

## Lab 2.3.1 Collatz's hypothesis

### Objectives

Familiarize the student with:

- using the **while** loop;
- converting verbally defined loops into actual C++ code.

### Scenario

In 1937, a mathematician named Lothar Collatz formulated an intriguing hypothesis that remains unsolved to this day (perhaps this would be a good challenge for you?) which can be described in the following way:

1. take any non-negative and non-zero integer number and name it  $c0$ ;
2. if it's even, evaluate a new  $c0$  as  $c0 / 2$
3. otherwise, if it's odd, evaluate a new  $c0$  as  $3 \cdot c0 + 1$
4. if  $c0 \neq 1$ , skip to point 2

The hypothesis says that, regardless of the initial value of  $c0$ , it will always (always!) go to 1.

Of course, it's an extremely complex task to use a computer in order to prove the hypothesis for any natural number (it may in fact need artificial intelligence), but you can use C++ to check some individual numbers. Maybe you can find the one that disproves the hypothesis and become a famous mathematician.

Okay, let's start. Write a program which reads one natural number and executes the above steps as long as  $c0$  remains different from 1. Moreover, we'll give you another task – we want you to count the steps needed to achieve the goal. Your code should output all intermediate values of  $c0$ , too – it'll be very illustrative, won't it?

Hint: the most important part of the problem is how to transform Collatz's idea into a "while" loop – this is the key to success.

Test your code using the data we've provided.

### Example input

15

### Example output

```
46
23
70
35
106
53
160
80
40
20
10
5
16
8
4
2
1
steps = 17
```

### Example input

16

### Example output

```
8
4
2
1
steps = 4
```

### Example input

```
1023
```

### Example output

```
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
```