

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ**

федеральное государственное автономное образовательное  
учреждение высшего образования

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

Инженерная школа информационных технологий и робототехники  
Отделение информационных технологий  
Направление информатика и вычислительная техника

**Отчет**  
по лабораторной работе № 2

по дисциплине  
«ПРОГРАММИРОВАНИЕ»

Вариант № 26

**Работа с базовым контейнером vector на языке программирования C++.**

Выполнил:

Студент группы 8В32

\_\_\_\_\_

Д.О.Карташов

Проверил:

Ассистент ОИТ ИШИТР

\_\_\_\_\_

А.Ю.Малкин

Томск 2024

## Цель работы

Получить навыки работы с базовым контейнером `vector` на языке программирования C++.

## Задание

Собрать и отладить проект на языке программирования C++ с применением CMake в VS Code по реализации длинной арифметики на основе базового контейнера `vector` согласно варианту

## Ход работы

Сначала проверяем правильность работы программы на небольших значениях.

На рисунке 1.1 демонстрируется результат выполнения функции `factorial()` с аргументом 5. Полученное значение совпадает с ожидаемым, что подтверждает корректность работы функции.



```
307
308     std::cin >> input_1;
309     std::cout << factorial(input_1) << std::endl;
310
```

PROBLEMS 1 OUTPUT TERMINAL PORTS DEBUG CONSOLE

5  
120  
□

Рисунок 1.1 – Отладка вычисления факториала числа

На рисунке 1.2 демонстрируется результат выполнения функции `power()` с аргументами 2 и 10. Полученное значение совпадает с ожидаемым, что подтверждает корректность работы функции.

```
310
311     std::cin >> input_1;
312     std::cout << power(2, input_1) << std::endl;
```

PROBLEMS 1 OUTPUT TERMINAL PORTS DEBUG CONSOLE

5  
120  
10  
1024  
□

Рисунок 1.2 – Отладка вычисления степени двойки

На рисунке 1.3 демонстрируется результат вычисления суммы результатов исполнения функции `power()`, вызванной с аргументами 2 и 5, и с аргументами 2 и 6. Полученное значение совпадает с ожидаемым, что подтверждает корректность работы функции.

```
313
314     std::cin >> input_1 >> input_2;
315     std::cout << power(2, input_1) + power(2, input_2) << std::endl;
316
```

PROBLEMS 1 OUTPUT TERMINAL PORTS DEBUG CONSOLE

5  
120  
10  
1024  
5 6  
96  
□

Рисунок 1.3 – Отладка вычисления суммы степеней двоек

На рисунке 1.4 демонстрируется проверка введённых значений на корректность. Так как значения не прошли проверку, программа запросила ввод повторно, что соответствует ожидаемому поведению.

```
316
317     std::cin >> input_1 >> input_2;
318     while (input_1 < input_2)
319     {
320         std::cout << "The first number should be greater or equal than the second number!" << std::endl;
321         std::cin >> input_1 >> input_2;
322     }
```

PROBLEMS 1 OUTPUT TERMINAL PORTS DEBUG CONSOLE

```
5
120
10
1024
5 6
96
2 4
The first number should be greater or equal than the second number!
█
```

Рисунок 1.4 – Отладка проверки введенных данных на корректность

На рисунке 1.5 демонстрируется результат вычисления разности результатов исполнения функции `power()`, вызванной с аргументами 2 и 5, и с аргументами 2 и 2. Полученное значение совпадает с ожидаемым, что подтверждает корректность работы функции.

```
320     std::cout << "The first number should be greater or equal than t
321     std::cin >> input_1 >> input_2;
322 }
323     std::cout << power(2, input_1) - power(2, input_2) << std::endl;
324
```

PROBLEMS 1 OUTPUT TERMINAL PORTS DEBUG CONSOLE

```
5
120
10
1024
5 6
96
2 4
The first number should be greater or equal than the second number!
5 2
28
█
```

Рисунок 1.5 – Отладка вычисления суммы степеней двоек

На рисунке 1.6 демонстрируется результат выполнения функции `fibonacci()` с аргументом 8. Полученное значение совпадает с ожидаемым, что подтверждает корректность работы функции.



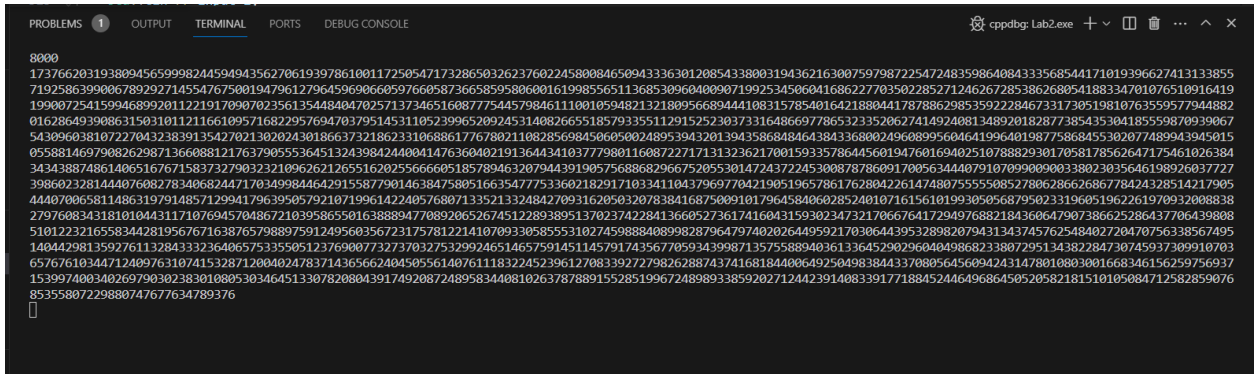


Рисунок 2.2 – Отладка вычисления степени двойки

На рисунке 2.3 демонстрируется результат вычисления суммы результатов исполнения функции `power()`, вызванной с аргументами 2 и 1337, и с аргументами 2 и 9999. Полученное значение совпадает с ожидаемым, что подтверждает корректность работы функции.

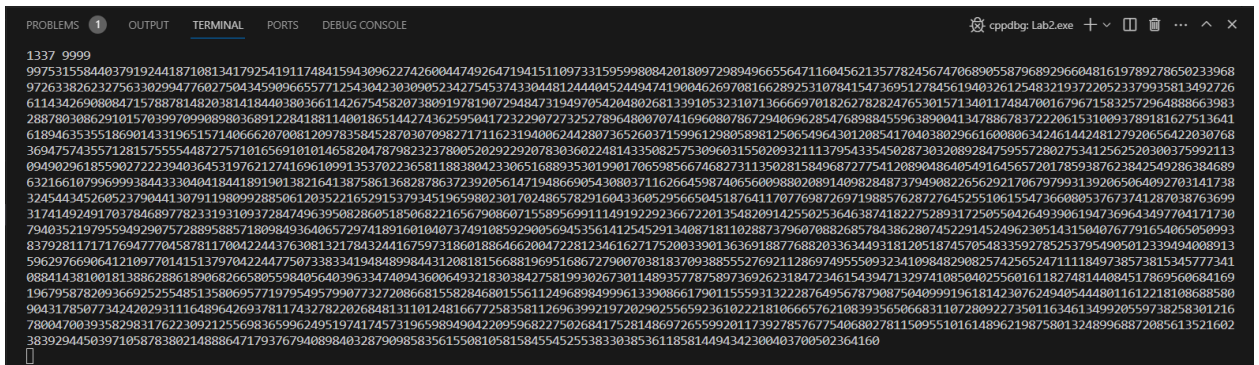


Рисунок 2.3 – Отладка вычисления суммы степеней двоек

На рисунке 2.4 демонстрируется результат вычисления разности результатов исполнения функции `power()`, вызванной с аргументами 2 и 10000, и с аргументами 2 и 1000. Полученное значение совпадает с ожидаемым, что подтверждает корректность работы функции.

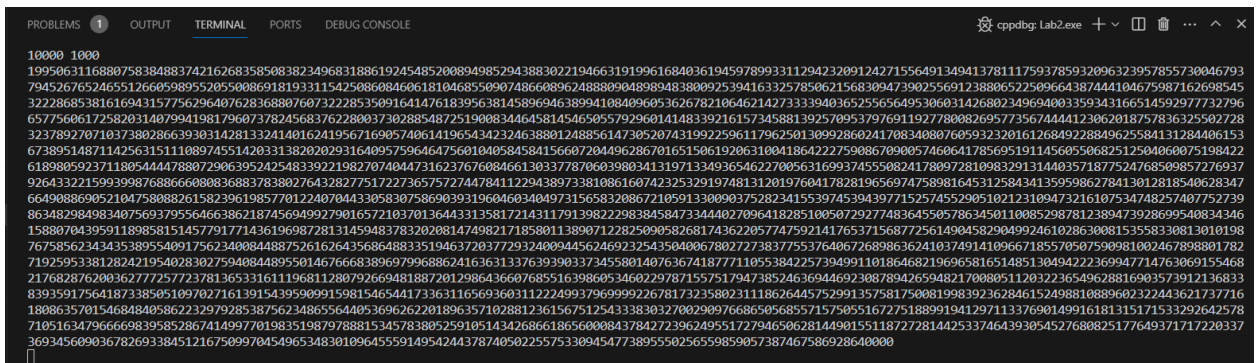
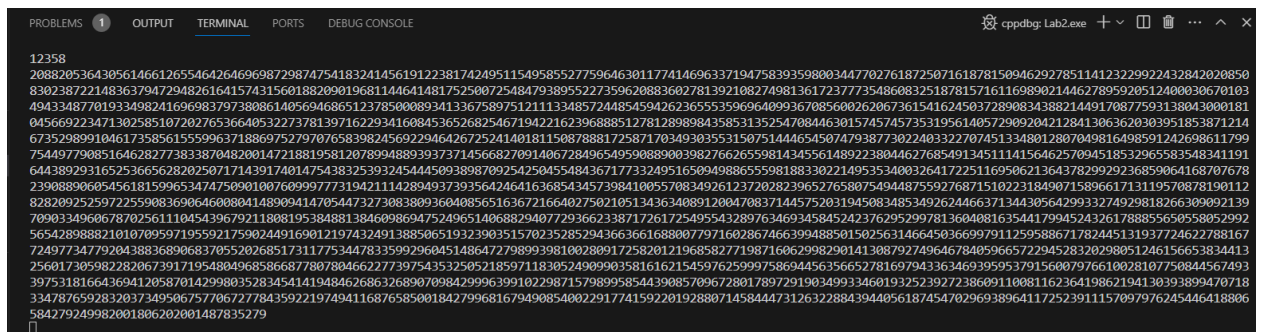


Рисунок 2.4 – Отладка вычисления суммы степеней двоек

На рисунке 2.5 демонстрируется результат выполнения функции `fibonacci()` с аргументом 12358. Полученное значение совпадает с ожидаемым, что подтверждает корректность работы функции.



```
PROBLEMS 1 OUTPUT TERMINAL PORTS DEBUG CONSOLE
cppdbg: Lab2.exe + - - - - -
12358
20882053643056146612655464264696987298747541832414561912238174249511549585527759646301177414696337194758393598003447702761872507161878159946292785114123229922432842020850
83023872214836379472948261641574315601882090196811446414817525007254847938955227359620883602781392108274081361723777354860832518781571611698902144627895920512400030670103
494334877019334982416969837973886140569468651237850008934133675897512113348572448545842623655535969640993670856002620673615416245037289083438821449170877593138043000181
045669224713025851072027653664053227378139716229241608453652682546710422162396888512781289898435853135254708463015745745735319561405729092042128413063620303951853871214
67352989104617258561555926371886975270767583982456922946426725241401811568788172587170349303553158751444654507470387730224033227074513348012867049816498591242608611799
7544977008516462827738338704820014721881958120789048803937214566827091406728406549590889003982766265598143455614892238044627685401345111415646257094518532965583548341191
644389203165253655282025071714391740147543832539324544450238987002542504548436717732495165994088555981883302214953534003264172251169506213643782992236859064168707678
2390880695456181509653474750001007609997773194211142894037393564246416368543457398410055708349261237202823965276580754944875592768715102231849071589661713119570878190112
82820025259722559083690646008041489004147054473273083809360408565163672166402750210513436348891200470837144575203104508348534026244663713443056429033274929818266309992139
70903349606787025611104543967921180819538488138460986047524965140688204077293662338717261725409543289763469245845242376295299781360408163544179945243261788855659558052902
56542898892101070950719559217590244016901219743249138850651932390351570235285204366366168800779716028674663904885015025631466450366097911259588671782445131937724622788167
72407724779204388368906837055202685173117753447833599206045148647279899398100280017258201219685827710871606299829014138879274964678405966577294528320298051246156653834413
256017305982282067391710548049685866877807804662277397543532505218507118305249099035816162154976259097586044563566527816979433634603959537915600797661002810775084567493
39753181664360412058701420980357834541410484676863268007098429906390102987175089585443908570967280178972919034993346010325392723860911008116236419862194130393899470718
33478765928320373495067577067277843592219749411687658500184279968167949085400229177415922019288071458444731263228843944056187454702969389641172523911157097976245446418806
584279249982001806202001487835279
```

Рисунок 2.5 – Отладка вычисления числа Фибоначчи

## Вывод

Получены навыки работы с базовым контейнером `vector` на языке программирования C++.

Ссылка на исполняемый код: [\*https://github.com/The-Aozzi/Lab2\*](https://github.com/The-Aozzi/Lab2)