**№001 (3 балла)**

#include <iostream>

using namespace std;

void main()

{

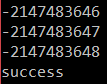
for (int i = 0; i < 20; i--) cout << i << endl;

cout << "success" << endl;

}

Будет ли выведено “success”? Почему?

Да, будет, так как рано или поздно значение i станет меньше минимально возможного значения целочисленного типа, и цикл завершится.



**№002**

#include <iostream>

using namespace std;

void main()

{

int sum = 32;

int ch = 1;

while (true)

{

sum += ch++;

if (!(sum % 5)) continue;

if (!(sum % 7)) break;

}

cout << sum;

}

Какое число будет выведено в консоль?

Так sum изменяется следующим образом: 32 33 35 38 42… – то число 35 будет пропущено, и в консоль будет выведено 42



**№003 (2 балла)**

Дана программа, находящая значения функции в данных точках

#include <iostream>

using namespace std;

void main()

{

for (int i = -5; i <= 5; i++)

{

if (i) continue;

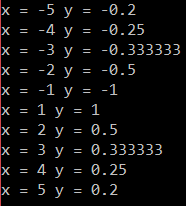
cout << "x = " << i << " y = " << 1.0 / i << endl;

}

}

Какой оператор необходимо добавить для того, чтобы избежать получения бесконечности?

Необходимо добавить оператор отрицания ‘!’, в if (i), чтобы пропустить деление на ноль.



**№004 (5 баллов)**

#include <iostream>

using namespace std;

void main()

{

char ch;

int sum = 0;

for (int i = 0;;i++)

{

cin >> ch;

if (i && (ch \* i == sum)) break;

sum += ch;

}

}

В данную программу ввели символы ‘a’, ‘b’, ‘c’. Какой символ нужно ввести четвёртым, чтобы прекратить работу программы?

Так как, чтобы завершить цикл, необходимо ввести символ, порядковый номер которого в таблице ASCII равен среднему арифметическому уже введённых символов, необходимо ввести символ ‘b’.



**№005 (3 балла)**

#include <iostream>

#include <string>

using namespace std;

void main()

{

string str;

cin >> str;

b: for (int i = 0; i < str.length(); i++)

{

if (str[i] == '0')

{

a: cout << str << endl;

goto a;

}

else

{

goto b;

}

}

}

На вход данной программы подана строка «1230». Что будет выведено в консоль?

Ничего выведено в консоль не будет, так как первый символ – «1», и программа будет бесконечно начинать один и тот же цикл с начала.



**№006 (2 балла)**

#include <iostream>

using namespace std;

int A[5];

void Sum()

{

int sum = 0;

for (int i = 0; i < 5; i++) sum += A[i];

cout << sum << endl;

}

void main()

{

atexit(Sum);

for (int i = 0; i < 5; i++) cin >> A[i];

}

На вход данной программы даны числа 1, 2, 3, 4, 5. Будет ли что-то выведено и если да, то что?

Будет выведена сумма введённых чисел, так как функция atexit() перед выходом из программы вызывает функцию Sum().



**№007 (3 балла)**

#include <iostream>

#include <string>

using namespace std;

void main()

{

cout << "Enter ch, 0 is forbidden, q for quit";

char ch = 0;

string str;

while (true)

{

cin >> ch;

if (ch == '0')

{

cout << "Error: forbidden character\n";

exit();

}

else if (ch == 'q') break;

else

{

str += ch;

}

}

cout << str << endl;

}

Выполнится ли данная программа? Почему?

Программа не выполнится, так как функция exit() имеет обязательный параметр: код ошибки. Если его не указать, компилятор выдаст ошибку.



**№008 (7 баллов)**

void main()

{

int i = 0;

for (;;)

i++;

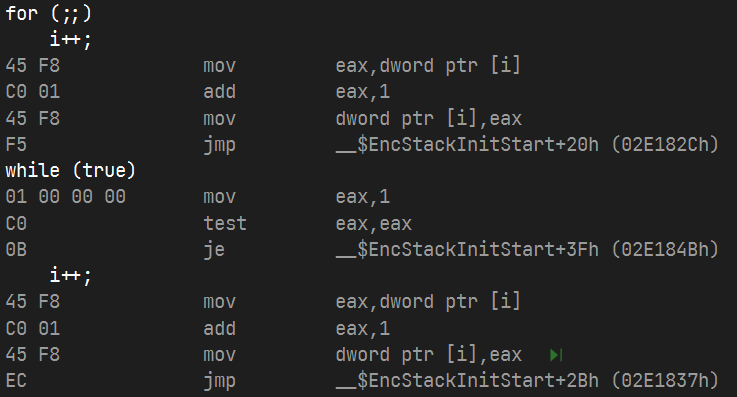
while (true)

i++;

}

Какой из двух операторов цикла является более эффективным для создания бесконечного цикла?

Более эффективным является for(;;), так как в нём пропущена проверка условия, и процессор не выполняет эту ненужную команду.



**№008 (5 баллов)**

#include <float.h>

#include <cmath>

#include <iostream>

void main()

{

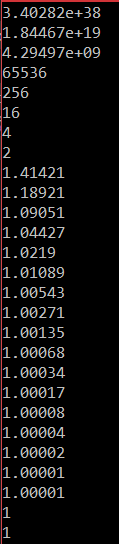
for (float i = FLT\_MAX; i > 0; i = sqrt(i))

std::cout << i << std::endl;

}

Сколько раз выполнится данный цикл?

Данный цикл будет выполняться бесконечно, так как ни из какого числа, большего единицы, нельзя получить число, меньшее единицы, при помощи операции извлечения корня.



**№009 (6 баллов)**

#include <iostream>

void exit()

{

exit(1);

}

void main()

{

atexit(exit);

exit(2);

}

С каким кодом завершит работу данная программа?

Данная программа завершит работу с кодом 1, так как функция exit(1) была вызвана после функции exit(2) и заменила код 2 на 1.



**№010 (5 баллов)**

#include <iostream>

#include <cmath>

using namespace std;

void main()

{

int x = 0;

while (5 \* sqrt(x) - 25)

{

for (int j = 0; j < 5 \* sqrt(x) - 1; j++) cout << " ";

cout << "#" << endl;

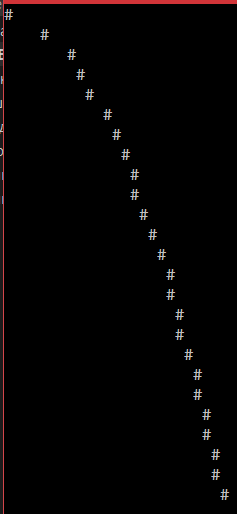
x++;

}

}

Что выводит в консоль данная программа?

Данная программа выводит в консоль график функции 5√x, пока x не станет равен 25.



**№011 (7 баллов)**

#include <iostream>

using namespace std;

void main()

{

for (int i = 0; i < 1, i != 8; i++) cout << i << endl;

}

Что выведет данная программа?

Данная программа выведет числа от 0 до 7, так как использование оператора "запятая" в проверяемом выражении приводит к тому, что левый аргумент будет пропущен, если у него нет побочных эффектов.



**№012 (4 балла)**

#include <iostream>

using namespace std;

void main()

{

for (char i = ' ';; i++) cout << i << endl;

}

Что выведет данная программа?

Данная программа будет бесконечно выводить символы таблицы Windows 1251 по порядку



**№013 (3 балла)**

#include <iostream>

using namespace std;

void main()

{

setlocale(LC\_ALL, "ru");

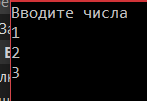
int n = 0;

for (printf("Вводите числа\n"); n - 3;) cin >> n;

}

Выполнится ли данная программа? Почему?

Данная программа выполнится, так как до первой точки с запятой не обязательно должен находиться инициализатор.



**№014 (5 баллов)**

#include <iostream>

using namespace std;

void main()

{

int i = 0;

int mas[1] = {1};

do

{

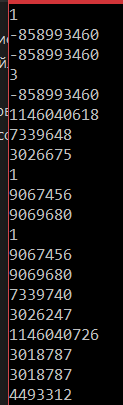
cout << \*(mas + i++) << endl;

} while (\*(mas + i));

}

Сколько раз будет выполнен данный цикл?

Данный цикл будет выполняться, пока по адресу mas + i не найдётся нулевое значение.



**№015 (6 баллов)**

#include <iostream>

using namespace std;

void main()

{

int i = -4;

while ((++i)++)

{

cout << i << endl;

}

}

Что будет выведено в консоль? Почему?

В консоль будут бесконечно выводиться чётные числа, начиная с 2, так как в проверке условия продолжения цикла участвуют значения -3, -1, 1 и так далее, и цикл будет продолжаться бесконечно.

