

Lab 2.3.8 Do it yourself: factorials

Objectives

Familiarize the student with:

- classic iterative algorithms;
- and improve the student's skills in using loops.

Scenario

We want to stay a little longer in the land of mathematics – we hope you won't hold that against us.

We're going to show you a function that comes from one of the most interesting parts of mathematics: combinatorics. Its name is "factorial" and it was initially used to find a number of possible *permutations* of a set e.g. if you have three books (A,B,C) on your shelf, there are six possible arrangements of them: ABC, ACB, BCA, BAC, CAB, CBA. This means that there are six possible permutations of a three-element set. Therefore, we can say that the factorial of 3 is equal to 6.

The last sentence is often shortened to the following form:

$$3! = 6$$

One of the possible formal definitions of the factorial function has prompted us to formulate the next task for you, as it's a classic example of using a loop. Which one would be better? "While"? "For"? It's up to you!

Here goes the definition:

- $0! = 1$ (strange, but true)
- $1! = 1$ (obvious)
- $n! = 1 * 2 * \dots * n$ (can you see a loop here?)

We aren't going to waste your time any longer – let's start coding! Your program should ask the user for the value of n and then output the value of $n!$. We should warn you that factorial values grow extremely fast – don't be surprised. That's why our example input data doesn't look very impressive, in contrast to the output.

Example input

5

Example output

10

Example input

10

Example output

3628800

Example input

20

Example output

2432902008176640000