

**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 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 – Отладка вычисления факториала числа

На рисунке 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 2 – Отладка вычисления степени двойки

На рисунке 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 3 – Отладка вычисления суммы степеней двоек

На рисунке 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 4 – Отладка проверки введённых данных на корректность

На рисунке 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 5 – Отладка вычисления суммы степеней двоек

На рисунке 6 демонстрируется результат выполнения функции `fibonacci()` с аргументом 8. Полученное значение совпадает с ожидаемым, что подтверждает корректность работы функции.


```
PROBLEMS 1 OUTPUT TERMINAL PORTS DEBUG CONSOLE
cppdbg: Lab2.exe + - - - ^ x

8000
17376620319380945659982445949435627061939786100117250547173286503262760224580084650943336301208543380031943621630075979872254724835986408433568544171019396627413133855
179258639006789292714554767500194796127964596906605976608736658959806016199856511368309604009071992534506041686227703502285271246267285386268054188334701076510916419
19900725415994689920112219170907023561354840470251737346516087754457984611100105948213180956689444108315785401642188944178788629853592228467331730519810763559577944882
01628649390863150310112116610957168229576947037951453110523996520924531408266551857933551129152530733164866977865323352667414924081348920182877385435304185559870939067
54309603810722704323839135427021302024301866373218623310688617767802110828569845060500248953943201394358684846438433680024960899560464199640198775868455302077489943945015
055881469790826298713608812176379055536451324398424400414763604021913644341037798011608722717131323621700159335786445601947601694025107888293017058178562647175461026384
34343887486140651676715837327903232109626212655162025566660518578946320794439190575688682966752055301472437224530087878609170056344407910709909090338023035646198926037727
3986023281444076082783406824471703499844642915587790146384758051663547775336021829171033411043796977042190519657861762804226147480755555085780628662686778424328514217905
44407006581148631979148571299417963950579210719961422405768071335213324842709316205032078384168750091017964584066285240107161561019930505687950233196051962261970932008838
2797608343181010443117107694570486721039585501638889477089206526745122893895137023742284136605273617416043159302347321706676417294976882184360647907386625286437706439808
5101223216558344281956761638765798897591249560356723175781221410709330585551027459888408982879647974020264495921703064439532898207943134745762548402720470756338567495
14044298135927611328433236406575355051237690077327303275329924651465759145114579174356770593439987135755889403613364529029604049868233807295134382284730745937309910703
65767163044712409763107415328712004024783714365662404505561407611183224523961270833927279826288747341681844006492504983844337080564560942431478010803001668364156259756937
153997400340269790302383010853034645133078208043917492087248958344081026378788915528519967248989338592027124423914083391771884524464968645052058218151010584712582859076
85355807229880747677634789376

```

Рисунок 8 – Значение 2^{8000}

На рисунке 9 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.

```
PROBLEMS 1 OUTPUT TERMINAL PORTS DEBUG CONSOLE
cppdbg: Lab2.exe + - - - ^ x

1337 9999
99753155844037919244187108134179254191174841594309622742600447492647194151109733159599808420180972989496655647116045621357782456747068095587968929660481619789278650233968
972633826273256330299477602758434590966557712543042303090523427545374330481244404524494719804626970816628925310784154736951278456194032612548372193722052379935813492726
6114426980884715788781482038141844038036611426754582073809197819072948473194970542048026813391053231071366669701826278282476530157134011748470016796715832572964888663983
28878038062910157039970990898036891228418811480186514427436259504172322907273252789648007074169608807872940696285476898845596389004134788678372220615310093789181627513641
189463535518690143139651571406662070081209783584528703070982717116231940062442807365260371599612980589812506549643012085417040380296616080063424614424817292065422030768
369475743557128157555544872751016569101014658204787982323780052029292078303602248143350825730960331550209321113795433545028730320892874595572802753412562520300375992113
049029618559027222394036451976212741696109913537022365811883804233065168893501990170659856674682731135028158496872775412089048640549164565720178593876723842549286384689
6321661079969938443304041844189190138216413875861368287863723920561471948663950436083711626459874065600980208914098284873794908226562921706797931392065064092703141738
324544352652573904413079199809288506120352216529153793451965980230170248657829160433605295665045187641170776987269719885762872764525510615547366080537673741287038763699
3174149249170378468977823319130937284749639508286051850682216567908607155895699111491922923672201354820914255025364638741822752893172505504264939061947369643497704171730
7940352197959492907572889588571809849364065729741891601040737491085929005694535614125452913408718110288737960708826857843862807452291452496230514315040077791654065050993
8379281171717694770458711700422443763081321784324411675973186018864662004722812346162717520033901363691887768820336344931812051874570548335927852537954905012339494008913
5962976690641210977014513797042244775073383419484898443120818156688196951686727900703818370938855527692112869749555093241098482908254265524711184973857381534577341
088414381001813886288618906826658059840564039633470943600649321830384275819930267301148935778758973692623184723461543947132974108504025609161182748144084517869560684169
1967958782093669255458315806957179795495799077327086681558284680155611249689849996133908661790115559313222876495678790875040999196184243076249405444801161221810868860
90431785077018373802866393031428133241401624195671690574061496543232463880124885614730520743199225961179625013099286024170834080760593232016126849228849625584131284406153
7800470039358298317623092125569836599624951947174573196598949042209596822750268417578148697265599201173927857677540680728175095510161489621987580132489968872085613521602
383929445039710587838021488864717937679408984032879098583561550810581584554525538330385361185814494342300403700502364160

```

Рисунок 9 – Значение $2^{1337} + 2^{9999}$

На рисунке 10 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.

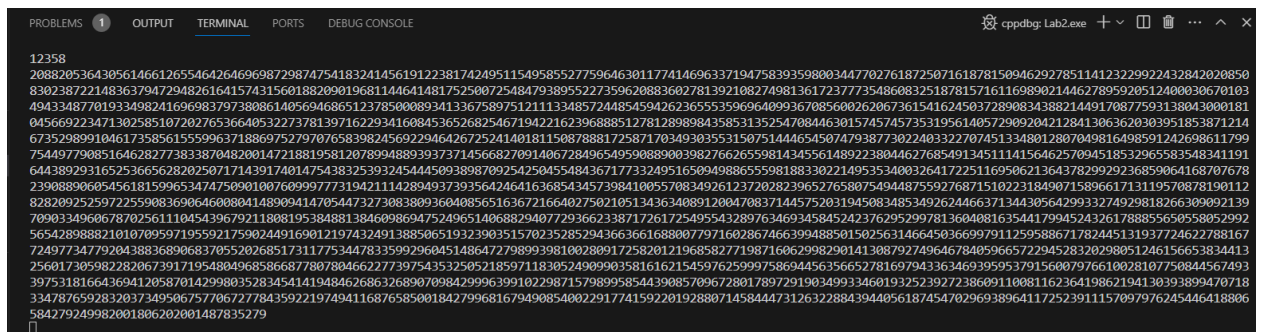
```
PROBLEMS 1 OUTPUT TERMINAL PORTS DEBUG CONSOLE
cppdbg: Lab2.exe + - - - ^ x

10000 1000
19950631168807583848837421626835850838234968318861924548520089498529438830221946631919961684036194597899331129423209124271556491349413781117593785932096323957855730046793
79452676524655126605989552055008691819331154250860846061810468550907486608962488809048989483800925394163325785062156830947390255691238806522509664387444104675987162698545
3222868538161694315775629640762836880760732285350916417618395638145896946389941084096053626782106462142733394036525565649530603142680234969400335934166514592977732796
657756061725820314079941981796073782456837622800373028854872519008344645814546505579296014148339216157345881392570953797691192778008269577356744441230620187583625502728
323789270710373802866393031428133241401624195671690574061496543232463880124885614730520743199225961179625013099286024170834080760593232016126849228849625584131284406153
6738951487114256541110897455142033138202029316409575964647560104058458415660720449628670165150619206310041864222759086709005746064178560519114560550682512504060075198422
6189086905210475808826158239619857701224070443305830758690391960460340049731565832086721059133009037528234155397453943977152574552905182131094732161075347482577752739
8634829849307569379556466386218745649929790165721037013644313581721431179139822298384584734440270964182851005072927748364550578634501100852987812389473928699540834346
15880704399118985815457791771436196872813145948378320208147498217185801138907122825905826817436220577475921417653715687725614904582904992461028630081535833081301098
7675856234343538955409175623400408875261626435686488335194637203772932400933452469303254350400671807771105384225739499110186468219696581261485130494222369947714763069155468
719259533812842419540282037675940848895501467666838969799688624163631337639903736469374139771105384225739499110186468219696581261485130494222369947714763069155468
21768287620036777275277381365331611196811280792669481887281298643660768551639860536022978175571794738524636944692308789426594821700808511203236549628816903573912136833
89359175641873385051097027171436196872813145948378320208147498217185801138907122825905826817436220577475921417653715687725614904582904992461028630081535833081301098
7675856234343538955409175623400408875261626435686488335194637203772932400933452469303254350400671807771105384225739499110186468219696581261485130494222369947714763069155468
719259533812842419540282037675940848895501467666838969799688624163631337639903736469374139771105384225739499110186468219696581261485130494222369947714763069155468
21768287620036777275277381365331611196811280792669481887281298643660768551639860536022978175571794738524636944692308789426594821700808511203236549628816903573912136833
89359175641873385051097027171436196872813145948378320208147498217185801138907122825905826817436220577475921417653715687725614904582904992461028630081535833081301098
7675856234343538955409175623400408875261626435686488335194637203772932400933452469303254350400671807771105384225739499110186468219696581261485130494222369947714763069155468
719259533812842419540282037675940848895501467666838969799688624163631337639903736469374139771105384225739499110186468219696581261485130494222369947714763069155468
3693456090367826933845121675099704549653483010964555914954244378740502255753309454773895550256559859657387467586928640000

```

Рисунок 10 – Значение $2^{10000} + 2^{1000}$

На рисунке 11 демонстрируется результат выполнения функции `fibonacci()` с аргументом 12358. Полученное значение совпадает с ожидаемым, что подтверждает корректность работы функции.



The screenshot shows a C++ IDE with a terminal window. The terminal displays the output of the `fibonacci` function for the input 12358. The output is a large number: 208820536430561466126554642646969872987475418324145619122381742495115495855277596463011774146963371947583935980034477027618725071618781599462927851141232299224328420208508302387221483637947294826164157431560188209019681144641481752500725484793895522735962088360278139210827498136172377735486083251878157161169890214462789592051240003067010349433487701933498241696983797388861405694686512378500089341336758975121132485724485458426236555359696409936708560026206736154162450372890834388214491708775931380430001810456692247130258510720276536640532273781397162292416084536526825467104221623968885127812898984358531352547084630157457457353195614057290920421284130636203039518538712146735298910461725856155592637188697527070765839824569229464267252414018115687881725871703493035531587514446545974703877302240332707451334801286704981649859124260861179975449770085164628277383387048200147218819581207890488039372145668270914067284065495908890039827662655981434556148922380446276854013451114156462570945185329655835483411916443892031652536552820250717143917401475438325393245444502389870025425045484367177324951659940885559818833022149535340032641722511695062136437829292236859064168707678239088906954561815096534747509010076999777319421114289493739356424641636854345739841005570834926123720282396527658075494487559276871510223184907158966171311957087819011282800252597225590836906460080414890941470544732730838093604085651636721664027502210513436348891200470837144575203104508348534926244663713443056429933274929818266309992139709033496067870256111045439679211808195384881384609860475249651406882040772936623387172617254955432897634692458452423762952997813604081635441799452432617888556595580529925654289889210107095071955921759024401690121974324913885065193239035157023528520436636616880077971602867466399488501502563146645036609791125958867178244513193772462278816772407724779204388368906837055202685173117753447833599206045148647279899398100280017258201219685827710871606299829014138879274964678405966577294528320298051246156653834413256017305982282067391710548049685866877807804662277397543532505218507118305249099035816162154976259097586044563566527816979433634603959537915600797661002810775084567493397531816643604120587014209803578345414194846768632680070984299063901022987157089958544390857096728017897291903499334601032539272386091100811623641986219413039389947071833478765928320373495067577067277843592219749411687658500184279968167949085400229177415922019288071458444731263228843944056187454702969389641172523911157097976245446418806584279249982001806202001487835279

Рисунок 11 – Значение 12358-ого числа Фибоначчи

Вывод

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

Приложение 1 – Код программы `main.cpp`

Ссылка на исполняемый код: *<https://github.com/The-Aozzi/Lab2>*