма, която да изчислява броя на секундите в една година

```
#include <iostream>
using namespace std;
int main() {
   cout<<"Number of seconds in a year is:";
   cout<<60*60*24*365;
   return 0;
```

```
}
```

Резултат:
Number of seconds in a year is:31536000

Зад.20 Напишете програма, която съдържа поне две функции и отпечатва съобщението „ I study in PGEA and I live in Sofia”

```
#include <iostream>
using namespace std;
void one(void);
void two(void);
int main() {
   one();
   two();
      return 0;
}
void one(void){
   cout<<"I study in PGEA and";
   }
void two(void){
   cout<<"I live in Sofia";
}
```

Резултат:
I study in PGEA andI live in Sofia

Задачи за домашна работа

Зад.8 Напишете програма, използваща функция на име outnum(). Нека тази функция да приема един целочислен аргумент и да го изписва на екран .

Зад. 9 Какво не е наред в тази програма?

```
#include<iostream>
#include <math.h>
using namespace std;
void sum(int x, int y);
int main(){
  sum(1, 20); /*аргументи*/
  sum(9, 6);
  sum(81, 9);
  return 0;
}
void sum(int x, int y)  /*извикване на функцията*/
{
  cout<<(x + y);
}
void sqr_it(int num);
int main(){
  sqr_it(10.0);
  return 0;
}
void sqr_it(int num)
{
  cout<<num * num;
}
```

Зад. 10
Гравитацията на Луната е около 17% от тази на Земята. Напишете програма. Която дава възможност на потребителя да въвежда своето тегло и след това да извежда действителното му тегло на Луната.

Зад. 11
Какво не е наред в тази програма?

```
/*въвеждане на число
cin>>num;
```

Зад. 12
Обикновена чаша има обем 8 унции. Напишете програма, която да конвертира унциите в чаши. Използвайте функция, наречена o_to_c() , за

да извършите конвертирането.
Извикайте я с броя на унциите и нека тя да ви връща съответния брой чаши.

Зад. 13 Кои са петте основни типа данни в C/C++ ?

Зад. 14 Какво не е наред с всяко от тези имена на променливи?

a/ short_fall
b/ $balance
c/ last+name
d/ 9times

ТЕСТ

1. **Какъв е резултатът при изпълнението на следната задача: 13%2 ?**
2. **Кой от операторите *, /, (), + , % е с най-висок приоритет?**
3. **Какъв ще е изходът след следното изпълнение: cout<<(4!=16); ?**
4. **Каква е кратката форма на записване на b=b-15 ?**
5. **Да се напише програма, която да изпълни c=√d и да изкара с на изхода**
6. **Да се напише програма, която вкарва четирицифрено число и го изкарва на изхода в обратен ред**
7. **Напишете прототипа на функцията void myfunc(void)**
8. **От какъв тип е връщаната стойност във функцията:**

```
int main(){
   double p;
  p= sqrt(10.0);
  cout<<pr;
  return 0;
}
```

9. **Напишете функция, която приема параметри и се нарича abv**
10. **Кои са петте основни типа данни в езика C/C++**

ТЕСТ – отговори

1. **Какъв е резултатът при изпълнението на следната задача: 13%2 ?**
   1

2. **Кой от операторите *, /, (), + , % е с най-висок приоритет?**

()

3. **Какъв ще е изходът след следното изпълнение: cout<<(4!=16); ?**

1 /истина/

4. **Каква е кратката форма на записване на b=b-160 ?**

b-=160

5. **Да се напише програма, която да изпълни c=√d и да изкара с на изхода**

```
int main ()
{
 int d,c;
 cout<<"vavedi d=";
 cin>>d;
 c=sqrt(d);
 cout<<c;
 return 0;
}
```

6. **Да се напише програма, която вкарва четирицифрено число и го изкарва на изхода в обратен ред**

```
   int a,c1,c2,c3,c4;
   cin>>a;
   c1=a/1000%10;
   c2=a/100%10;
   c3=a/10%10;
   c4=a%10;
   cout<<c4<<c3<<c2<<c1 <<endl;
   return 0;
}
```

7. **Напишете прототипа на функцията void func3(void)**

```
void func3(void);
```

8. **От какъв тип е връщаната стойност във функцията:**

```
int main(){
   double p;
  p= sqrt(10.0);
  cout<<pr;
  return 0;
}
```

double

9. Напишете функция, която приема параметри и се нарича abv

void func abv(int x,int y)

10. Кои са петте основни типа данни в езика C/C++ ?

int, float, double, void, char