

Ravesli [Ravesli](#)

- [Уроки по C++](#)
- [OpenGL](#)
- [SFML](#)
- [Qt5](#)
- [RegExр](#)
- [Ассемблер](#)
- [Купить .PDF](#)





Урок №67. Цикл while

 [Юрий](#) |

- [Уроки C++](#)

|

 Обновл. 24 Сен 2020 |

 76302

[↑](#)  58

На этом уроке мы детально рассмотрим цикл while, его конструкцию, особенности и использование.

Оглавление:

1. [Цикл while](#)
2. [Бесконечные циклы](#)
3. [Счетчик цикла while](#)
4. [Итерации](#)
5. [Вложенные циклы while](#)
6. [Тест](#)

Цикл while

Цикл **while** является самым простым из 4-х циклов, которые есть в языке C++. Он очень похож на [ветвление if/else](#):

```
while (условие)  
    тело цикла;
```

Цикл while объявляется с использованием **ключевого слова while**. В начале цикла обрабатывается условие. Если его значением является true (любое ненулевое значение), то тогда выполняется тело цикла.

Однако, в отличие от оператора `if`, после завершения выполнения тела цикла, управление возвращается обратно к `while` и процесс проверки условия повторяется. Если условие опять является `true`, то тогда тело цикла выполняется еще раз.

Например, следующая программа выводит все числа от 0 до 9:

```
1 #include <iostream>
2
3 int main()
4 {
5     int count = 0;
6     while (count < 10)
7     {
8         std::cout << count << " ";
9         ++count;
10    }
11    std::cout << "done!";
12
13    return 0;
14 }
```

Результат выполнения программы:

0 1 2 3 4 5 6 7 8 9 done!

Рассмотрим детально эту программу. Во-первых, инициализируется переменная: `int count = 0;`. Условие `0 < 10` имеет значение `true`, поэтому выполняется тело цикла. В первом `стейтменте` мы выводим `0`, а во втором — выполняем инкремент переменной `count`. Затем управление возвращается к началу цикла `while` для повторной проверки условия. Условие `1 < 10` имеет значение `true`, поэтому тело цикла выполняется еще раз. Тело цикла будет повторно выполняться до тех пор, пока переменная `count` не будет равна `10`, только в том случае, когда результат условия `10 < 10` будет `false`, цикл завершится.

Тело цикла `while` может и вообще не выполняться, например:

```
1 #include <iostream>
2
3 int main()
4 {
5     int count = 15;
6     while (count < 10)
7     {
8         std::cout << count << " ";
9         ++count;
10    }
11    std::cout << "done!";
12
13    return 0;
14 }
```

Условие `15 < 10` сразу принимает значение `false`, и тело цикла пропускается. Единственное, что выведет эта программа:

done!

Бесконечные циклы

С другой стороны, если условие цикла всегда принимает значение `true`, то и сам цикл будет выполняться бесконечно. Это называется **бесконечным циклом**. Например:

```
1 #include <iostream>
2
3 int main()
4 {
5     int count = 0;
6     while (count < 10) // это условие никогда не будет false
7         std::cout << count << " "; // поэтому эта строка будет выполняться постоянно
8
9     return 0; // а эта строка никогда не выполнится
10 }
```

Поскольку переменная `count` не увеличивается на единицу в этой программе, то условие `count < 10` всегда будет `true`. Следовательно, цикл никогда не будет завершен, и программа будет постоянно выводить `0 0 0 0 0....`

Мы можем преднамеренно объявить бесконечный цикл следующим образом:

```
1 while (1) // или while (true)
2 {
3     // Этот цикл будет выполняться бесконечно
4 }
```

Единственный способ выйти из бесконечного цикла — использовать операторы [return](#), [break](#), [goto](#), выбросить исключение или воспользоваться [функцией exit\(\)](#).

Программы, которые работают до тех пор, пока пользователь не решит остановить их, иногда преднамеренно используют бесконечные циклы вместе с операторами `return`, `break` или функцией `exit()` для завершения цикла. Распространена такая практика в серверных веб-приложениях, которые работают непрерывно и постоянно обслуживают веб-запросы.

Счетчик цикла while

Часто нам нужно будет, чтобы цикл выполнялся определенное количество раз. Для этого обычно используется переменная в виде счетчика цикла. **Счетчик цикла** — это целочисленная переменная, которая объявляется с единственной целью: считать, сколько раз выполнялся цикл. В вышеприведенных примерах переменная `count` является счетчиком цикла.

Счетчикам цикла часто дают простые имена, такие как `i`, `j` или `k`. Однако в этих именах есть одна серьезная проблема. Если вы захотите узнать, где в вашей программе используется счетчик цикла и воспользуетесь функцией поиска символов `i`, `j` или `k`, то в результате получите половину своей программы, так как `i`, `j` или `k` используются во многих именах. Следовательно, лучше использовать `iii`, `jjj` или `kkk` в качестве имен для счетчиков. Они более уникальны, их значительно проще найти, и они выделяются в коде. А еще лучше использовать «реальные» имена для переменных, например, `count` или любое другое имя, которое предоставляет контекст использования этой переменной.

Также для счетчиков цикла лучше использовать тип [signed int](#). Использование `unsigned int` может привести к неожиданным результатам. Например:

```
1  #include <iostream>
2
3  int main()
4  {
5      unsigned int count = 10;
6
7      // Считаем от 10 к 0
8      while (count >= 0)
9      {
10         if (count == 0)
11             std::cout << "blastoff!";
12         else
13             std::cout << count << " ";
14         --count;
15     }
16
17     return 0;
18 }
```

Взгляните на эту программу еще раз и постарайтесь найти ошибку.

Оказывается, эта программа представляет собой бесконечный цикл. Она начинается с вывода `10 9 8 7 6 5 4 3 2 1 blastoff!`, как и предполагалось, но затем «сходит с рельсов» и начинает отсчет с `4294967295`. Почему? Потому что условие цикла `count >= 0` никогда не будет ложным! Когда `count = 0`, то и условие `0 >= 0` имеет значение `true`, выводится `blastoff`, а затем выполняется декремент переменной `count`, происходит [переполнение](#) и значением переменной становится `4294967295`. И так как условие `4294967295 >= 0` является истинным, то программа продолжает свое выполнение. А поскольку счетчик цикла является типа `unsigned`, то он никогда не сможет быть отрицательным, а так как он никогда не сможет быть отрицательным, то цикл никогда не завершится.

Правило: Всегда используйте тип `signed int` для счетчиков цикла.

Итерации

Каждое выполнение цикла называется **итерацией** (или «*повтором*»).

Поскольку тело цикла обычно является **блоком**, и поскольку этот блок выполняется по новой с каждым повтором, то любые переменные, объявленные внутри тела цикла, создаются, а затем и уничтожаются по новой. В следующем примере переменная `z` создается и уничтожается 6 раз:

```
1  #include <iostream>
2
3  int main()
4  {
5      int count = 1;
6      int result = 0; // переменная result определена здесь, поскольку она нам понадобится
7
8      while (count <= 6) // итераций будет 6
9      {
10         int z; // z создается здесь по новой с каждой итерацией
11
12         std::cout << "Enter integer #" << count << ':';
13         std::cin >> z;
14
15         result += z;
16
17         // Увеличиваем значение счетчика цикла на единицу
18         ++count;
19     } // z уничтожается здесь по новой с каждой итерацией
20
21     std::cout << "The sum of all numbers entered is: " << result;
22
23     return 0;
24 }
```

Для фундаментальных типов переменных это нормально. Для не фундаментальных типов переменных (таких как **структуры** или классы) это может сказаться на производительности. Следовательно, не фундаментальные типы переменных лучше определять перед циклом.

Обратите внимание, переменная `count` объявлена вне тела цикла. Это важно и необходимо, поскольку нам нужно, чтобы значение переменной сохранялось на протяжении всех итераций (не уничтожалось по новой с каждым повтором цикла).

Иногда нам может понадобиться выполнить что-то при достижении определенного количества итераций, например, вставить символ новой строки. Это легко осуществить, используя оператор остатка от деления со счетчиком цикла:

```
1  #include <iostream>
2
3  int main()
4  {
5      int count = 1;
6      while (count <= 50)
7      {
8          // Выводим числа до 10 (перед каждым числом добавляем 0)
```

```
9      if (count < 10)
10         std::cout << "0" << count << " ";
11     else
12         std::cout << count << " "; // выводим остальные числа
13
14     // Если счетчик цикла делится на 10 без остатка, то тогда вставляем символ нов
15     if (count % 10 == 0)
16         std::cout << "\n";
17
18     // Увеличиваем значение счетчика цикла на единицу
19     ++count;
20 }
21
22 return 0;
23 }
```

Результат выполнения программы:

```
01 02 03 04 05 06 07 08 09 10
11 12 13 14 15 16 17 18 19 20
21 22 23 24 25 26 27 28 29 30
31 32 33 34 35 36 37 38 39 40
41 42 43 44 45 46 47 48 49 50
```

Вложенные циклы while

Также одни циклы while могут быть вложены внутри других циклов while. В следующем примере внутренний и внешний циклы имеют свои собственные счетчики. Однако, обратите внимание, условие внутреннего цикла использует счетчик внешнего цикла!

```
1  #include <iostream>
2
3  int main()
4  {
5      int outer = 1;
6      while (outer <= 5)
7      {
8          int inner = 1;
9          while (inner <= outer)
10             std::cout << inner++ << " ";
11
12         // Вставляем символ новой строки в конце каждого ряда
13         std::cout << "\n";
14         ++outer;
15     }
16
17     return 0;
18 }
```

Результат выполнения программы:

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

Тест

Задание №1

Почему в программе, приведенной выше, переменная `inner` объявлена внутри блока `while`, а не сразу после объявления переменной `outer` (вне блока `while`)?

Ответ №1

Переменная `inner` объявлена внутри блока `while` так, чтобы она была восстановлена (и повторно инициализирована значением 1) каждый раз, когда выполняется внешний цикл. Если бы переменная `inner` была объявлена вне внешнего цикла `while`, то её значение никогда не было бы сброшено до 1, или нам бы пришлось это сделать самостоятельно с помощью операции присваивания. Кроме того, поскольку переменная `inner` используется только внутри внешнего цикла `while`, то имеет смысл объявить её именно там. Помните, что переменные нужно объявлять максимально близко к их первому использованию!

Задание №2

Напишите программу, которая выводит буквы английского алфавита от `a` до `z` вместе с кодами из [ASCII-таблицы](#).

Подсказка: Чтобы выводить символы как целые числа — используйте [оператор `static_cast`](#).

Ответ №2

```
1 #include <iostream>
2
3 int main()
4 {
5     char mychar = 'a';
6     while (mychar <= 'z')
7     {
8         std::cout << mychar << " " << static_cast<int>(mychar) << "\n";
9         ++mychar;
10    }
11
12    return 0;
13 }
```

Задание №3

Измените программу из последнего подраздела «Вложенные циклы» так, чтобы она выводила следующее:

```
5 4 3 2 1
4 3 2 1
3 2 1
2 1
1
```

Ответ №3

```
1 #include <iostream>
2
3 int main()
4 {
5     int outer = 5;
6     while (outer >= 1)
7     {
8         int inner = outer;
9         while (inner >= 1)
10             std::cout << inner-- << " ";
11
12         // Вставляем символ новой строки в конце каждого ряда
13         std::cout << "\n";
14         --outer;
15     }
16
17     return 0;
18 }
```

Задание №4

Теперь сделайте так, чтобы цифры выводились следующим образом (используя программу из предыдущего задания):

```
    1
  2 1
3 2 1
4 3 2 1
5 4 3 2 1
```

Подсказка: Разберитесь сначала, как вывести числа следующим образом:

```
X X X X 1
X X X 2 1
X X 3 2 1
X 4 3 2 1
5 4 3 2 1
```


Ответ №4

```
1  #include <iostream>
2
3  int main()
4  {
5      // Цикл с 1 до 5
6      int outer = 1;
7
8      while (outer <= 5)
9      {
10         // Числа в рядах появляются в порядке убывания, поэтому цикл начинаем с 5 и до
11         int inner = 5;
12
13         while (inner >= 1)
14         {
15             // Первое число в любом ряде совпадает с номером этого ряда,
16             // поэтому числа должны выводиться только если <= номера ряда (в противном
17             if (inner <= outer)
18                 std::cout << inner << " ";
19             else
20                 std::cout << "   "; // вставляем дополнительные пробелы
21
22             --inner;
23         }
24
25         // Этот ряд вывели, переходим к следующему
26         std::cout << "\n";
27
28         ++outer;
29     }
30 }
```

Оценить статью:

 (221 оценок, среднее: 4,88 из 5)[← Урок №66. Оператор goto](#)[Урок №68. Цикл do while →](#)

Комментариев: 58



1. *Юрий:*

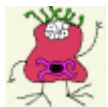
[26 ноября 2020 в 17:34](#)

4 задача

```

1  #include <iostream>
2
3  int main()
4  {
5      int outer = 5;
6      int outer2 = 1;
7      while (outer >= 1)
8      {
9          int inner = outer;
10         while (inner >= 1)
11         {
12             inner--;
13             std::cout << " ";
14         }
15         outer--;
16         int inner2 = outer2;
17         while (inner2 > 0)
18         {
19             std::cout << inner2-- << " ";
20         }
21         std::cout << "\n";
22         ++outer2;
23     }
24     return 0;
25 }
```

[Ответить](#)



2. *Алексей Л.:*

[25 октября 2020 в 23:24](#)

"Следовательно, лучше использовать iiii, jjjj или kkk в качестве имен для счетчиков."
Фу, громоздко, и не красиво.

[Ответить](#)



3. *Иван:*

[23 октября 2020 в 18:46](#)

Мой вариант ромбика, получился не большой:)

```

1  #include <iostream>
```

```
2
3 int main()
4 {
5     int outer = 1;
6     int variable = 2;
7     while (outer <= 5)
8     {
9
10         if (outer <= 3)
11         {
12             int space = outer;
13             while (space <= 2)
14             {
15                 std::cout << " ";
16                 ++space;
17             }
18             int inner1 = 1;
19             while (inner1 <= outer)
20                 std::cout << inner1++ << " ";
21
22             int inner2 = outer - 1;
23             while (inner2 >= 1)
24                 std::cout << inner2-- << " ";
25
26         }
27         else
28         {
29             int space = outer;
30             while (space >= 4)
31             {
32                 std::cout << " ";
33                 --space;
34             }
35             int inner1 = 1;
36             while (inner1 <= variable)
37                 std::cout << inner1++ << " ";
38
39             int inner2 = 1;
40             while (inner2 < variable)
41                 std::cout << inner2++ << " ";
42             --variable;
43         }
44
45         std::cout << "\n";
46         ++outer;
47     }
48
49     return 0;
```

51 | }

[Ответить](#)4. *Иван:*[23 октября 2020 в 17:42](#)

Задание №2 измененный мой вариант без кодов.

```
1  #include <iostream>
2
3  int main()
4  {
5      char symbol1(65);
6      char symbol2(97);
7      int variable(1);
8
9      while (symbol1 <= 90)
10     {
11
12         std::cout << symbol1 << " ";
13         while (symbol2 <= 122)
14         {
15             std::cout << symbol2 << "; ";
16             break;
17         }
18
19         if (variable % 4 == 0)
20         {
21             std::cout << "\n";
22             if (variable == 24)
23                 std::cout << " ";
24         }
25
26         ++symbol1;
27         ++symbol2;
28         ++variable;
29     }
30
31     std::cout << "\n";
32
33     return 0;
34 }
35 }
```

Задание №4 мой вариант.

```
1  #include <iostream>
2
```

```
3 int main()
4 {
5     int outer = 1;
6
7     while (outer <= 5)
8     {
9         int space = outer;
10        while (space <= 4)
11        {
12            std::cout << " ";
13            ++space;
14        }
15        int inner = 1;
16        while (inner <= outer)
17            std::cout << inner++ << " ";
18
19        std::cout << "\n";
20        ++outer;
21    }
22    return 0;
23 }
```

[Ответить](#)



5. *Ruslan:*

[16 сентября 2020 в 16:22](#)

Ромб)

```
1 #include <iostream>
2
3 int main ()
4 {
5     std::cout << "Enter number of rhombus\n";
6     int var_count {};
7     std::cin >> var_count;
8     std::cout << '\a' << "\n\n\n";
9
10    // верхняя часть
11    int outer {1};
12    while (outer <= var_count)
13    {
14        // левый верхний треугольник
15        int inner {var_count};
16        while (inner >= 1)
17        {
18            if (inner <= outer)
19                std::cout << inner << ' ';
```

```
20         else if (inner < 10)
21             std::cout << " ";
22         else
23             std::cout << "   ";
24         --inner;
25     }
26     // правый верхний треугольник
27     inner = 1;
28     while (++inner <= outer)
29     {
30         std::cout << inner << ' ';
31     }
32
33     std::cout << '\n';
34     ++outer;
35 }
36
37 // нижняя часть
38 outer = (var_count - 1);
39 while (outer > 0)
40 {
41     // левый нижний треугольник
42     int inner {var_count};
43     while (inner > 0)
44     {
45         if (inner <= outer)
46             std::cout << inner << ' ';
47         else if (inner < 10)
48             std::cout << " ";
49         else
50             std::cout << "   ";
51
52         --inner;
53     }
54     // правый нижний треугольник
55     inner = 1;
56     while (++inner <= outer)
57     {
58         std::cout << inner << ' ';
59     }
60
61     std::cout << '\n';
62     --outer;
63 }
64
65 std::cout << "\n\n\n";
66
67 return 0;
68
```

}

[Ответить](#)6. *Ruslan:*[16 сентября 2020 в 12:55](#)

Задание № 4

```
1  #include <iostream>
2
3  int main ()
4  {
5      int outer {1};
6      while (outer <= 5)
7      {
8          int inner {5};
9          while (inner >= 1)
10         {
11             if (inner <= outer)
12                 std::cout << inner << ' ';
13             else
14                 std::cout << " ";
15             --inner;
16         }
17
18         std::cout << '\n';
19         ++outer;
20     }
21
22     return 0;
23 }
```

[Ответить](#)7. *Ruslan:*[15 сентября 2020 в 16:40](#)

Задание № 3

```
1  #include <iostream>
2
3  int main ()
4  {
5      int outer {5};
6      while (outer > 0)
7      {
8          int inner {outer};
```

```
9         while (inner > 0)
10             std::cout << inner-- << ' ';
11
12         std::cout << '\n';
13         --outer;
14     }
15
16     return 0;
17
18 }
```

[Ответить](#)8. *Ruslan:*[14 сентября 2020 в 10:35](#)

Задание № 2

```
1 #include <iostream>
2 int main ()
3 {
4     std::cout << "Code: " << "Symbol: " << '\n';
5
6     char count {97};
7     while (count <= 122)
8     {
9         std::cout << static_cast<int>(count) << '\t' << count << '\n';
10        ++count;
11    }
12
13
14    return 0;
15 }
```

[Ответить](#)9. *Вася:*[11 августа 2020 в 14:09](#)

Вот более простое решение четвертого задания:

```
1 #include <iostream>
2
3 int main() {
4     int rows = 1;
5     while (rows <= 5) {
6         int roomForSpace = 5 - rows;
7         int roomForNum = 5 - roomForSpace;
```



```

8      int counter = roomForNum;
9      while (roomForSpace > 0) {
10         std::cout << " ";
11         roomForSpace--;
12     }
13     while (roomForNum > 0) {
14         std::cout << roomForNum;
15         roomForNum--;
16     }
17     std::cout << "\n";
18     rows++;
19 }
20 }

```

[Ответить](#)

1. foo:

[8 сентября 2020 в 19:41](#)

```

1  #include <iostream>
2
3  int main() {
4      std::cout <<
5          "          1\n"
6          "          2 1\n"
7          "          3 2 1\n"
8          "          4 3 2 1\n"
9          "5 4 3 2 1\n";
10     return 0;
11 }

```

😊 шутка

[Ответить](#)

10. Владислав:

[26 июля 2020 в 23:18](#)

Ромб-челлендж.

Идеально ровный, почти красивый ромб.

Максимальное число (размер ромба) по краям, единица посередине.

```

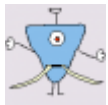
1  #include <iostream>
2  // размерность ромба от 0 до 99
3
4  // вводим число - размер ромба
5  int enterNum()
6  {

```

```
7     using std::cout;
8     using std::cin;
9     int num;
10
11     while (1) {
12         cout << "Enter num --> ";
13         cin >> num;
14
15         // проверка
16         if (cin.fail())
17         {
18             cin.clear();
19             cin.ignore(32767, '\n');
20         }
21         else if (num < 0)
22             cin.ignore(32767, '\n');
23         else if (num > 99)
24             cin.ignore(32767, '\n');
25         else
26         {
27             cin.ignore(32767, '\n');
28             return num;
29         }
30     }
31 }
32
33 // правильно отображаем элемент
34 void correctView(int correct, int cur)
35 {
36     using std::cout;
37
38     if (correct == 2 && cur >= 10)
39         cout << cur << ' ';
40     else if (correct == 2 && cur <= 9)
41         cout << ' ' << cur << ' ';
42     else if (correct == 1)
43         cout << cur << ' ';
44 }
45
46 int main()
47 {
48     using namespace std;
49
50     int num = enterNum();
51
52     // состояние для корректного отображения
53     // 1, если число >= 0 и <= 9
54     // 2, если число >= 10 и <= 99
55 }
```

```
56 int correct;
57 if (num >= 0 && num <= 10)
58     correct = 1;
59 if (num >= 10 && num <= 99)
60     correct = 2;
61 // тройной пробел, если состояние 2
62 // двойной пробел, если состояние 1
63
64 // верхний треугольник
65 for (int i(0); i < num; ++i)
66 {
67     // пробелы перед набором
68     for (int space(i + 1); space < num; ++space)
69         if (correct == 2)
70             cout << "  ";
71         else
72             cout << " ";
73
74     // набор от num до num - i после пробелов
75     for (int curBefore(num); curBefore >= num - i; --curBefore)
76         correctView(correct, curBefore);
77
78     // набор от num - i + 1 до num после первого набора
79     for (int curAfter(num - i + 1); curAfter <= num; ++curAfter)
80         correctView(correct, curAfter);
81
82     cout << '\n';
83 }
84
85 // нижний треугольник
86 for (int i(num - 1); i >= 0; --i)
87 {
88     // пробелы перед набором
89     for (int space(i); space < num; ++space)
90         if (correct == 2)
91             cout << "  ";
92         else
93             cout << " ";
94
95     // набор от num до num - i + 1 после пробелов
96     for (int curBefore(num); curBefore >= num - i + 1; --curBefore)
97         correctView(correct, curBefore);
98
99     // набор от num - i + 2 до num после первого набора
100    for (int curAfter(num - i + 2); curAfter <= num; ++curAfter)
101        correctView(correct, curAfter);
102
103    cout << '\n';
104
```

```
105     }  
106  
    return 0;  
}
```

[Ответить](#)11. *Даня:*[25 июля 2020 в 13:58](#)

Вот мой вариант 4 задания без использования if/else

```
1  #include <iostream>  
2  
3  int main() {  
4      int outer (1);  
5      int counter (5);  
6  
7      while (outer <= 5) {  
8          int iner = outer;  
9  
10         while (counter > 1) {  
11             std::cout << " ";  
12             counter--;  
13         }  
14  
15         while (iner >= 1) {  
16  
17             std::cout << iner-- << " ";  
18         }  
19  
20         std::cout << '\n';  
21  
22         counter = 5 - outer;  
23         outer++;  
24     }  
25  
26     return 0;  
27 }  
28 }
```

[Ответить](#)12. *Onium:*[24 июня 2020 в 18:29](#)

Третье задание

```
1 #include <iostream>
2
3 int main()
4 {
5     int outer = 1;
6
7     while (outer <= 5)
8     {
9
10        int inner = 6 - outer;
11        while (inner >= 1)
12            std::cout << inner-- << " ";
13
14
15        std::cout << '\n';
16
17        ++outer;
18
19
20    }
21
22    return 0;
23 }
```

Четвертое задание

```
1 #include <iostream>
2
3 int main()
4 {
5     int outer = 1;
6
7     while (outer <= 5)
8     {
9
10        int number = 4;
11        while (number >= outer)
12        {
13            std::cout << " " << " ";
14            number--;
15        }
16
17        int inner = outer;
18        while (inner >= 1)
19            std::cout << inner-- << " ";
20
21
22        std::cout << '\n';
23 }
```

```

24         ++outer;
25
26
27     }
28
29     return 0;
30 }

```

Ответить



13. Максим:

[28 мая 2020 в 22:22](#)

Ромб-челендж.

Ровный красивый ромб с единицами по краям.

Подходит для любого числа. Красота ограничивается только размерами экрана. У меня, например, наложения начинаются только после 40.

```

1  #include <iostream>
2  #include <cmath>    // для abs()
3
4  // определяем разряд для числа (нужен для определения кол-ва пробелов)
5  int gradesNumber (int value)
6  {
7      int grade = 1;
8      while(value / 10 > 0)
9      {
10         ++grade;
11         value /= 10;
12     }
13     return grade;
14 }
15
16 int main()
17 {
18     std::cout << "Enter an integer: ";
19     int value;
20     std::cin >> value;
21
22     // задаем разряд для максимального числа (нужен для определения кол-ва пробелов)
23     int maxGrade = gradesNumber(value);
24
25     int line = - value + 1;
26
27     // цикл для строк
28     while (abs(line) < value)    // abs() - модуль числа
29     {
30         int column = - value + 1;

```

```

31 //цикл для столбцов
32 while(abs(column) < value)
33 {
34     if(value - abs(line) - abs(column) <= 0) // условие для "пустых" мест
35     {
36         // прописываем пробелы на "пустых" местах
37         int grade = 0;
38         while(grade <= maxGrade)
39         {
40             std::cout << " ";
41             ++grade;
42         }
43     }
44
45     else
46     {
47         // прописываем число
48         std::cout << value - abs(line) - abs(column);
49
50         // и пробел/пробелы после числа
51         int grade = gradesNumber(value - abs(line) - abs(column));
52         while(grade <= maxGrade)
53         {
54             std::cout << " ";
55             ++grade;
56         }
57     }
58     ++column;
59 }
60
61 std::cout << "\n"; // перенос в конце строки
62 ++line;
63 }
64 return 0;
65 }

```

[Ответить](#)



14. Юшка:

[16 мая 2020 в 11:02](#)

задание №2

```

1 #include <iostream>
2
3 const int A = 97;
4 const int Z = 122;
5

```

```
6 void printLiter();
7
8 int main()
9 {
10     printLiter();
11
12     return 0;
13 }
14
15 void printLiter()
16 {
17     int a = A;
18     while(a <= Z)
19     {
20         std::cout << static_cast<char>(a) << ' ' ;
21         a++;
22     }
23 }
```

[ОТВЕТИТЬ](#)15. *Orest:*[18 марта 2020 в 14:05](#)

Мое решения ромба)

```
1 #include <iostream>
2 #include <string>
3 using namespace std;
4
5 int main()
6 {
7     cout << "Enter the interval: ";
8     int width;
9     cin >> width;
10
11     int step = 1;
12     int reverseStep = width;
13     int iteration = 1;
14
15     while (iteration <= (width * 2 - 1))
16     {
17         int inner = width;
18         while (inner > (step % (width + 1)))
19         {
20             cout << " ";
21             --inner;
22         }
```



```

23     int tmp_inner = inner + 1;
24     while (tmp_inner != -inner)
25     {
26         if (inner == 1 || inner == 0)
27         {
28             --inner;
29             continue;
30         }
31         printf("%2i ", abs(inner--));
32         //cout << abs(inner--) << " ";
33     }
34     if (step < reverseStep)
35         ++step;
36     else
37         step = --reverseStep;
38
39     ++iteration;
40     cout << "\n";
41 }
42 return 0;
43 }

```

[Ответить](#)



16.  Вадим:

[27 февраля 2020 в 13:09](#)

Мой вариант задания №4

```

1  #include <iostream>
2  #include <string>
3  using namespace std;
4
5  /*using status_t = int;
6  status_t editData();
7
8  enum class Enemy {
9      ORGE,
10     GOBLIN,
11     SKELETON,
12     ORC,
13     TROLL
14 };
15
16 struct The_monster {
17     Enemy enemy_type;
18     string name;
19     int health;

```

```
20 };
21
22 string Get_Monster(The_monster enemy) {
23
24     if (enemy.enemy_type == Enemy::ORGE) return "Orge";
25     if (enemy.enemy_type == Enemy::GOBLIN) return "Goblin";
26     if (enemy.enemy_type == Enemy::SKELETON) return "Skeleton";
27     if (enemy.enemy_type == Enemy::ORC) return "Orc";
28     if (enemy.enemy_type == Enemy::TROLL) return "Troll";
29 }
30
31 void Print_Monster(The_monster monster) {
32     cout << "This " << Get_Monster(monster) << " is named " << monster.name << "
33 }
34
35
36 struct Advertising {
37     int total_customers;
38     double persentsge_link;
39     double average_profit;
40 };
41
42
43 double Print_data(Advertising data) {
44
45     double profit = static_cast<double>(data.total_customers) * data.persentsge_
46     return profit;
47 }
48
49 struct Fraction {
50     int division;
51     int divider;
52 };
53
54 void multiply(Fraction f1, Fraction f2) {
55     Fraction new_fraction;
56     new_fraction.division = f1.division * f2.division;
57     new_fraction.divider = f1.divider * f2.divider;
58     cout << new_fraction.division << " / " << new_fraction.divider;
59 }
60
61
62 enum Sign {
63
64     adition,
65     subtraction,
66     multiplication,
67     division,
68 }
```

```
69     taking_remainder
70 };
71
72 int calculatte(int a, int b, Sign math) {
73     switch (math) {
74         case adition:
75             return a + b;
76             break;
77         case subtraction:
78             return a - b;
79             break;
80         case multiplication:
81             return a * b;
82             break;
83         case division:
84             return a / b;
85             break;
86         case taking_remainder:
87             return a % b;
88             break;
89         default:
90             cout << "Error";
91             break;
92     }
93 }
94
95
96 int calculatte(int a, int b, char math) {
97     switch (math) {
98         case '+':
99             return a + b;
100            break;
101         case '-':
102             return a - b;
103            break;
104         case '*':
105             return a * b;
106            break;
107         case '/':
108             return a / b;
109            break;
110         case '%':
111             return a % b;
112            break;
113         default:
114             cout << "Error";
115             break;
116     }
117 }
```

```
118     }
119 }
120
121
122 enum class Animal {
123     pig,
124     chicken,
125     goal,
126     cat,
127     dog,
128     ostrich
129 };
130
131 string getAnimalName(Animal animal) {
132     switch(animal) {
133         case Animal::pig:
134             return "Pig";
135             break;
136         case Animal::chicken:
137             return "Chicken";
138             break;
139         case Animal::goal:
140             return "Goal";
141             break;
142         case Animal::cat:
143             return "Cat";
144             break;
145         case Animal::dog:
146             return "Dog";
147             break;
148         case Animal::ostrich:
149             return "Ostrich";
150             break;
151         default:
152             cout << "Error";
153             break;
154     }
155 }
156
157
158 int printNumberOfLegs(Animal animal) {
159     switch (animal) {
160         case Animal::pig:
161             return 4;
162             break;
163         case Animal::chicken:
164             return 2;
165     }
166 }
```

```
167         break;
168     case Animal::goal:
169         return 4;
170         break;
171     case Animal::cat:
172         return 4;
173         break;
174     case Animal::dog:
175         return 4;
176         break;
177     case Animal::ostrich:
178         return 2;
179         break;
180     default:
181         cout << "Error";
182         break;
183     }
184 }
185
186 }
187
188 void PrintMessage() {
189
190     cout << "A " << getAnimalName(Animal::cat) << " has " << printNumberOfLegs(A
191     cout << "A " << getAnimalName(Animal::chicken) << " has " << printNumberOfLeg
192 }*/
193
194 int main()
195 {
196     int outer = 1;
197
198     while (outer <= 5) {
199         int inner = 1;
200         int xx = 4 - outer;
201         while (xx >= 0) {
202             cout << " " << " ";
203             xx--;
204         }
205         int i = outer;
206         while (inner <= outer) {
207
208             inner++;
209             cout << i-- << " ";
210         }
211         i--;
212         cout << endl;
213         outer++;
214     }
215 }
```

[Ответить](#)17. *ttshkall:*[11 января 2020 в 16:06](#)

вот мое решение ромба:

```
1  #include <iostream>
2  using namespace std;
3  int main()
4  {
5      int a = 0;
6      cin >> a;
7      cout << "\n\n\n";
8      int x = 1;
9      while (x <= a)
10     {
11         int y = a;
12         while (y >= 1)
13         {
14             if (y <= x)
15                 cout << y << " ";
16             else if (y >= 10)
17                 cout << " ";
18             else
19                 cout << " ";
20             y--;
21         }
22         {
23
24             y = 2;
25             while (y <= a)
26             {
27                 if (y <= x)
28                     cout << y << " ";
29                 y++;
30
31             }
32         }
33         cout << "\n";
34         x++;
35     }
36     x = a - 1;
37     while (x >= 1)
38     {
39         int y = a;
40         while (y >= 1)
41         {
```

```
42         if (y <= x)
43             cout << y << " ";
44         else if (y >= 10)
45             cout << " ";
46         else
47             cout << " ";
48
49         y--;
50     }
51     {
52         int y = 2;
53         while (y <= a)
54         {
55             if (y <= x)
56                 cout << y << " ";
57             else if (y >= 10)
58                 cout << " ";
59             else
60                 cout << " ";
61             y++;
62         }
63     }
64     cout << "\n";
65     x--;
66 }
67 }
68 }
```

[Ответить](#)



18. *Марат:*

[27 декабря 2019 в 14:12](#)

Вписываюсь в ромб челлендж

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      short outNum(10);
8      short countStrCol = outNum * 2 - 1;
9      short strNum(1);
10     //short colNum(1);
11     short i(1);
12
13     while (strNum <= countStrCol)
```

```
14 {
15     short colNum(1);
16     short j(0);
17     while (colNum <= countStrCol)
18     {
19         short A = outNum * 2 - i - j;
20         if (colNum < outNum)
21         {
22             if (A < outNum)
23             {
24                 cout << A << " ";
25                 ++j;
26             }
27             else
28             {
29                 cout << " ";
30                 ++j;
31             }
32         }
33         else
34         {
35             if (A < outNum)
36             {
37                 cout << A << " ";
38                 --j;
39             }
40             else
41             {
42                 cout << " ";
43                 --j;
44             }
45         }
46         ++colNum;
47     }
48     if (strNum < outNum)
49         ++i;
50     else
51         --i;
52     ++strNum;
53     cout << "\n";
54 }
55 return 0;
56 }
```

[Ответить](#)



19. *Илья:*

[25 декабря 2019 в 08:30](#)

Хотите пострадать фигней попробуйте этот код)))

```

1  #include <iostream>
2
3  int main()
4  {
5      long x = 0;
6      while (x<2147483647)
7      {
8          std::cout << x << " ";
9          ++x;
10     }
11
12 }
```

[Ответить](#)



20. **Владимир:**

[12 декабря 2019 в 21:00](#)

№4:

```

1  // ОБРАТНЫЙ ОТСЧЕТ
2
3  #include<iostream>
4  using std::cout;
5  /*-----
6                                     //ФУНКЦИЯ ВВОДА ЧИСЛА, КОНТРОЛЯ ЕГО ДИАПАЗОНА И ВОЗВРА
7
8  int inputNumber()
9  {
10     int t;
11     cout << "\n\tInput integer \t";
12     std::cin >> t;
13     if (t > 99)
14         t = 99; // если вводимое пользователем число более 99 - обрезаем его до
15     return t;
16 }
17
18 /*-----
19                                     // ФУНКЦИЯ ОПРЕДЕЛЕНИЯ ОБЩЕГО КОЛИЧЕСТВА ЕДИНИЧНЫХ СИМВ
20
21 int totalSumbolInLine(int t)
22 {
23     int e, d;
24     if (9 >= t)
25     {
26         e = t;
```

```

27     d = 0;
28 }
29 else
30 {
31     e = 9;
32     d = t - 9;
33 }
34 int numTotalSumbol = e + 2 * d ;
35 return numTotalSumbol;    //общее количество знаков в строке
36 }
37 /*-----
38
39     //ФУНКЦИЯ ОПРЕДЕЛЕНИЯ КОЛИЧЕСТВА ЕДИНИЧНЫХ ЧИСЛОВЫХ ЗНАКОВ В
40
41 int maxLine(int i)
42 {
43     int max = 1 + i; //максимальное число в текущей строке
44     ++i;
45     int e1, d1;
46     if (9 >= max)
47     {
48         e1 = max;
49         d1 = 0;
50     }
51     else
52     {
53         e1 = 9;
54         d1 = max - 9;
55     }
56     int maxSumbol = e1 + 2 * d1 ;
57     return maxSumbol;
58 }
59
60 /*-----
61
62 int main()
63 {
64     int t = inputNumber();    // вводимое пользователем
65
66     int numTotalSumbol = totalSumbolInLine(t);    // общее количество
67
68     int numLines=t;    // счетчик цикла по
69
70     int ii=1;    // максимальное чи
71
72     int i = 0;
73
74     while ( numLines > 0)    //цикл печати всей матрицы
75

```

```

76     {
77
78         int maxSumbol=maxLine(i);           //количество единиц
79         ++i;
80
81         int numSpacesInLine = numTotalSumbol-maxSumbol;           //количество единиц
82
83
84         while ( numSpacesInLine > 0 )           //цикл печати пробелов в текущей строке
85         {
86             cout << " ";
87             --numSpacesInLine;
88         }
89
90         int iii = ii;
91         while ( iii > 0 )           //цикл печати цифр в текущей строке
92         {
93             cout << iii;
94             --iii;
95         }
96
97         cout << "\n";           // перенос строки
98
99         --numLines;
100        ++ii;
101    }
102    return 0;
}

```

Ответить

21. Владимир:

[10 декабря 2019 в 15:55](#)

Задание 3 (консоль интереснее развернуть в полный экран):

```

1  #include<iostream>
2  using std::cout;
3
4  int main()
5  {
6      int t;
7      cout << "\n\tInput integer less 80 \t";
8      std::cin >> t;
9      while (t > 0)
10     {
11         int tt = t;
12         cout << "\t";

```

```

13     while (tt > 0)
14         cout << tt--<<" ";
15     cout << std::endl;
16     t--;
17 }
18 return 0;
19 }

```

[Ответить](#)



22. *Владимир:*

[10 декабря 2019 в 15:00](#)

Задание 2:

```

1  #include<iostream>
2  using std::cout;
3
4  void printSumbolCode(char a)
5  {
6      cout << "\n\tSumbol  " << a << "  has code  " << static_cast<int>(a);
7  }
8
9  int main()
10 {
11     char n = 65;
12     while (n <= 90)
13     {
14         printSumbolCode(n);
15         n++;
16     }
17     cout << "\n";
18     return 0;
19 }

```

[Ответить](#)



23. *Анастасия Лузинсан:*

[1 ноября 2019 в 21:32](#)

Пришлось потратить 4 с лишним часов, чтобы более-менее разобраться, как сделать 4 задание с возможностью ввода двузначных чисел. Зато получилась звёздочка 😊
(не советую вводить числа, большие 38, ибо так все цифры сбиваются)

Результат:

```

1  //          1          /
2  //          2 1 2      /

```

```

3 //      3 2 1 2 3      /
4 //      4 3 2 1 2 3 4  /
5 //     5 4 3 2 1 2 3 4 5 /
6 //      4 3 2 1 2 3 4  /
7 //      3 2 1 2 3      /
8 //      2 1 2          /
9 //      1              /

```

Код:

```

1  #include <iostream>
2
3  void print(int a)
4  {
5      int outer = 1;
6      while (outer <= a) //кол-во строчек в ромбе: от 1 до введенное числа -> первая
7      {
8          int inner = a; //крайнее значение звезды(но до этого выведем пропуски)
9          int count = 1; //позиция в строке
10
11         while (count <= a)
12         {
13             if (inner > outer && inner >= 10)
14                 std::cout << " "; //обычные пропуски для чисел >= 10
15             else if (inner > outer) //пропуски для чисел от 1 до 9
16                 std::cout << " ";
17             else
18                 std::cout << inner << " "; //числа внутри ромба
19
20             inner--;
21             count++;
22         }
23
24         inner = 2;
25         while (count <= a*2) //тоже самое, только в обратном порядке
26         {
27             if (inner <= outer)
28                 std::cout << inner << " ";
29             else
30                 std::cout << " ";
31             inner++;
32             count++;
33         }
34
35         std::cout << "\n";
36         outer++;
37     }
38

```

```
39
40 outer = (a-1);//заполняем вторую нижнюю половину звезды
41 while (outer >= 1) //всё тоже самое, но наоборот
42 {
43     int inner = a;
44     int count = 1;
45     while (count <= a)
46     {
47         if (inner > outer && inner >= 10)
48             std::cout << "  ";
49         else if (inner > outer)
50             std::cout << " ";
51         else
52             std::cout << inner << " ";
53
54         inner--;
55         count++;
56     }
57
58     inner = 2;
59     while (count <= a*2)
60     {
61         if (inner <= outer)
62         {
63             std::cout << inner++ << " ";
64         }
65         else
66             std::cout << " ";
67         count++;
68
69     }
70
71     std::cout << "\n";
72     outer--;
73
74 }
75 }
76
77
78
79 int main()
80 {
81     std::cout << "Please, enter a number for piramide: ";
82     int a;
83     std::cin >> a;
84     print(a);
85 }
```

[Ответить](#)24. *Кекс:*[29 сентября 2019 в 15:02](#)

Здравствуйте, скажите, пожалуйста, как возвратить значение переменной из while?

[Ответить](#)25. *Игорь:*[20 сентября 2019 в 13:54](#)

додумался только до такого варианта!

```
1  #include "pch.h"
2  #include <iostream>
3  using namespace std;
4
5  int main()
6  {
7      int outer=1;
8      while (outer <= 5)
9      {
10         int miss = 4;
11         while (miss >= outer)
12         {
13             cout << " ";
14             --miss;
15         }
16         int inner = 1;
17         while (inner <= outer)
18             cout << inner++ << " ";
19         cout << "\n";
20         ++outer;
21     }
22 }
```

[Ответить](#)26. *Ivan:*[30 августа 2019 в 12:00](#)

```
1  #include <iostream>
2
3  int main()
4  {
5      int outer = 5;
```

```

6   while (outer >= 1)
7   {
8       int inner = 5;
9       while (inner >= 1)
10          if (inner > (6 - outer)) {
11              std::cout << " ";
12              inner--;
13          }
14          else
15              std::cout << inner-- << " ";
16      std::cout << "\n";
17      outer--;
18  }
19
20  return 0;
21 }

```

Ответить



27. **zashiki:**

[28 августа 2019 в 19:25](#)

тут ромб-челлендж, значит...

Попробовала так, чтобы цифры в обратном порядке шли к центру.
вводить лучше до 9, так как дальше числа едут.

Как сократить код? И как сделать, чтобы числа не ехали, если больше 9?

```

1  #include<iostream>
2
3  int main()
4  {
5      int halfWide;    //пол-ширины ромба
6      std::cout<<"enter a number from 2 to 9: ";
7      std::cin>>halfWide;
8      int halfVisota{1}; //пол-высоты ромба
9
10     while(halfVisota<=halfWide) //цикл для верхнего треугольника у ромба
11     {
12         int numMirror=halfWide; //numMirror ряд чисел на одной строке
13         while (numMirror>1) //как в примере урока, строим ряд с пробелами
14         {
15             if (numMirror>halfVisota)
16                 std::cout<<" ";
17             else
18                 std::cout<<numMirror<<" ";
19             --numMirror;
20         }

```



```

21     while(numMirror<=halfVisota)    //достаиваем возрастающий ряд
22     {
23         std::cout<<numMirror++<<" ";
24     }
25     std::cout<<"\n";
26     ++halfVisota;
27 }
28
29 halfVisota=halfWide-1;
30 while(halfVisota>=1) //цикл для нижнего треугольника у ромба
31 {
32     std::cout<<" ";
33     int numMirror=halfWide-1;
34     while(numMirror>1)//как в примере урока, строим ряд с пробелами
35     {
36         if(numMirror>halfVisota)
37             std::cout<<" ";
38         else
39             std::cout<<numMirror<<" ";
40         --numMirror;
41     }
42     while(numMirror<=halfVisota)//достаиваем ряд
43     {
44         std::cout<<numMirror++<<" ";
45     }
46     std::cout<<"\n";
47     --halfVisota;
48 }
49 return 0;
50 }

```

Ответить



1. zashiki:

30 августа 2019 в 12:22

как выровнить долго думала, а оказалось все очень примитивно.
 надо там, где строят пробелы, прописать лишнюю строчку.
 для верхнего треугольника

```

1     if (numMirror>halfVisota)
2     {
3         std::cout<<" ";
4         if(numMirror>=10)//а это дополнительный сдвиг из пробелов для чисел б
5         std::cout<<" ";
6     }

```

для нижнего треугольника:

```

1 while(halfVisota>=1)
2 {
3     (halfWide>9)?std::cout<<"    ":std::cout<<" ";//если чисел больше 9, то
4     .
5     .
6     .
7     if(numMirror>halfVisota)
8     {
9         std::cout<<" ";
10        if(numMirror>=10)//а это дополнительный сдвиг из пробелов для чисел
11        std::cout<<" ";
12    }

```

так можно делать треугольники из 2значных цифр, с шириной насколько позволяет ide

Ответить



2. Saliwer:

[2 сентября 2019 в 17:24](#)

Получилось 2 варианта решения до 3-значных чисел.

1 вариант базируюсь на знаниях полученных до этого урока.

Для красивого вывода используется функция printf() и ввода scanf_s().

```

1 #include <iostream>
2
3 void PrintRhomb(int value);
4 int main() {
5     int x;
6     printf("Enter the value: ");
7     scanf_s("%d", &x);
8     if (x <= 1 || x > 999)
9         printf("1");
10    else
11        PrintRhomb(x);
12    system("pause");
13    return 0;
14 }
15 void PrintRhomb(int value){
16     //Количество строк
17     int countRow = value + (value-1);
18     //Количество столбцов
19     int countColum = countRow;
20     //счётчик строк
21     int i = 1;
22     //Флаг, по достижению которого заполняется ромб
23     int tempValue = value;
24     //Цикл по строкам

```

```

25 while(i <= countRow){
26     //Счётчик столбцов
27     int j = 1;
28     //Заполнитель ромба
29     int fill = 1;
30     //Цикл по строкам
31     while(j <= countColumn)
32     {
33         //Ограничиваем ромб на 2 части
34         //1 - Вывод пробелов
35         if (j < tempValue || j >= (countColumn - tempValue + 2))
36             printf("  ");
37         //2 - Вывод цифр
38         else
39         {
40             //Если левая часть ромба - инкрементируем заполнитель
41             if (j < value)
42                 printf("%3d", fill++);
43             //Если правая - декрементируем
44             else
45                 printf("%3d", fill--);
46         }
47         //следующий столбец
48         ++j;
49     }
50     //Следующая строка
51     printf("\n");
52     if (i < value)
53         tempValue--;
54     else
55         tempValue++;
56     ++i;
57 }
58 }

```

2 вариант — используется цикл for() и тернарный оператор.

Для красивого вывода использую функцию std::setw() из библиотеки <iomanip>

```

1 #include <iostream>
2 #include <iomanip>
3 int main() {
4     std::cout << "Enter the value: ";
5     int x;
6     std::cin >> x;
7     if (x <= 1 || x > 999)
8         std::cout << "1";
9     else
10        PrintRhomb(x);

```

```

11     system("pause");
12     return 0;
13 }
14
15 void PrintRhomb(int value) {
16     int countRow = value + (value - 1);
17     int tempValue = value;
18     for (int i = 1; i <= countRow; ++i)
19     {
20         for (int j = 1, fill = 1; j <= countRow; ++j) {
21             std::cout << std::setw(3);
22             if (j < tempValue || j >= (countRow - tempValue + 2))
23                 std::cout << " ";
24             else
25                 (j < value) ? std::cout << fill++ : std::cout << fill--;
26         }
27         std::cout << '\n';
28         (i < value) ? tempValue-- : tempValue++;
29     }
30 }

```

Ответить



1. **zashiki:**

[3 сентября 2019 в 16:53](#)

спасибо за 2й вариант, только жаль, что не вставлены комментарии к строкам.

Попробовала повторить для убывающего к центру ряда, но полностью не копировала, вышло хуже.

```

1  #include <iostream>
2  #include<iomanip>
3  void printRomb(int num);
4
5  int main()
6  {
7      std::cout<<"введите число от 1 до 999: ";
8      int num;
9      std::cin>>num;    //num- пользовательское число
10
11     if(num<=1||num>999)    //проверка ввода
12         std::cout<<"число вне заданного диапазона";
13     else
14         printRomb(num);
15
16     return 0;
17 }
18

```

```

19 void printRomb(int num)          //функция ромба
20 {
21     int numRows{1}, ii{1};        //numRow - числовой ряд
22     while(++ii<=2*num)             //ii-счетчик высоты ромба
23     {
24         int rowWide {num+numRow}, printNumRow{numRow}; //printNumRow -
25         while(--rowWide>0)         //rowWide-ширина ряда
26         {
27             std::cout << std::setw(3); //setw для красивого вывода
28             if(rowWide>numRow*2-1) //в этом условии строится ряд (проб
29             std::cout<<" ";
30             else
31                 (rowWide>numRow)? std::cout<<printNumRow--:std::cout<<printNumRow
32         }
33         std::cout<<"\n";
34         (ii<=num)?++numRow:--numRow; //до половины высоты ромба число
35     }
36 }

```

Ответить



1. Saliwer:

[4 сентября 2019 в 10:49](#)

У Вас неплохо получилось:). Можно было в условии while(—rowWide > 0) убрать >0, если изначально число положительное, то по достижении переменной rowWide значения 0 (0 = false), произойдёт выход из цикла. Также, что б ромбик был красивее, можно было в строчке int printNumRow{numRow}; Вместо numRows использовать просто num.

Добавил комменты:)

```

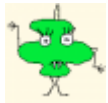
1  int main() {
2      std::cout << "Enter the value: ";
3      int x;
4      std::cin >> x;
5      if (x <= 1 || x > 999)
6          std::cout << "1";
7      else
8          PrintRhomb(x);
9      system("pause");
10     return 0;
11 }
12
13 void PrintRhomb(int value) {
14     int countRow = value + (value - 1); //countRow - количество стр
15     int tempValue = value; // tempValue - отвечает за количество в
16     for (int i = 1; i <= countRow; ++i) // i - счётчик строк

```

```

17     {
18         for (int j = 1, fill = value; j <= countRow; ++j) // j - с
19         {
20             std::cout << std::setw(3); //std::setw(3) - устанавлив
21             //вырезаем место в левой части ромба для пробелов ИЛИ
22             if (j < tempValue || j >= (countRow - tempValue + 2))
23                 std::cout << " ";
24             //Если условие выше ложно - значит мы в столбике где н
25             else
26                 //Проверяем в какой мы части ромба (левой или прав
27                 (j < value) ? std::cout << fill-- : std::cout << f
28         }
29         std::cout << '\n';
30         //Количество выводимых пробелов в верхней половине ромба t
31         (i < value) ? tempValue-- : tempValue++;
32     }
33 }

```



28. Алексей:

[17 июля 2019 в 16:14](#)

Второе сделал следующим образом.

```

1  #include <iostream>
2
3  int main()
4  {
5      int iii = 97;
6      while (iii <= 122)
7      {
8          char ch(iii);
9          std::cout << ch << " - " << static_cast<int>(ch) << "\n";
10         ++iii;
11     }
12 }

```

[Ответить](#)



29. Dima Malovanyu:

[11 мая 2019 в 18:19](#)

Тоже интересная программа с использованием while, если писать самому, то далеко не очевидная. Попробуйте сами!

Задание: Напишите программу используя выкл while, которая просит ввести число от 1 до 9, а потом делает из него ромб чисел, по типу

```

1
1 2 1
1 2 3 2 1
1 2 1
1

```

Важно: программа должна выводить числа по уменьшению (1 всегда по краю)
УДАЧИ!!!

```

1  #include <iostream>
2
3  void stepone(int pyramidnumber){
4      int hight = 1;
5      while (hight <= pyramidnumber)
6      {
7          int width = pyramidnumber * 2 - 1;
8          int stat = width/2;
9          while (width != 0)
10         {
11             if (width == hight + stat)
12             {
13                 int tempa = 1;
14                 int tempb = hight-1;
15                 while (tempa <= hight)
16                 {
17                     std::cout << tempa << " ";
18                     ++tempa;
19                 }
20                 while (tempb != 0)
21                 {
22                     std::cout << tempb << " ";
23                     --tempb;
24                 }
25             }
26             else std::cout << " ";
27             --width;
28         }
29         std::cout << std::endl;
30         ++hight;
31     }
32 }
33
34
35 void steptwo (int pyramidnumber){
36     int hight = pyramidnumber-1;
37     while (hight !=0)
38     {
39         int width = pyramidnumber;
40         while (width != 0 )

```

```
41     {
42         if (hight - width == 0)
43         {
44             int tempa = 1;
45             int tempb = hight - 1;
46             while (tempa <= hight)
47             {
48                 std::cout << tempa << " ";
49                 ++tempa;
50             }
51             while (tempb != 0)
52             {
53                 std::cout << tempb << " ";
54                 --tempb;
55             }
56
57             }
58             else std::cout << " ";
59             --width;
60         }
61
62         std::cout << std::endl;
63         --hight;
64     }
65 }
66
67 int main() {
68     std::cout << "Enter doublepiramid number: ";
69     int a;
70     std::cin >> a;
71     stepone(a);
72     steptwo(a);
73     return 0;
74 }
```

[Ответить](#)



1. *Алексей:*

[18 июля 2019 в 12:35](#)

Из головы не выходит)
Вот как вывести после)

В тесте простые проги, просто немного фантазии, хотя 4й сразу не сделал.

[Ответить](#)



2. *Mirovengil:*

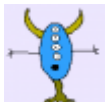
[21 августа 2019 в 18:29](#)

Можно короче (знаю, что лучше функциями):

```

1  #include <iostream>
2  #include <cstdlib>
3  int main()
4  {
5      setlocale(LC_ALL, "Rus");
6      int a;
7      std::cin >> a;
8      int i = 0;
9      while (i < a){
10         int j = 1;
11         while (j <= i){
12             std::cout << j++ << " ";
13         }
14         while (j >= 1){
15             std::cout << j-- << " ";
16         }
17         std::cout << std::endl;
18         i++;
19     }
20 }
```

[Ответить](#)



3. *zashiki:*

[30 августа 2019 в 17:04](#)

попыталась, чтобы ромб был ровным и красивым при числах больше 9.

```

1  #include<iostream>
2
3  int upperTriangle(int halfWide,int halfHeight,int row)//функция для верхнего
4  {
5      while(halfHeight<=halfWide)//цикл для левой части, как в примере урока с про
6      {
7          row=halfHeight-halfWide+1;
8          while(row<halfHeight)
9          {
10             if(row<1)
11                 std::cout<<" ";
12             else
13             {
14                 if(row<9)//ставятся числа и заодно выравниваются пробелы
15                     std::cout<<row<<" ";
16                 else
17
```

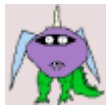
```
18         std::cout<<row<<" ";
19     }
20     ++row;
21 }
22 while(row>=1)//цикл для правой части
23 {
24     if(row<=10)//ставятся числа и заодно выравниваются пробелы
25     std::cout<<row<<" ";
26     else
27     std::cout<<row<<" ";
28     --row;
29 }
30 std::cout<<"\n";
31 ++halfHeight;
32 }
33 return halfHeight;
34 }
35
36 void lowerTriangle(int halfWide,int halfHeight,int row)//функция для нижнего
37 {
38     while(halfHeight>=1)//цикл для левой части, как в примере урока с пробелами
39     {
40         --halfHeight;
41         row=halfHeight-halfWide;
42         while(row<halfHeight)
43         {
44             if(row<1)
45                 std::cout<<" ";
46             else
47                 {if(row<9)//ставятся числа и заодно выравниваются пробелы
48                 std::cout<<row<<" ";
49                 else
50                 std::cout<<row<<" ";
51                 }
52             ++row;
53         }
54         row-=2;
55         if(row==9||row==8)//не знаю как по другому, без этого условия эти 2 правых р
56         std::cout<<" ";
57         while(row>=1)//цикл для правой части
58         { if(row<=10)//ставятся числа и заодно выравниваются пробелы
59         std::cout<<row<<" ";
60         else
61         std::cout<<row<<" ";
62         --row;
63         }
64         std::cout<<"\n";
65     }
66 }
```

```

67 }
68
69 int main()
70 {
71     int halfWide, halfHeight{1}, row{0}; //halfWide-полширины ромба, halfHeight-пол
72     std::cout<<"enter a number: ";
73     std::cin>>halfWide;
74
75     halfHeight=upperTriangle(halfWide, halfHeight, row); //функция для верхнего тре
76     lowerTriangle(halfWide, halfHeight, row); //функция для нижнего треугольника
77     return 0;
78 }

```

[Ответить](#)



30. Максим:

[2 мая 2019 в 14:56](#)

```

1  #include <iostream>
2
3  int main()
4  {
5      int a=1; //определяем переменную для счета строк.
6      while (a<=5)
7      {
8          int b = 5; /*определяем переменную для счета столбцов.
9                      Поскольку числа уменьшаются слева на право
10                     её значение должно быть большим числом*/
11          while (b>a) /*цикл печатает пробел если номер столбца
12                      (или число) меньше номера строки*/
13          {
14              std::cout << " ";
15              b--;
16          }
17          while (b >= 1) /*цикл печатает число если номер столбца
18                        (или число) больше номера строки*/
19          {
20              std::cout << b;
21              b--;
22          }
23          std::cout<< "\n";
24          a++;
25      }
26
27 }

```

[Ответить](#)



31. Константин:

[19 апреля 2019 в 13:10](#)

Юра, если захочешь, можешь этот код как задание №5 вывесить. Я его написал в процессе выполнения задания №4 (которое, как ни парадоксально, я не смог осилить:-) Ромб из чисел:

```

1  #include <iostream>
2
3  int main()
4  {
5      int outer = 1;
6      while(outer <= 9)
7      {
8          int inner = 9;                                // 0
9          while(inner >= 1)                               // 1 0
10         { inner-- ;                                     // 2 1 0
11             if(inner >= outer)                         // 3 2 1 0
12                 std::cout << ' ';                     // 4 3 2 1 0
13             else std::cout << inner << " ";} // 5 4 3 2 1 0
14         std::cout << "\n";                             // 6 5 4 3 2 1 0
15         ++outer;                                       // 7 6 5 4 3 2 1 0
16     }                                                // 8 7 6 5 4 3 2 1 0
17     int auter = 9;                                   // 8 7 6 5 4 3 2 1 0
18     while(auter >= 1)                               // 7 6 5 4 3 2 1 0
19     {                                                // 6 5 4 3 2 1 0
20         int enner = 9;                               // 5 4 3 2 1 0
21         while(enner >= 1)                             // 4 3 2 1 0
22         { enner-- ;                                   // 3 2 1 0
23             if(enner >= auter)                       // 2 1 0
24                 std::cout << ' ';                   // 1 0
25             else std::cout << enner << " ";} // 0
26         std::cout << "\n";
27         --auter;
28     }
29 }
30 return 0;

```

[Ответить](#)


32. Владислав:

[23 марта 2019 в 14:29](#)

Все разобрался 😊

Вот решение если кому нужно делал в DEV C++

```

1  #include<conio.h>

```

```
2 #include<iostream>
3 #include<stdio.h>
4 #include<stdlib.h>
5
6 int main()
7 {   float k,s,i;
8     i = 0.1;
9     s = 0;
10    k = -1;
11    while(i<=1.9)
12    {
13        s=s+i*k;
14        i=i+0.3;
15        k=-k;
16    };
17    std::cout << "Vyrazhenie=" <<s<<std::endl;
18    return 0;
19 }
```

[Ответить](#)33. *Владислав:*[23 марта 2019 в 14:18](#)

Порылся вроде должно выглядеть так, но не работает.

```
1 #include<conio.h>
2 #include<iostream>
3 #include<stdio.h>
4 #include<stdlib.h>
5
6 int main()
7 {   int k,s,i;
8     i = 0.1;
9     s = 0;
10    k = -1;
11    while(i < 1.9)
12    {
13        s=-s+i*k;
14        i=i+0.3;
15        k=-k;
16        std::cout << "Vyrazhenie=" <<s<<std::endl;
17    }
18    return 0;
19 }
```

[Ответить](#)



34. Владислав:

[23 марта 2019 в 13:41](#)

Всем, здравствуйте! Извиняюсь за наверное слишком простой вопрос, но может кто помочь сделать такое задание: Написать программу для вычисления заданного выражения и вывода на экран полученного значения, используя циклический оператор While. Выражение: $0,1 + 0,4 - 0,7 + \dots - 1,9$. Не могу сам сообразить как это сделать, в программировании ноль не сделать нужно 😞

[Ответить](#)

35. Вячеслав:

[11 марта 2019 в 02:45](#)

вот задание №3:

```
1 #include "pch.h"
2 #include <iostream>
3
4 int main()
5 { //объявляем локально пространство имен
6     using namespace std;
7     //инициализируем целочисленную переменную outer значением 5
8     //с которого должен начаться вывод в консоль
9     int outer = 5;
10    //создаем внешний цикл с условием от какой и до какой цифры должен быть вывод
11    while (outer >= 1)
12    { //инициализируем переменную inner значением outer внутри цикла
13        //так как она там и используется
14        int inner = outer;
15        //создаем внутренний цикл с использованием переменной inner
16        //с тем же условием что и внешний цикл
17        while (inner >= 1)
18        { //выводим в консоль пост декремент переменной inner
19            //если бы мы применили пре декремент вывод начался бы с цифры 4
20            cout << inner-- << " ";
21            //каждая инвертация должно начинаться с новой строки
22            cout << "\n";
23            //тут можно применить пре декремент т. к. переменная объявлена вне цикла
24            --outer;
25        }
26    }
27    return 0;
28 }
```

[Ответить](#)



36. Вячеслав:

[11 марта 2019 в 02:12](#)

вот первое задание :

```

1 #include "pch.h"
2 #include <iostream>
3
4 int main()
5 { //объявляем локально пространство имен using statements
6     using namespace std;
7     //объявляем переменную символьного типа с первой буквой английского алфавита
8     char symbols('a');
9     //объявляем цикл while с условием вывести все буквы английского алфавита до z
10    while (symbols <= 'z')
11    { //выводим в консоль по одной букве и соответствующей ей цифре в таблице ASCII
12        //преобразуем с помощью static_cast тип char в тип int
13        cout << symbols << " " << static_cast<int>(symbols) << "\n";
14        //добавляем по одной паре(буква - цифра) в каждую строку
15        ++symbols;
16    }
17
18    return 0;
19 }

```

[Ответить](#)

37. Alexey:

[8 февраля 2019 в 10:11](#)

```

1 for (int i = 1; i < 6; i++) {
2     for (int k = 5 - i; k > 0; k--) {
3         std::cout << " ";
4     }
5     for (int j = 1; j <= i; j++) {
6         std::cout << j << " ";
7     }
8     std::cout << "\n";
9 }

```

или так:

```

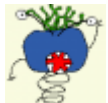
1 while (i < 6) {
2     int j{ 1 }, k{ 5 };
3     while (k - i > 0) {
4         std::cout << " ";

```

```
5 | k-;  
6 | }  
7 | while (j <= i) {  
8 | std::cout << j << " ";  
9 | j++;  
10 | }  
11 | std::cout << "\n";  
12 | i++;  
13 | }
```

[Ответить](#)38. *kmish:*[4 февраля 2019 в 16:14](#)

```
1 | int main()  
2 | {  
3 |     int outer = 1;  
4 |     int mass_size = 5;  
5 |     while (outer <= mass_size)  
6 |     {  
7 |         int space = mass_size - outer;  
8 |         while (space >= 1)  
9 |         {  
10 |             std::cout << " ";  
11 |             space -= 1;  
12 |         }  
13 |  
14 |         int inner = outer;  
15 |         while (inner >= 1)  
16 |         {  
17 |             std::cout << inner << " ";  
18 |             inner -= 1;  
19 |         }  
20 |  
21 |         // делаем новую строку в конце каждого ряда  
22 |         std::cout << "\n";  
23 |         outer += 1;  
24 |     }  
25 |  
26 |     return 0;  
27 | }
```

[Ответить](#)39. *Алексей:*[4 октября 2018 в 07:34](#)

Задание 3:

```
1 #include <iostream>
2
3 int main()
4 {
5     int outer = 1;
6     while (outer <= 5)
7     {
8         //изменяем цикл в 3 действия :)
9         int inner = 5; //изменяем inner с 1 на 5
10        while (inner >= outer) //меняем сравнение с <= на >=
11            std::cout << inner-- << " "; //уменьшаем inner
12
13        // делаем новую строку в конце каждого ряда
14        std::cout << "\n";
15        ++outer;
16    }
17    return 0;
18 }
```

[Ответить](#)

40. Adel:

[20 сентября 2018 в 15:02](#)

Здравствуй Юрий! Спасибо за такой пост! Все было очень понятно, кроме , посл. примера(треугольник). Можете объяснить каждую строку?!!!

[Ответить](#)

41. Игорь:

[19 сентября 2018 в 04:18](#)

```
1 #include <iostream>
2
3 int main()
4 {
5     const int SIZE = 5; // Константа, кол-во рядов
6     int outer = 1;
7     while (outer <= SIZE) // 5 рядов цифр
8     {
9         int inner = outer; // Внутренняя переменная зависит от внешней,
10        while(inner++ < SIZE) // (SIZE - outer) пробелов внутреннего цикла
11            std::cout << " ";
12        inner = outer;
13        while (inner > 0) // inner == кол-ву цифр, начальное значение == outer
14    }
```

```

15         std::cout << inner-- << " ";
16
17         std::cout << "\n"; // Перевод на новую строку каждую итерацию внешнего цикла
18         ++outer; // Увеличиваем начальное значение и кол-во цифр для вывода.
19     }
20
21     return 0;
    }

```

[Ответить](#)42. *Tairak:*[14 июля 2018 в 23:35](#)

Спасибо за уроки, но я не могу понять как выводиться результат в последнем примере. Если цикл while повторяет себя до выполнения условия, то почему std::cout сработывает с каждым разом на один раз больше? Почему на последней строчке 5 цифр, хоть и используется только один cout?

[Ответить](#)1. *Юрий:*[15 июля 2018 в 15:40](#)

Потому что используется инкремент outer:

```
1 ++outer;
```

[Ответить](#)43. *Денис:*[21 мая 2018 в 19:50](#)

Как вариант:

```

1 #include <iostream>
2
3 int main()
4 {
5     int outer = 1;
6     while (outer <= 5)
7     {
8         int space = 4;
9         while (space >= outer)
10        {
11            std::cout << " ";
12            --space;
13        }

```

```

14     int inner = outer;
15     while (inner >= 1)
16     {
17
18         std::cout << inner-- << " ";
19     }
20     // делаем новую строку в конце каждого ряда
21     std::cout << "\n";
22     ++outer;
23 }
24
25 return 0;
26 }


```

[Ответить](#)

1.  Юрий:
[21 мая 2018 в 20:54](#)

Неплохо.

[Ответить](#)

44.  master114:
[8 мая 2018 в 12:26](#)

Поделюсь своим вариантом для 4 задачи

```

1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      int outj = 1;           // счетчик для строк
7      while (outj <= 5)      // магич 5 - это число строк
8      {
9          int intj = outj;    // счетчик для чисел
10
11         int countProbel = 5- outj;    // количество пропущенных чисел
12         while (countProbel > 0)        // цикл для вывода пустых чисел в строке
13         {
14             cout << " ";
15             --countProbel;
16         }
17
18         while (intj > 0)                // чисел в каждой строке не больше кол
19         {
20             cout << intj << " ";

```

```
21         --intj;  
22  
23     }  
24     cout << "\n";  
25     ++outj;  
26 }  
27  
28 return 0;  
29 }
```

[Ответить](#)

1. *Юрий:*
[8 мая 2018 в 23:29](#)

Как вариант может быть.

[Ответить](#)

45. *Николай:*
[22 марта 2018 в 01:13](#)

Добрый вечер! Скажите, пожалуйста, на каком уроке и будет ли вообще разбор таких понятий, как printf и scanf?

[Ответить](#)

1. *Юрий:*
[22 марта 2018 в 22:31](#)

Привет, printf и scanf в этих уроках не рассматриваются, но, возможно, позднее напишу статью о них (но это точно будет не скоро).

[Ответить](#)

46. *Максим:*
[15 марта 2018 в 05:56](#)

```
1 #include "stdafx.h"  
2 #include <iostream>  
3  
4 int main()  
5 {  
6     // первая переменная отвечает за количество строк и столбцов сразу  
7     // и изменив одно значение можно одновременно менять размер всей лесенки  
8     int stage(5);  
9     // вторая переменная - счетчик цикла
```

```
10  int count(1);
11  while (count <= stage) {
12
13      // третья переменная для обратного отсчета и вывода в консоль
14      int var = stage;
15
16      while (var >= 1) {
17
18          // внутренний цикл выставляет пробелы, 1я строка меньше на 1 пробел, 2я - на
19          // после каждого пробела уменьшаем 3ю переменную
20          while (var > count) {
21              std::cout << " " << "\t";
22              var--;
23          }
24          // на выходе из цикла, попадаем на уровень выше и уже выводим значения var в консоль
25          std::cout << var << "\t";
26          var--;
27      }
28      // переходим на следующую строку и уменьшаем количество пробелов с конца
29      std::cout << "\n";
30      count++;
31  }
32  system("PAUSE");
33  return 0;
34 }
```

[Ответить](#)

Добавить комментарий

Ваш E-mail не будет опубликован. Обязательные поля помечены *

Имя *

Email *

Комментарий






☐ Сохранить моё Имя и E-mail. Видеть комментарии, отправленные на модерацию

☐ Получать уведомления о новых комментариях по электронной почте. Вы можете [подписаться](#) без комментирования.

[Отправить комментарий](#)

[TELEGRAM](#)  [КАНАЛ](#)
[ПАБЛИК](#) 

ТОП СТАТЬИ

-  [Словарь программиста. Сленг, который должен знать каждый кодер](#)
-  [Урок №1. Введение в программирование](#)
-  [70+ бесплатных ресурсов для изучения программирования](#)
-  [Урок №1: Введение в создание игры «SameGame» на C++/MFC](#)
-  [Урок №4. Установка IDE \(Интегрированной Среды Разработки\)](#)

- [Ravesli](#)
- - [О проекте/Контакты](#) -
- - [Пользовательское Соглашение](#) -
- - [Все статьи](#) -
- Copyright © 2015 - 2020