| Gerb-BMSTU_01 | **Министерство образования и науки Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |
| --- | --- |

ФАКУЛЬТЕТ Робототехники и комплексной автоматизации

КАФЕДРА Системы автоматизированного проектирования (РК-6)

**ОТЧЕТ О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ**

Студент Гусаров Аркадий Андреевич

Группа РК6-53Б

Тип задания Лабораторная работа №4

Тема лабораторной работы Разбиения целых чисел

Студент **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Гусаров А.А.**

*подпись, дата фамилия, и.о.*

Преподаватель **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Волосатова Т.М**

*подпись, дата фамилия, и.о.*

Преподаватель **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Родионов С.В**

*подпись, дата фамилия, и.о.*

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Москва, 2021 г.*

**Задание на лабораторную работу**

Перечислить все разбиения заданного целого числа n > 0 на слагаемые, количество которых равно заданной величине m. Для генерации разбиений следует применить алгоритм Гинденбурга, а слагаемые каждого разбиения должны быть записаны в порядке не убывания своих величин слева направо и разделены знаком +.

**Цель выполнения лабораторной работы**

**Цель** – ознакомление и применение на практике алгоритма Гинденбурга для разбиения целых чисел.

**Теоретическая часть**

Алгоритм Гинденбурга порождает разбиения любых натуральных чисел в порядке увеличения количества слагаемых, а разбиения равной длины перечисляются в лексиграфическом порядке. В общем случае шаги алгоритма Гинденбурга начинаются с разбиения, которое состоит только из одного слагаемого, равного **n (p1=n)**. У последующих разбиений число слагаемых m не убывает, а сами слагаемые записываются в неубывающем порядке своих величин. При этом каждое очередное разбиение на **m** слагаемых строится по текущему разбиению следующим образом. Нужно просмотреть текущее разбиение справа налево, с целью найти наибольшее слагаемое **pi**, которое отличается от последнего слагаемого p\_m не меньше, чем на **2**: **pm − pi ≥ 2.**

Слагаемое **pi** нужно увеличить на **1** и присвоить его новое значение всем слагаемым **pj**, справа от него до предпоследнего слагаемого, включительно. Указанное преобразование текущего разбиения можно формально записать следующим образом:

**pj = pi + 1 для всех j = i, …, (m − 1).**

С учетом полученных значений последнее слагаемое **pm** вычисляется по следующей остаточной формуле:

**pm = n − (p1 + … + pj + … + pm−1)**.

**Пример алгоритма**

В следующем примере приведены все последовательные разбиения числа **7** на **4** слагаемых, которые получены таким способом, а образующее слагаемое **1** выделено подчеркиванием:

**1 + 1 + 1 + 4 −> 1 + 1 + 2 + 3 −> 1 + 2 + 2 + 2 .**

**Код программы**

*Файл main.cc:*

#include <iostream>

#include <vector>

using namespace std;

void printVector(vector<int> vec)

{

for (int i = 0; i < vec.size() - 1; i++)

cout << vec[i] << " + ";

cout << vec[vec.size() - 1] << endl;

}

vector<int> startVectorGenerator(int n, int m)

{

vector<int> arr;

arr.resize(m);

for (int i = 0; i < m; i++)

arr[i] = 1;

arr[arr.size() - 1] = n - (m - 1);

return arr;

}

int Gitenburg(vector<int> vec)

{

int elIdx = vec.size() - 2;

while (true)

{

if (elIdx == -1)

elIdx = vec.size() - 2;

vec[vec.size() - 1] -= 1;

vec[elIdx] += 1;

elIdx -= 1;

if (vec[vec.size() - 1] < vec[vec.size() - 2])

return 0;

printVector(vec);

}

return 0;

}

int main(int argc, const char \*argv[])

{

int n, m;

cout << "Введите целое число больше 0: ";

cin >> n;

cout << "Введите количество слагаемых: ";

cin >> m;

if ((n \* m == 0) || (n < 0) || (m < 0) || (m > n))

{

cout << "Ошибка ввода" << endl;

return 1;

}

if (m == 1)

{

cout << n << endl;

return 0;

}

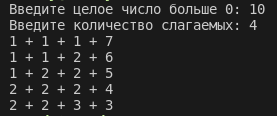
vector<int> vec = startVectorGenerator(n, m);

printVector(vec);

return Gitenburg(vec);

}

**Результат работы программы**



**Список использованных источников**

1. *Волосатова Т.М. курс лекций по дисциплине «Методы комбинаторных вычислений».*