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 $\mathbf{n\ (p_1=n)}$. 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 $\mathbf{p_i}$, которое отличается от последнего слагаемого p_m не меньше, чем на **2**: $\mathbf{p_m - p_i \geq 2.}$

Слагаемое $\mathbf{p_i}$ нужно увеличить на **1** и присвоить его новое значение всем слагаемым $\mathbf{p_j}$, справа от него до предпоследнего слагаемого, включительно. Указанное преобразование текущего разбиения можно формально записать следующим образом:

$$\mathbf{p_j = p_i + 1 \text{ для всех } j = i, \ldots, (m - 1).}$$

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

$$p_m = n - (p_1 + \ldots + p_j + \ldots + p_{m-1}).$$

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 + <u>1</u> + 4 –> 1 + <u>1</u> + 2 + 3 –> 1 + 2 + 2 + 2 .

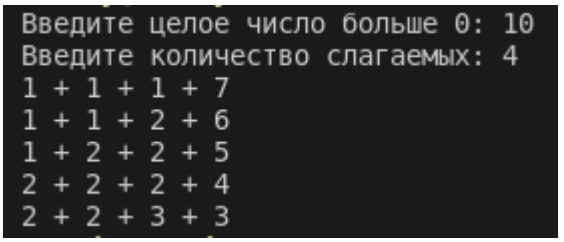
Код программы

<u>Файл main.cc:</u>

```
#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».*