Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №4

за 3 семестр

По дисциплине: «Дискретная математика»

Тема: «Комбинаторика»

Выполнил:

Студент 2 курса

Группы ПО-6(1)

Мартынович Д. М.

Проверил:

Глущенко Т. А.

Брест 2021

Цель: изучить теорию по теме «Комбинаторика», ознакомиться с понятиями, данными в теории. По данному варианту программно реализовать предложенные задачи.

***Задание.***

1. Номер вашего варианта равен остатку от деления вашего порядкового номера по списку на 12. Если ваш порядковый номер кратен 12, выполняете 12 вариант.
2. По пункту 4: решить рекуррентное соотношение и программно реализовать вычисление 5 члена числовой последовательности через рекурсию и полученное решение.
3. При разложении степени  коэффициенты при произведениях , , рассчитываются по формуле:  
    (число перестановок с повторениями) и носят название полиномиальных или мультиномиальных коэффициентов.  
   Например, вычислим коэффициент при произведении  в разложении  
   .  
   Он равен: .
4. Написать программу генерации перестановок (рекурсивный вариант). Четные номера – реализуют лексикографический порядок генерации. Нечетные номера – реализуют антилексикографический порядок генерации.
5. Написать программу генерации k- элементных подмножеств n-элементного множества в лексикографическом порядке. Вход: k – мощность подмножества, n - мощность множества.

Вариант 10

1. Сколькими способами можно выбрать две книги по разным темам, когда на полке находятся *15* книг по информатике, *12* книг по математике и *10* книг по химии?
2. Комитет из *20* членов избирает председателя и секретаря. Сколькими способами это можно сделать?
3. Имеются семь различимых шаров, и требуется положить три шара в первую коробку, два шара — во вторую и два шара — в третью коробку. Сколькими способами можно это сделать?
4. Разложить .
5. Решить линейное однородное рекуррентное соотношение *2-го* порядка:  
   

Решение:

№1

15\*12 = 180 - информатика и математика

12\*10 = 120 - математика и химия

15\*10 = 150 - информатика и химия

Всего вариантов: 180+120+150 = 450

Ответ: 450 способами.

№2

Первого члена, неважно, председателя или секретаря, можно выбрать 20 способами. Ну а второго, то его можно выбрать уже только 19 способами, так как среди вакансий одна уже занята. Получаем 20 \* 19 = 380 способов.

Ответ: 380 способами.

№3

7 \* 6 \* 5 + 4 \* 3 + 2 \* 1 = 224

Ответ: 224 различными способами.

4 Разложить (a + b)7:

(a + b)7 = a7 + 7a6b + 21a5 b2 + 35a4 b3 + 35a3 b4 + 21a2 b5 + 7ab6 + b7

5. Решить линейное однородное рекуррентное соотношение 2-го порядка:

a0 = 1.

a1 = -8.

an = -4\*an-1 -4\*an-2, a ≥ 2.

pn=- 4pn-1-4pn-2

p2+4p+4=0

p1,2=-2

Коэффициенты найдем, исходя из начальных условий:

Отсюда получаем,

an=C\*pn+D\*pn\*n

1. По пункту 4: решить рекуррентное соотношение и программно реализовать вычисление 5 члена числовой последовательности через рекурсию и полученное решение.

#include <iostream>

using namespace std;

int Recurs(int);

int main() {

setlocale(0, "rus");

cout << "Решение линейного однородного рекуррентного соотношения 2-го порядка: " << endl;

int n = 5;

cout << "5 член числовой последовательности:\n";

cout << "Рекуррентно: ";

cout << (1 + n \* 3) \* pow(-2, n);

cout << endl << "Рекурсивно: ";

cout << Recurs(n);

int size = 3;

int\* mas = new int[size];

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

{

mas[i] = i + 1;

}

int ch = 1;

for (int i = 1; i < size + 1; i++)

{

ch = ch \* i;

}

}

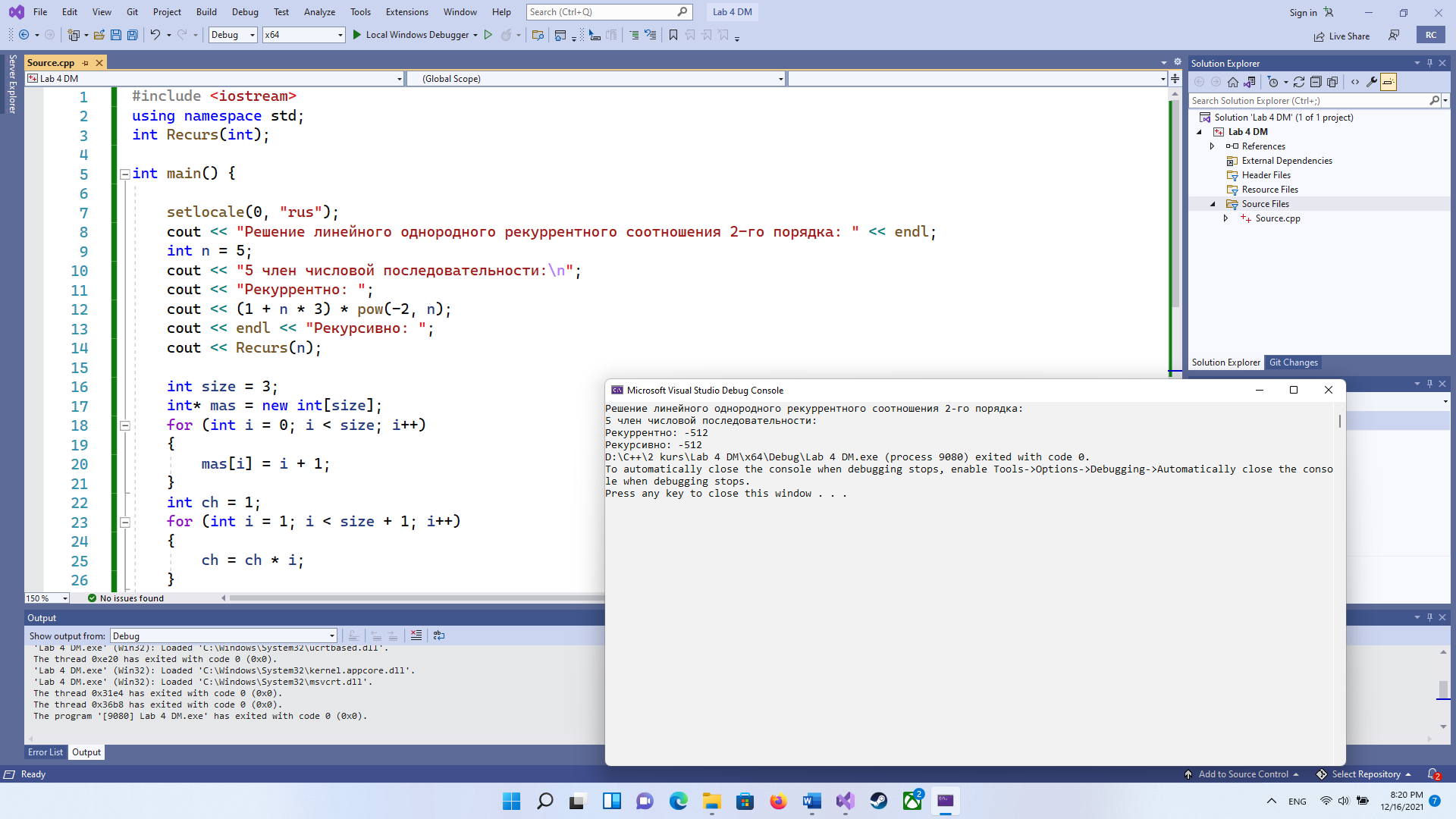
int Recurs(int n) {

if (n == 0) return 1;

else if (n == 1) return -8;

else return (-4) \* Recurs(n - 1) - 4 \* Recurs(n - 2);

}



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

#include <iostream>

#include <cmath>

using namespace std;

void antilex(int\*, int, int);

void reverse(int\* B, int size);

int main()

{

int choise;

int N, n, k;

cout << "Enter the number of items: ";

cin >> N;

int\* A = new int[N];

int\* B = new int[N];

bool exit = true;

while (exit)

{

{

cout << " Antilexicographic order:" << endl;

for (int i = 0; i < N; i++) {

A[i] = i + 1;

}

antilex(A, N - 1, N);

break;

}

delete[] A;

delete[] B;

return 0;

}

}

void antilex(int\* A, int size, int N)

{

if (size == 0)

{

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

{

cout << A[i] << " ";

}

cout << endl;

}

else

{

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

{

antilex(A, size - 1, N);

if (A[i] < A[size])

{

swap(A[i], A[size]);

reverse(A, size - 1);

}

}

}

}

void reverse(int\* B, int size)

{

int i = 0,

j = size;

while (i < j)

{

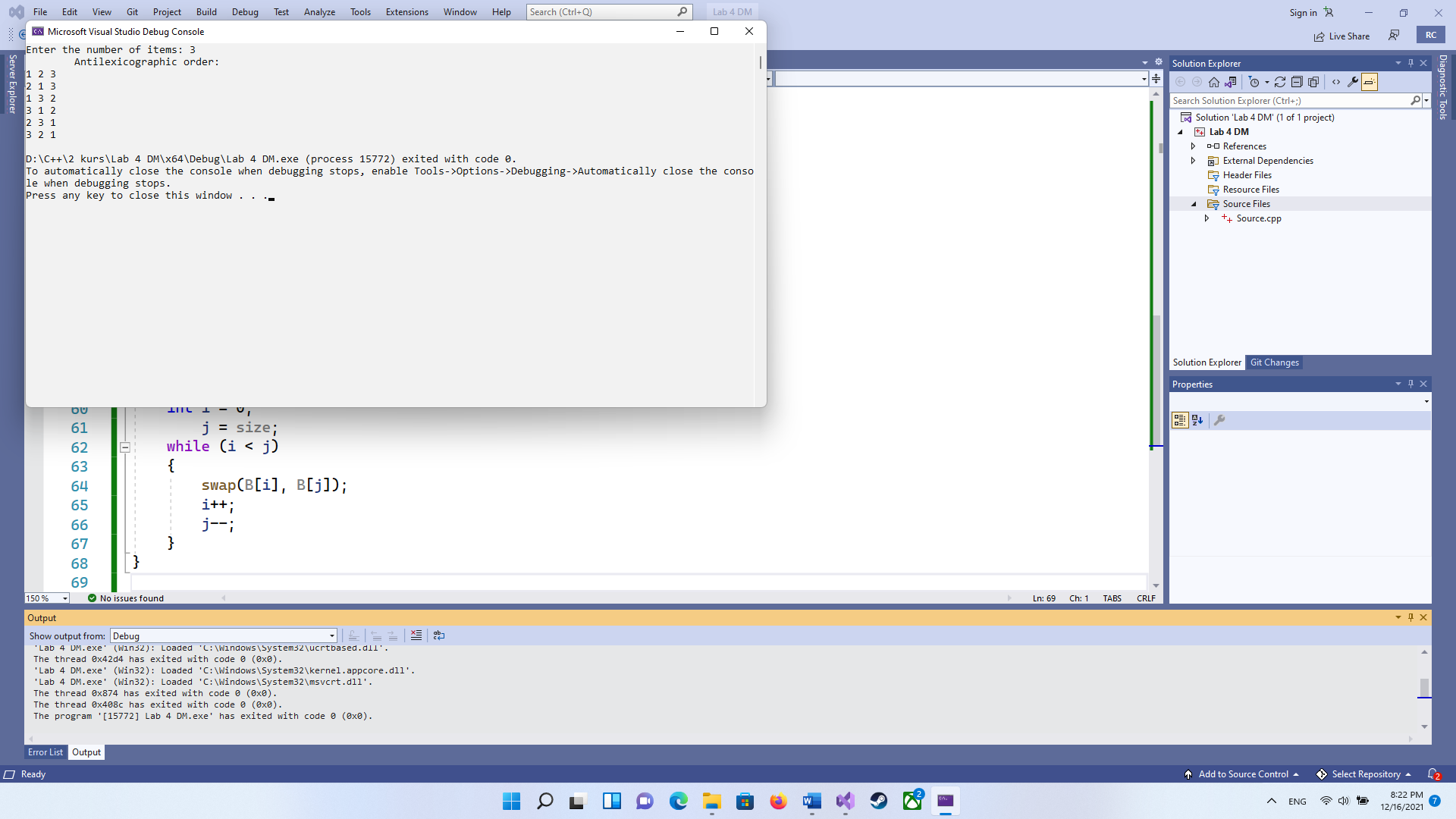
swap(B[i], B[j]);

i++;

j--;

}

}



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

Код:

#include <iostream>

using namespace std;

void gen(int, int, int[], int);

int main()

{

setlocale(0, "");

int i, n, k;

int b[100];

cout << "Enter n(power of the multitude):";

cin >> n;

cout << "Enter k (power of a subset): ";

cin >> k;

gen(n, k, b, 0);

system("pause");

return 0;

}

void gen(int n, int k, int b[], int i)

{

if (i == k)

{

for (int j = 0; j < k; j++) cout << b[j];

cout << endl;

}

else

{

for (b[i] = (i > 0 ? b[i - 1] + 1 : 1); b[i] <= n; b[i]++)

gen(n, k, b, i + 1);

}

}

