

## Программирование на C++. Лабораторная работа №1.

Выполните следующие задания .

### Часть 1.

1.1. Добавьте несколько круглых скобок (ни одной, одну или две пары) в приведенный ниже код, чтобы получить ожидаемые результаты. Попробуйте сделать это перед запуском программы.

```
#include <iostream>
#include <iomanip>
#include <string>
int main()
{
    float v=2;
    float result = v + 1 * 2;
    std::cout << "result: " << result << " expected result : 6" << std::endl;

    result = v + 1 * v + 2 * 2;
    std::cout << "result: " << result << " expected result : 24" << std::endl;

    result = v - 1 * 2 + 2 * 2;
    std::cout << "result: " << result << " expected result : 6" << std::endl;

    result = v + v * v + v * 2;
    std::cout << "result: " << result << " expected result : 32" << std::endl;

    result = (int)v % 2 * v + 2 * 2;
    std::cout << "result: " << result << " expected result : 0" << std::endl;
}
```

**1.2.** Пасха - это так называемый подвижный праздник, и его дата зависит от двух астрономических явлений: начала весны в Северном полушарии и первого полнолуния, наступающего после нее. Вы можете подумать, что определить дату Пасхи сложно, это связано со сложными астрономическими вычислениями, но есть алгоритм Карла Гаусса определения даты пасхи. Существует много вариаций алгоритма.

Одна из более простых форм, адаптированных для 20-го и 21-го веков.

Единственные данные, которые он использует - это номер года.

Алгоритм:

1. разделите год на **19** и найдите остаток – отнесите его к **k**;
2. разделите год на **4** и найдите остаток – отнесите его к **b**;
3. разделите год на **7** и найдите остаток – присвойте его **c**;
4. возьмите **a**, умножьте его на **19**, добавьте **24**, разделите на **30** и найдите остаток – присвойте его **d**
5. разделите **(2b + 4c + 6d + 5)** на **7** и найдите остаток - присвойте его **e**;
6. проверьте значение **d + e**;
7. если оно меньше **10**, Пасха приходится на **(d + e + 22)** день марта;
8. в противном случае он приходится на **(d + e – 9)** день апреля;
9. конец

### Задание:

Напишите программу для определения даты Пасхи. Программа должна запросить у пользователя номер года и вывести дату в виде Дня месяца, например, 5 апреля.

*Пример ввода*  
2016

*Пример ввода*  
1980

*Пример ввода*  
2026

*Пример вывода*  
March 27

*Пример вывода*  
April 6

*Пример вывода*  
April 5

**1.3.** В 1937 году математик Лотар Коллатц сформулировал следующую гипотезу (она остается неразгаданной и сейчас):

1. возьмите любое неотрицательное и ненулевое целое число и назовите его **c0**;
2. если оно четное, то вычислите новый **c0** как **c0 / 2**
3. если это нечетно, вычислите новый **c0** как **3 \* c0 + 1**
4. если **c0 ≠ 1**, переходите к пункту 2

Гипотеза гласит, что, независимо от начального значения **c0**, оно всегда (всегда!) будет равно **1**.

Например, для числа 3 получаем:

3 — нечётное,  $3 \times 3 + 1 = 10$

10 — чётное,  $10 : 2 = 5$

5 — нечётное,  $5 \times 3 + 1 = 16$

16 — чётное,  $16 : 2 = 8$

8 — чётное,  $8 : 2 = 4$

4 — чётное,  $4 : 2 = 2$

2 — чётное,  $2 : 2 = 1$

**1**

#### **Задание:**

Напишите программу, которая считывает одно натуральное число и выполняет вышеуказанные шаги до тех пор, пока **c0** остается отличным от **1**.

Подсчитайте шаги, необходимые для достижения цели.

Ваш код должен выводить все промежуточные значения **c0**.

Подсказка: самая важная часть - преобразовать идею Коллатца в цикл "while".

Протестируйте свой код, используя следующие данные:

*Пример ввода*

15

*Пример вывода*

46

23

70

35

106

53

160

80

40

20

10

5

16

8

4

2

1

steps = 17

*Пример ввода*

16

*Пример вывода*

8

4

2

1

steps = 4

*Пример ввода*

1023

*Пример вывода*

3070

1535

4606

2303

6910

3455

10366

5183

```

15550
7775
23326
11663
34990
17495
52486
26243
78730
39365
118096
59048
29524
14762
7381
22144
11072
5536
2768
1384
692
346
173
520
260
130
65
196
98
49
148
74
37
112
56
28
14
7
22
11
34
17
52
26
13
40
20
10
5
16
8
4
2
1
steps = 62

```

1.4. Из курса математического анализа известно, что число  $\pi$  может быть вычислено с использованием ряда Лейбница по формуле:

$$\frac{\pi}{4} = \frac{1}{1} - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \square$$

Обратите внимание на то, что ряд является знакочередующимся: члены ряда с нечетными номерами входят в формулу со знаком плюс, а те члены ряда, номера которых четные – со знаком минус.

Если условиться, что нумерация членов ряда начинается с нуля, то общий член ряда в таком случае можно представить формулой:

$$a_i = \frac{1}{2i + 1}$$

### Задание

Написать и отладить программу, вычисляющую значение числа  $\pi$ , используя ряд Лейбница. Пользователь вводит некоторое натуральное число – количество итераций при вычислении числа  $\pi$  (иными словами, задается количество дробей, которые необходимо прибавить/вычесть при вычислении). Очевидно, что при малых значениях числа итераций полученный результат может заметно отличаться от истинного значения числа  $\pi$ . Чем большим будет количество итераций, тем ближе полученное значение будет к истинному.

*Пример ввода:*

10000

*Пример вывода:*

Pi = 3.1414926535900344895

## Часть 2.

2.1. Что будет выведено на экран:

```
#include <iostream>

using namespace std;

int main()
{
    float var = .1;

    var = var + 1.;
    var = var + 1e1;
    var = var + 1e-2;
    cout << var << endl;
}
```

2.2. Что будет выведено на экран:

```
#include <iostream>

using namespace std;

int main()
{
    int i = 10, j = 3;
    int k = (i % j * i / 3) / (j % i - j / i);

    cout << k << endl;
}
```

2.3. Что будет выведено на экран, если введено значение i=10?

```
#include <iostream>

using namespace std;

int main()
{
    int i, j;

    cin >> i;
    j = i;
    if (i >= 9)
    |   i += 9 - i;
    j /= i;
    cout << j;
}
```

2.4. Что будет выведено на экран, если введено значение a=3.0?

```
#include <iostream>

using namespace std;

int main()
{
    float a, b;

    cin >> a;
    b = a / 2;
    if (b >= 1)
    |   a += b;
    if (a > b)
    |   b += a;
    cout << a + b;
}
```

2.5. Что будет выведено на экран, если введено значение a=2.0?

```
#include <iostream>
#include <cmath>

using namespace std;

int main()
{
    float a;

    cin >> a;
    a *= a;
    a *= a;
    a = sqrtf (a) * sqrtf (a);
    cout << a;
}
```

**Часть 3.** Напишите и отладьте следующие программы:

- 3.1. Дано двузначное число. Составить программу определения, является ли сумма его цифр двузначным числом. Если нет, то через, сколько чисел встретится первое двузначное число, удовлетворяющее этому условию.
- 3.2. Дано трехзначное число, у которого число единиц не превосходит числа сотен. Проверить, является ли оно палиндромом, т. е. числом, которое одинаково читается слева направо и справа налево. Если не является, то вывести ближайшее следующее число-палиндром.
- 3.3. Составить программу для определения вида параллелограмма по прилежащим сторонам  $a$ ,  $b$  и углу между ними  $x$  (в градусах). Определить один из видов: ромб, прямоугольник, квадрат, обыкновенный параллелограмм.
- 3.4. Даны четыре числа  $a$ ,  $b$ ,  $c$ ,  $d$ . Известно, что одно отлично от других, равных между собой. Вывести это число на экран. Например, даны числа  $a = 5$ ,  $b = 6$ ,  $c = 5$ ,  $d = 5$ . Требуется вывести 6.
- 3.5. Дано четырехзначное число. Определить:
- а) равна ли сумма двух первых его цифр сумме двух его последних цифр;
  - б) кратна ли трем сумма его цифр;
  - с) кратно ли произведение его цифр числу  $a$ .
- 3.6. В небоскребе  $N$  этажей и всего один подъезд; на каждом этаже по 3 квартиры; лифт может останавливаться только на нечетных этажах. Человек садится в лифт и набирает номер нужной ему квартиры  $M$ . На какой этаж лифт должен доставить пассажира?
- 3.7. Удвоить трехзначное число, введенное с клавиатуры, если оно содержит в своей записи хотя бы одну 1, и возвести в квадрат в противном случае.