|  |  |
| --- | --- |
| 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 №2

Тема лабораторной работы Факториальный алгоритм левого дополнения с инкрементом

Студент **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_Гусаров А.А.\_**

*подпись, дата фамилия, и.о.*

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

*Москва, 2021 г.*

# Задание на лабораторную работу

Перечислить все перестановки первых 5-ти натуральных чисел, используя факториальный алгоритм левого дополнения с инкрементом.

# Цель выполнения лабораторной работы

**Цель** – ознакомление и применение на практике факториального метода получения перестановок.

# Теоретическая часть. Описание алгоритма

Наиболее простые алгоритмы перестановок основаны на рекурсивном определении факториала.

Алгоритм получает перестановки n чисел из перестановки n-1 чисел.

В факториальном алгоритме левого дополнения с инкрементом каждую перестановку (n-1) натуральных чисел нужно последовательно сопоставить со всеми различными целыми значениями m от 1 до n. При этом необходимо увеличить на 1 значения всех элементов pj, которые не меньше очередного значения параметра m:

pj <- pj + 1; pj ≥ m=1, … n.

Затем следует дополнить модифицированную перестановку текущим значением

параметра m. В итоге из каждой перестановки (n-1) элементов получается n перестановок целых чисел от 1 до n.

# Пример алгоритма

Минимальная начальная перестановка состоит из одного числа {1}.

**Последовательность действий:**

|  |  |  |
| --- | --- | --- |
| *№* | *Начальная перестановка* | *Конечная перестановка* |
| 1 | {1}, m = 1 | {1,1+1} = {1, 2} |
|  | {1}, m = 2 | {2,1} |
| 2.1 | {2, 1}, m = 1 | {1, 2+1, 1+1} = {1, 3, 1} |
|  | {2, 1}, m = 2 | {2, 2+1, 1+1} = {2, 3, 1} |
|  | {2, 1}, m = 3 | {1, 2, 3} |
| 2.2 | {1, 2}, m = 1 | {1, 1+1, 2+1} = {1, 2, 3} |
|  | {1, 2}, m = 2 | {2, 1, 2+1} = {2, 1, 3} |
|  | {1, 2}, m = 3 | {3, 1, 2} |
| … | … | … |

Далее действия выполняются рекурсивно, что в конечном итоге позволит получить все перестановки первых 5-ти натуральных чисел.

# Код программы

#include <iostream>

#include <vector>

void print\_vect(std::vector<int> &arr) {

for (auto &i: arr)

std::cout << i << " ";

std::cout << std::endl;

}

void FactorialLeftAddition(std::vector<int> &arr, int &globalCounter, int maxNumber) {

int mPosition = 0;

auto iter = arr.cbegin();

arr.insert(iter, 0);

std::vector<int> arrayCopy = arr;

int m = 1;

for ( ; m != arr.size() + 1; m++) {

arr = arrayCopy;

arr[mPosition] = m;

for (int j = 1; j != arr.size() + 1; j++) {

if (arr[j] >= m)

arr[j]++;

}

if (arr.size() == maxNumber) {

std::cout << globalCounter++ << ") ";

print\_vect(arr);

} else if (m != arr.size() + 1)

FactorialLeftAddition(arr, globalCounter, maxNumber);

else

return;

}

}

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

int globalCounter = 1;

int arrayLength = 5;

std::vector<int> array;

array.push\_back(1);

FactorialLeftAddition(array, globalCounter, arrayLength);

return 0;

}

# Результат работы программы

|  |  |  |
| --- | --- | --- |
| 1) 1 2 3 4 5  2) 2 1 3 4 5  3) 3 1 2 4 5  4) 4 1 2 3 5  5) 5 1 2 3 4  6) 1 3 2 4 5  7) 2 3 1 4 5  8) 3 2 1 4 5  9) 4 2 1 3 5  10) 5 2 1 3 4  11) 1 4 2 3 5  12) 2 4 1 3 5  13) 3 4 1 2 5  14) 4 3 1 2 5  15) 5 3 1 2 4  16) 1 5 2 3 4  17) 2 5 1 3 4  18) 3 5 1 2 4  19) 4 5 1 2 3  20) 5 4 1 2 3  21) 1 2 4 3 5  22) 2 1 4 3 5  23) 3 1 4 2 5  24) 4 1 3 2 5  25) 5 1 3 2 4  26) 1 3 4 2 5  27) 2 3 4 1 5  28) 3 2 4 1 5  29) 4 2 3 1 5  30) 5 2 3 1 4  31) 1 4 3 2 5  32) 2 4 3 1 5  33) 3 4 2 1 5  34) 4 3 2 1 5  35) 5 3 2 1 4  36) 1 5 3 2 4  37) 2 5 3 1 4  38) 3 5 2 1 4  39) 4 5 2 1 3  40) 5 4 2 1 3 | 41) 1 2 5 3 4  42) 2 1 5 3 4  43) 3 1 5 2 4  44) 4 1 5 2 3  45) 5 1 4 2 3  46) 1 3 5 2 4  47) 2 3 5 1 4  48) 3 2 5 1 4  49) 4 2 5 1 3  50) 5 2 4 1 3  51) 1 4 5 2 3  52) 2 4 5 1 3  53) 3 4 5 1 2  54) 4 3 5 1 2  55) 5 3 4 1 2  56) 1 5 4 2 3  57) 2 5 4 1 3  58) 3 5 4 1 2  59) 4 5 3 1 2  60) 5 4 3 1 2  61) 1 2 3 5 4  62) 2 1 3 5 4  63) 3 1 2 5 4  64) 4 1 2 5 3  65) 5 1 2 4 3  66) 1 3 2 5 4  67) 2 3 1 5 4  68) 3 2 1 5 4  69) 4 2 1 5 3  70) 5 2 1 4 3  71) 1 4 2 5 3  72) 2 4 1 5 3  73) 3 4 1 5 2  74) 4 3 1 5 2  75) 5 3 1 4 2  76) 1 5 2 4 3  77) 2 5 1 4 3  78) 3 5 1 4 2  79) 4 5 1 3 2  80) 5 4 1 3 2 | 81) 1 2 4 5 3  82) 2 1 4 5 3  83) 3 1 4 5 2  84) 4 1 3 5 2  85) 5 1 3 4 2  86) 1 3 4 5 2  87) 2 3 4 5 1  88) 3 2 4 5 1  89) 4 2 3 5 1  90) 5 2 3 4 1  91) 1 4 3 5 2  92) 2 4 3 5 1  93) 3 4 2 5 1  94) 4 3 2 5 1  95) 5 3 2 4 1  96) 1 5 3 4 2  97) 2 5 3 4 1  98) 3 5 2 4 1  99) 4 5 2 3 1  100) 5 4 2 3 1  101) 1 2 5 4 3  102) 2 1 5 4 3  103) 3 1 5 4 2  104) 4 1 5 3 2  105) 5 1 4 3 2  106) 1 3 5 4 2  107) 2 3 5 4 1  108) 3 2 5 4 1  109) 4 2 5 3 1  110) 5 2 4 3 1  111) 1 4 5 3 2  112) 2 4 5 3 1  113) 3 4 5 2 1  114) 4 3 5 2 1  115) 5 3 4 2 1  116) 1 5 4 3 2  117) 2 5 4 3 1  118) 3 5 4 2 1  119) 4 5 3 2 1  120) 5 4 3 2 1 |

# Список использованных источников

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