PiE *C++* Final Assignment

Name: Duc Nguyen,
student number: s1630512*

November 7, 2015

*Email address for correspondence: nguyenmanhduc@student.utwente.nl or ngmaduc@gmail.com

Exercise 1

Question 1.1

Write a *C++* program to compute the first N prime numbers, where N is given by the user. Use dynamic arrays to store the primes and use this information in the mod test.

Answer.

Three functions are used for this question including `bruteForce`, `modTest` and `print_primes` for finding prime numbers and printing out the result in console. Passing by reference are chosen to avoid unnecessary copy of variables.

```
std::vector <unsigned long int> bruteForce (int &n);
std::vector <unsigned long int> modTest (int &n);
void print_primes (int &n, const std::vector <unsigned long int>& primes);
```

The method chosen to create dynamic arrays to store the primes is `std::vector`. The range $[0, 2147483647]$ of unsigned long int fits to the scope of the question. The idea of `bruteForce`, `modTestDiv` are shown in the following code snippets.

`bruteForce`

```
        primes.push_back(2);
         unsigned long int c;//need to be the type of primes for mode test
         int count = 1;
         for (int count = 1; count < n; ){//counter from 1; "2" included before
             for (c = 2; c < num; c++){
                 if (num % c == 0) {//mod test from 2 to n
                     break;
                 }
             }
             if (c == num) {//to this point means no divisor up to n, Prime!
                 primes.push_back(num);//push to result vector of Prime
                 count++;//increase counter
             }
             num++;
         }
```

`modTestDiv`

```
        primes.push_back(2);
        for (int count = 1; count < n; ) {//counter from 1; "2" included ↩
            before
             bool isPrime = true;
             for (int i = 0; i < primes.size(); i++){
```

```
                if (num % primes[i] ==0) {//non-primes  are products of primes
                  isPrime = false;
                  break;
                }
            }
            if (isPrime == true) {
                primes.push_back(num);
                count++;//increase counter
            }
            num++;
        }
```

Question 1.2

Write to the screen a list of the first 10000 primes in the format below; where $p(n)$ is the n^{th} prime number. Report only the last five lines. Comment on the behaviour of the ratio $n * ln(p(n))/p(n)$ as n gets large.

Answer.

The void `print_ratio` (int &n, const `std::vector` <unsigned long int>& `primes`) and prime number finding functions together generate the required ratio. As n gets large, the ratio tends to converge to 1. Until $10^5 - th$ prime number, the ratio is 1.103.

The last five lines are and the `print_ratio` are listed below:

```
9996      :       104707  :       1.10348856177824
9997      :       104711  :       1.10356044403989
9998      :       104717  :       1.10361306655082
9999      :       104723  :       1.10366568381267
10000     :       104729  :       1.10371829582629
```

`print_ratio`

```
void print_ratio (int &n, const std::vector <unsigned long int>& primes){
    std::cout << "n\t:\t p(n)\t:\t n*ln( p(n) )/p(n) " << std::endl;
    for (int i = primes.size; i < primes.size(); i++) {
        std::cout << i+1 << "\t:\t" << primes[i] <<"\t:\t"
            <<std::fixed << std::setprecision(14) <<
                double((i+1)*log(primes[i])/primes[i])<< std::endl;
        }//set precision used for increase decimal displayed
}
```

Question 1.3

Based on question 2, give an estimate of the $10^6 - th$ prime number.

Answer.

We use 1.1 for the value of the ratio with $n = 10^6$:

$$10^6 * ln(p(10^6))/p(10^6) \approx 1.1$$

Using Wolfram Alpha to solve this equation, the estimate of the $10^6 - th$ prime number is:

$$p(10^6) \approx 15022800$$

Question 1.4

Instead of writing to the screen, write to a file (on disk) a list containing just the prime numbers. Print eight numbers per line, such that all numbers have the same space.

Answer.

Question 2

Use symbolic dierentiation from the matlab symbolic toolbox to derive the magnitude of the force as a function of r_{ij} .

ANSWER

Question 3

Plot the force fij as a function of the distance r_{ij}.

ANSWER

Question 4

Write two matlab functions that solve the differential equations, for a 1D system of particles, using Euler Forward algorithm and the Verlet algorithm. Comment on which algorithm you prefer and state why?

ANSWER

Question 5

Plot the potential V_{ij} as a function of the distance rij . Plot separately both the attractive and repulsive contribution.

ANSWER

Question 6

Plot the potential V_{ij} as a function of the distance rij . Plot separately both the attractive and repulsive contribution.

ANSWER