

Lab 3.4.6 Prime numbers – how do we find them?

Objectives

Familiarize the student with:

- classic notions and algorithms;
- and improve the student's skills in defining and using functions.

Scenario

A natural number is **prime** if it is greater than 1 and has no divisors other than 1 and itself.

Complicated? Not at all. Look: 8 isn't a prime number as you can divide it by 2 and 4 (we can't use divisors equal to 1 and 8 as the definition prohibits this). On the other hand, 7 is a prime number since we can't find any legal divisors for it.

Your task is to write a function that checks whether or not a number is prime or not.

The function:

- is called "isPrime";
- takes one int argument (the value to check);
- returns "true" if the argument is a prime number or "false" otherwise;
- should be **mute**.

Hint: try to divide the argument by all subsequent values (starting from 2) and check the remainder – if it's 0, your number cannot be a prime; think carefully when you should stop the process and check whether you could use the "sqrt()" function (which you already know).

Complete the code below.

Run your code and check whether your output is the same as ours.

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

using namespace std;

bool isPrime(int num) {

    // Insert your code here
}

int main(void) {
    for(int i = 0; i <= 21; i++)
        if(isPrime(i))
            cout << i << " ";
    cout << endl;
    return 0;
}
```

Example output

2 3 5 7 11 13 17 19