***Министерство образования Республики Беларусь***

***Учреждение образования***

***«Брестский государственный технический университет»***

***Кафедра ИИТ***

**Лабораторная работа №5**

**По дисциплине Дискретная математика за II семестр**

**Тема: «Комбинаторика»**

**Выполнил:**

Студент группы ИИ-15 (1)

1-го курса

Волк И. А.

**Проверил:**

Глущенко Т.А.

Брест 2018

Цель: познакомиться с комбинаторикой.

**Задание:**

1. Написать программу генерации перестановок (рекурсивный вариант). Четные номера – реализуют лексикографический порядок генерации. Нечетные номера – реализуют антилексикографический порядок генерации.

Код программы:

#include "stdafx.h"

#include <Windows.h>

#include <iostream>

#include <cmath>

using namespace std;

int showPermutations(int\*, int, int, bool);

template<typename T>

void swap(T\*, T\*);

int main()

{

const int sizeOfArr = 4;

int arr[sizeOfArr];

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

arr[i] = i + 1;

int numOfPermutations = showPermutations(arr, sizeOfArr, 0, 0);

cout << "Num of permutations: " << numOfPermutations << endl;

system("pause");

return 0;

}

int showPermutations(int\* arr, int size, int level, bool std)

{

int numOfPermutations = 0;

for (int i = level, j = level; j < size; j++)

{

swap(&arr[i], &arr[j]);

if (!std||i!=j)

{

for (int x = 0; x < size; x++)

cout << arr[x] << " ";

cout << endl;

numOfPermutations++;

}

if(level!=size-2)

numOfPermutations += showPermutations(arr, size, level + 1, true);

swap(&arr[i], &arr[j]);

}

return numOfPermutations;

}

template<typename T>

void swap(T\* a, T\* b)

{

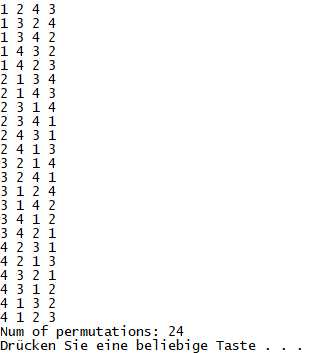
T var = \*a;

\*a = \*b;

\*b = var;

}

Output:



1. Написать программу генерации *k*- элементных подмножеств *n*-элементного множества в лексикографическом порядке. Вход: *k* – мощность подмножества, *n* - мощность множества.

Код программы:

#include "stdafx.h"

#include <Windows.h>

#include <iostream>

#include <cmath>

using namespace std;

void subsets(int, int);

int main()

{

int n = 9, k = 8;

subsets(n, k);

system("pause");

return 0;

}

void subsets(int n, int k)

{

int shift = 1;

if (n < k)

{

cout << "Error!" << endl;

return;

}

int \* marks = new int[k];

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

marks[i] = i;

int \* arr = new int[n];

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

arr[i] = i + shift;

int mark = k - 1;

while (1)

{

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

cout << arr[marks[i]] << " ";

cout << endl;

bool doInc = true;

if (marks[mark] == (n - k + mark))

{

doInc = false;

while (marks[mark] == (n - k + mark))

{

mark--;

if (mark == -1)

return;

}

for (int i = mark, j = ++marks[mark]; i < k; i++, j++)

marks[i] = j;

mark = k - 1;

while (marks[mark-1] == (n - k + mark))

{

mark--;

if (mark == -1)

return;

}

}

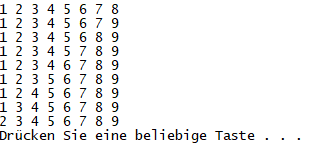
if(doInc)

marks[mark]++;

}

}

Output:



Вывод: познакомился с комбинаторикой.