

Lab 2.3.2 Some actual evaluations – finding the value of π

Objectives

Familiarize the student with:

- using the **for** loop;
- classical iterative algorithms;
- the question of a calculation's accuracy.

Scenario

One of the methods used to find the value of π (let's add: not a very effective method) is the Leibniz formula. At first glance, it looks complicated, but if you look at it carefully, you'll see a very simple recurrence and (we can bet on it!) you'll be able to imagine a draft of a very simple code implementing Leibniz's idea.

Here you are:

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

Note:

- you need to add a number of fractions – the sum will show you an approximate value of a quarter of π ;
- some of the fractions are positive, some are negative – can you see the regularity?

Leibniz's formula needs a very large number of fractions to achieve good accuracy (you'll see this soon), but that's not a problem – we don't actually want to discover the value of π . We just want to check if we can find it.

Your task is to complete the code below. The code should ask the user to enter a number of totaled fractions (in other words, the number of iterations) and to print the computed value of π . As we need good accuracy and a very large number of iterations, we use a **double** instead of a **float** and a **long** instead of an **int**.

Test your code using the data we've provided.

```
#include <iostream>

using namespace std;

int main(void) {
    double pi4 = 0.;
    long    n;

    cout << "Number of iterations? ";
    cin >> n;

    // Insert your code here

    cout.precision(20);
    cout << "Pi = " << (pi4 * 4.) << endl;
    return 0;
}
```

Example input

10000

Example output

Pi = 3.1414926535900344895

Example input

1000000

Example output

Pi = 3.1415916535897743245

Example input

1000000000

Example output

Pi = 3.1415926525880504272