

C++ для ЕГЭ

с:

# Обязательные ключевые слова

```
#include <iostream> // 1
using namespace std; // 2
int main() { // 3

    return 0; // 4
}
```

- 1) «#include» - директива(все директивы начинаются с «#»), с ее помощью подключаются библиотеки функций, что позволяет использовать их в коде. После <...> точка с запятой не ставится.
- 2) «using namespace std;» - задает стандартное пространство имен, если не прописать эту строчку, то в будущем необходимо добавлять к некоторым функциям «std::» (например «std::cout << x;», «std::cin >> x;»)
- 3) «int main () {...}» - главная функция программы
- 4) «return 0;» - выполняется последней в теле функции «main ()». Возвращает после выполнения программы в систему 0, что означает правильное выполнение программы.

# Типы данных

```
#include <iostream>
using namespace std;
int main() {
    int a, b = 4, c = 5;
    double d = 0.97;
    bool e = 0;
    char f = '4', g = 't';
    unsigned int k;
    return 0;
}
```

- int - целочисленный тип данных
- double - вещественный тип данных (дроби)
- bool - логический тип данных, может иметь значения 0(нет) и 1(да)
- char - символьный тип данных. Хранит в себе 1 символ. (char f = '4'; переменная f хранит в себе код символа 4, а не само число)
- unsigned int - тот же int, но с удвоенным положительным диапазоном. Отрицательная часть диапазона перекинута вправо от нуля. Переменная этого типа не может быть отрицательной, в ЕГЭ использовать только когда прямым текстом написано «значение переменной может быть огромно»

# Массивы

```
#include <iostream>
using namespace std;
const int N=10; // 1
int main() {
    int a[N] = {1, 4, 3, 7, -1, 5, 11, 404, 44, 120};
    double b[3];
    char c[5] = {"1234"}, d[] = {"Isaac"}; // 2
    return 0;
}
```

- Массив - набор упорядоченных значений одного типа. Имеет вид: «x[N]», где N - количество элементов. Обращение к элементу используется запись «x[3]», но отсчет индексов элементов начинается с 0, то есть эта запись даст доступ к 4-му элементу. Номер последнего элемента для массива «x[N]» - «N-1».
- Для лучшего восприятия массива можно представить книжную полку, пронумерованную слева направо начиная с 0. «x[0]» - первая книга на полке, и если книг 7, то последняя будет иметь номер 6.

# Массивы

```
#include <iostream>

using namespace std;

const int N=10; // 1

int main() {
    int a[N] = {1, 4, 3, 7, -1, 5, 11, 404, 44, 120};
    double b[3];
    char c[5] = {"1234"}, d[] = {"Isaac"}; // 2

    return 0;
}
```

- 1) Часто для количества все элементов используется константная переменная, обычно задают перед функцией «main ()». Важно: нельзя задать размер массива обычной переменной, только либо константой, либо числом
- При объявлении массива можно сразу задать все значения. Если не задавать значения массиву, в его ячейках будет храниться мусор, это важно учесть.
- 2) Для символьного массива свои правила:
  - 1.места выделяется на 1 символ больше под «\0» - символ конца строки.
  - 2.инициализация берется в двойные кавычки (одиночные используются в простой символьной переменной, для 1 символа).
  - 3.мы можем не задавать размер символьного массива(строки), если проинициализировали(задали значение) ее

# Ввод/вывод

```
#include <iostream>      // ПРИМЕР 1
using namespace std;

int main() {
    int a, b;
    cout << "Введите значение переменной a:";
    cin >> a;
    cout << "Значение a = " << a;

    cin >> a >> b;
    cout << a << ' ' << b;

    return 0;
}
```

```
Введите значение переменной a:777
Значение a = 77790
12
90 12Program ended with exit code: 0
```

```
#include <iostream>      // ПРИМЕР 2
using namespace std;

int main() {
    int a, b;
    cout << "Введите значение переменной a:";
    cin >> a;
    cout << "Значение a = " << a << endl;

    cin >> a >> b;
    cout << a << ' ' << b << endl;

    return 0;
}
```

```
Введите значение переменной a:777
Значение a = 777
90
12
90 12
Program ended with exit code: 0
```

- «cout << ...;» - оператор вывода информации на экран, текст можно вывести, заключив его в кавычки. С помощью «<<» можно объединять несколько операторов вывода в один. Нельзя записать:  
«cout << a, b;» и «cout << "Текст", a;»
- «cin >> ...;» - оператор ввода. Задает значение вводимое с клавиатуры в переменную. Можно комбинировать несколько операторов с помощью «>>», тогда после нажатия «Enter» данные будут записываться на следующую переменную.
- В примере 2 используется «endl» в конце некоторых операторов вывода, эта команда переносит строку. В «cin» после каждого ввода значения происходит автоматический переход на следующую строку.
- Для вывода всей строки имеет смысл запись: «cout << a << endl;». «a[]» писать не нужно

# УСЛОВИЯ

- if (<условие>) {...} - оператор условия, если выполняется условия в круглых скобках, то выполняется код в фигурных скобках, иначе игнорирует его.
- else if (<условие>) {...} - проверяется условие в круглых скобках, если условие в предыдущем операторе не выполняется.
- else {...} - выполняется, если в предыдущих операторах условия не верны.
- В комбинации условных операторов, после выполнения кода в {...} программа продолжает читать код после блока условий (отмеченно «1» в коде)

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

int main() {
    int a = 10;
    if (a > 5) {
        cout << "больше 5" << endl;
    } else if (a < 5) {
        cout << "меньше 5" << endl;
    } else {
        cout << "равно 5" << endl;
    }
    // 1
    return 0;
}
```

---

# УСЛОВИЯ

```
#include <iostream>

using namespace std;

int main() {
    int x = 25;
    if ((x%5==0 && x>=10) || x!=0) {
        cout << "Удовлетворяет условиям" << endl;
    }
    return 0;
}
```

- В круглых скобках можно объединять несколько условий. Два амперсанда ( && ) означают «и», две вертикальные палки ( || ) означают «или»
- Запись «x%5==0» сравнивает остаток от деления на 5 с нулем (если остаток равен нулю, то переменная кратна числу).
- В сравнении всегда используются два знака равно, если будет один, программа засчитает это за приравнивание
- «x!=0» - означает x не равен нулю



# ЦИКЛЫ

- for (<переменная счетчика>;<условие>;<действие после выполнения кода в теле>) {...} - цикл, который в самом начале проверяет условие, если оно верно, то выполняет код в теле со стартовым значением переменной счетчика. После выполнения кода в цикле выполняется действие, заключенное перед закрывающей круглой скобкой.
- i++ означает i+1, то есть если i было равно 5, то после i++ его значение будет равно 6
- «x» в итоге будет равно 20

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

int main() {
    int x = 0, i;
    for (i=0; i<20; i++) {
        x++;
    }
    cout << x << endl;
    return 0;
}
```

---

# Циклы + массивы

```
#include <iostream>

using namespace std;

int main() {
    int x[10] = {5, 4, 10, 4, 15, 1, 7, 5, 9, 10};
    for (int i = 0; i<10; i++) {
        if (x[i]%5==0) {
            cout << "Индекс: " << i << " Значение: " << x[i] << endl;
        }
    }
    return 0;
}
```

```
Индекс: 0 Значение: 5
Индекс: 2 Значение: 10
Индекс: 4 Значение: 15
Индекс: 7 Значение: 5
Индекс: 9 Значение: 10
Program ended with exit code: 0
```

```
#include <iostream>

using namespace std;

int main() {
    int x[10];
    for (int i = 0; i<10; i++) {
        x[i] = i;
        cout << x[i] << " ";
    }
    cout << endl;
    return 0;
}
```

```
0 1 2 3 4 5 6 7 8 9
Program ended with exit code: 0
```

- Счетчик цикла можно объявлять в круглых скобках, но для каждого. Но тогда переменная становится локальной и существует только во время выполнения цикла. для каждого нового цикла нужно заново прописывать «int i» в круглых скобках

# C1 (1)

**C 1 № 3649.** Требовалось написать программу, которая решает уравнение  $x^2 + c = 0$  относительно  $x$  для любого числа  $c$ , введенного с клавиатуры. Все числа считаются действительными. Программист торопился и написал программу неправильно.

- Неправильную программу разбирать не буду, лучше напишу рабочую.
- Смысл задачи - найти  $x$ .  $x$   
= корень из  $(-c)$
- для нахождения корня нужно подключить библиотеку «math.h»

```
#include <iostream>
#include <math.h>

using namespace std;

int main() {
    double x, c;
    cout << "Введите c: ";
    cin >> c;
    if (c <= 0) {
        x = sqrt(-c);
    }
    cout << x << endl;
    return 0;
}
```

```
Введите c: -7
2.64575
Program ended with exit code: 0
```

```
Введите c: -9
3
Program ended with exit code: 0
```

Используем тип double,  
для большей точности

sqrt(-c); - корень из -c

# C1 (2)

**C 1 № 3652.** Требовалось написать программу, которая решает неравенство  $(x + a)/(bx) < 0$  относительно  $x$  для любого ненулевого числа  $b$  и любого неотрицательного числа  $a$  и  $(a \geq 0, b \neq 0)$ , введенных с клавиатуры. Все числа считаются действительными. Программист торопился и написал программу неправильно.

- Здесь не смотря на неправильные программы я не понял смысл задания. Требуется просто проверить условия.
- Используем двойное условие
- В таких задачах, если не говорится о целочисленном типе лучше использовать double

ПРОГРАММА НА ПАСКАЛЕ	ПРОГРАММА НА БЕЙСИКЕ	ПРОГРАММА НА СИ
<pre>var a,b,x: real; begin readln(a,b,x); if a = 0 then if b &gt; 0 then write ('нет решений') else write('x &gt; 0 или x &lt;0') else write(-a,'&lt; x &lt;0'); end.</pre>	<pre>INPUT a, b, x   IF a = 0 THEN   IF b &gt; 0 THEN   PRINT "нет решений"   ELSE   PRINT "x&gt;0 или x&lt;0"   ENDIF   ELSE   PRINT -a,"ENDIF   END</pre>	<pre>void main(void) { float a,b,x; scanf("%f%f%f", &amp;a,&amp;b,&amp;x); if (a==0) if (b&gt;0) printf ("нет решений"); else printf ("x&gt;0 или x&lt;0"); else printf("%f}</pre>

Неправильная программа

```
#include <iostream>

using namespace std;

int main() {
    double a, b;
    cout << "Введите a: ";
    cin >> a;
    cout << "Введите b: ";
    cin >> b;
    if (a >= 0 && b != 0) {
        cout << "x > 0 или x < 0" << endl;
    }
    else {
        cout << "нет решений" << endl;
    }
    return 0;
}
```

```
Введите a: 1
Введите b: 5
x > 0 или x < 0
Program ended with exit code: 0
```

```
Введите a: -4
Введите b: 5
нет решений
Program ended with exit code: 0
```

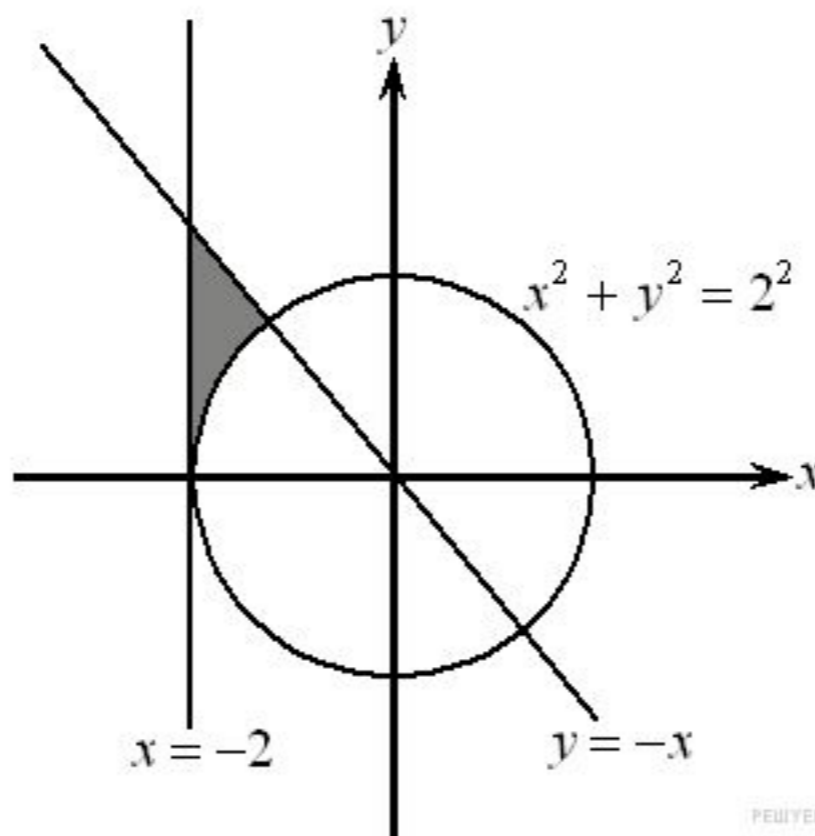
```
Введите a: 1
Введите b: 0
нет решений
Program ended with exit code: 0
```

```
Введите a: -1
Введите b: 0
нет решений
Program ended with exit code: 0
```

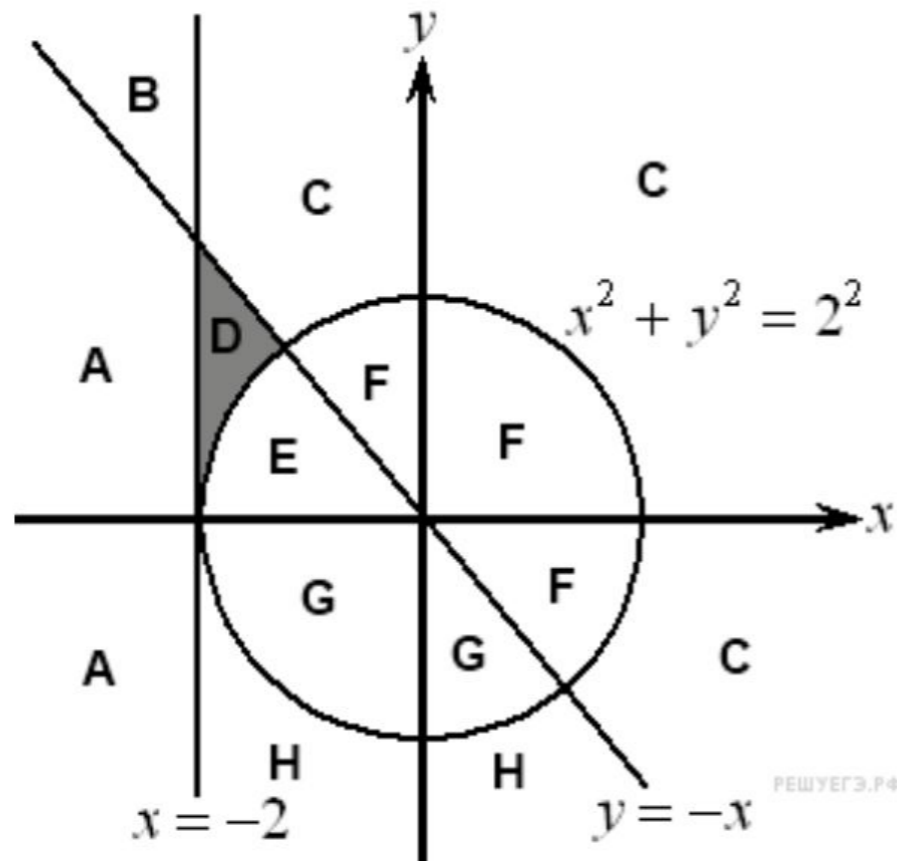
# C1 (3)

**C 1 № 2801.** Требовалось написать программу, которая вводит с клавиатуры координаты точки на плоскости ( $x, y$  — действительные числа) и определяет принадлежность точки заштрихованной области, включая её границы. Программист торопился и написал программу неправильно.

ПРОГРАММА НА ПАСКАЛЕ	ПРОГРАММА НА БЕЙСИКЕ	ПРОГРАММА НА СИ
<pre>var x,y: real; begin readln(x,y); if x*x+y*y&gt;=4 then if x&gt;= -2 then if y&lt;= -x then write('принадлежит') else write('не принадлежит') end.</pre>	<pre>INPUT x, y IF x*x+y*y&gt;=4 THEN IF x&gt;= -2 THEN IF y&lt;= -x THEN PRINT "принадлежит" ELSE PRINT "не принадлежит" ENDIF ENDIF ENDIF END</pre>	<pre>void main(void) {float x,y; scanf("%f %f",&amp;x,&amp;y); if (x*x+y*y&gt;=4) if (x&gt;= -2) if (y&lt;= -x) printf("принадлежит"); else printf("не принадлежит"); }</pre>



В таких задачах нужно определять  
необходимы знак сравнения для  
каждой  $f(x)$



$y = -x$  : необходимо, чтобы  $y$  был  
меньше или равно  $-x$ , тогда будут  
выбраны области A, D, E, G, H

$x^2 + y^2 = 4$  : необходимо, чтобы  
 $x^2 + y^2 \geq 4$ . теперь останутся  
выделенными области A, D, H

$x = -2$  : необходимо, чтобы  $x \geq -2$ ,  
тогда останутся области D, H

по рисунку видно, что нужная  
область лежит выше оси X, значит  
нужно ввести новое условие  $y \geq 0$ ,  
тогда останется область D



```
#include <iostream>

using namespace std;

int main() {
    double x, y;
    cout << "Введите x: ";
    cin >> x;
    cout << "Введите y: ";
    cin >> y;
    if (x*x+y*y >= 4 && x >= -2 && y < -x && y >= 0) {
        cout << "Принадлежит" << endl;
    }
    else {
        cout << "Не принадлежит" << endl;
    }

    return 0;
}
```

```
Введите x: 1
Введите y: 4
Не принадлежит
Program ended with exit code: 0
```

```
Введите x: -2
Введите y: 1
Принадлежит
Program ended with exit code: 0
```

# C2 (1)

**C 2 № 2902.** Дан целочисленный массив из 30 элементов. Элементы массива могут принимать целые значения от 0 до 100 – баллы учащихся выпускного класса за итоговый тест по информатике. Для получения положительной оценки за тест требовалось набрать не менее 20 баллов. Опишите на русском языке или на одном из языков программирования алгоритм, который находит и выводит минимальный балл среди учащихся, получивших за тест положительную оценку. Известно, что в классе хотя бы один учащийся получил за тест положительную оценку. Исходные данные объявлены так, как показано ниже. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать часть из них.

```
const
    N=30;
var
    a: array [1..N] of integer;
    i, j, min: integer;
begin
    for i:=1 to N do readln(a[i]);
    ...
end.
```

По началу кода понимаем, какие введены переменные и используем их в своей программе, обязательно первым действием должен быть цикл, так же, как и в примере.  $i, j$  - чаще всего обозначают счетчик цикла

```
#include <iostream>

using namespace std;

const int N = 30;

int main() {
    int a[N], i, min;
    for (i=0; i<N; i++) {
        cin >> a[i];
    }
    min = a[0];           // 1
    for (i=0; i<N; i++) {
        if (min > a[i]) {
            min = a[i];
        }
    }
    cout << "Минимальный балл: " << min << endl;
    return 0;
}
```

---

В первом цикле забиваем значения массиву. 1) Приравниваем переменную `min` к первому элементу массива, чтобы потом было с чем сравнивать. Во втором цикле сравниваем каждую ячейку массива с `min`, если значение ячейки оказалось меньше, берем число в `min`. Выводим. Не забываем про `const int N = 30;`

В итоге мы не использовали переменную `j`.

# C2 (2)

**C 2 № 2901.** Дан целочисленный массив из 30 элементов. Элементы массива могут принимать значения от 0 до 1000. Опишите на русском языке или на одном из языков программирования алгоритм, который позволяет подсчитать и вывести среднее арифметическое элементов массива, имеющих нечетное значение. Гарантируется, что в исходном массиве хотя бы один элемент имеет нечетное значение.

Исходные данные объявлены так, как показано ниже. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать часть из них.

Бейсик	Паскаль
<pre>N=30 DIM A(N) AS INTEGER DIM I, X, Y AS INTEGER DIM S AS SINGLE FOR I = 1 TO N INPUT A(I) NEXT I ... END</pre>	<pre>const N=30; var a: array [1..N] of integer; i, x, y: integer; s: real; begin for i:=1 to N do readln(a[i]); ... end.</pre>
Си	Алгоритмический
<pre>#include #define N 30 void main(void) { int a[N]; int i, x, y; float s; for (i=0; scanf("%d", &amp;a[i]); ... }</pre>	<pre>алг нач цел N = 30 целтаб a[1:N] цел i, x, y вещ s нц для i от 1 до N ввод a[i] кц ... кон</pre>

```

#include <iostream>

using namespace std;

const int N = 30;

int main() {
    int a[N], i, x=0, y=0;
    double s;
    for (i=0; i<N; i++) {
        cin >> a[i];
    }
    for (i=0; i<N; i++) {
        if (a[i]%2==1) {
            x+=a[i];
            y++;
        }
    }
    s=x/y;
    cout << "Среднее: " << s << endl;
    return 0;
}

```

Так как  
среднее  
число  
может быть  
дробным,  
используем  
double

Переменные:  $i$  - счетчик цикла,  $x$  - сумма нечетных чисел,  $y$  - кол-во нечетных чисел,  $s$  - среднее арифм.

Изначально объявляем  $x$  и  $y = 0$ , чтобы можно было оперировать ими. В 1 цикле задаем массив, во 2-м находим ячейки с нечетными значениями, суммируем значение в переменную  $x$ , добавляем 1 в  $y$ . Запись  $x+=a[i]$ ; тоже самое, что и  $x=x+a[i]$ ;

также существуют и  $x^*=...;$   $x/=...;$   $x-=...;$

Вычисляем среднее арифм. и выводим его.

C4 ПОТОМ C: