МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА ФРАНКА

Факультет електроніки і комп’ютерних технологій

Кафедра системного проєктування

**Звіт**

про виконання лабораторної роботи № 1

з дисципліни «Прикладна статистика та ймовірнісні процеси»

«Комбінаторика»

**Виконав:**

студент 2 курсу

групи Феп-22

Линва В.А.

**Перевірив:**

Доцент кафедри РКИ, к.ф.-м.н

Сас Н. Б.

**Львів 2022**

**Мета:** Ознайомитись із темою «Комбінаторика». Виконати поставлені перед мною завдання.

**Хід роботи**

**Варіант №3**

1. Написати програму яка виконує реалізацію індивідуального завдання:

«Задане додатне ціле число n і невідємне ціле число r, r<n. Розташувати у лексикографічному порядку всі розміщення без повторень із r елементів множини {1, 2, …, n}.

1. Побудувати розклад ( x + y )6.

Результат виконання:

Розташування у лексикографічному порядку всі розміщення без повторень.

|  |
| --- |
|  |

Розклад бінома ( х + у )6

|  |
| --- |
|  |

**Висновок:** в процесі виконання цієї роботи, я пригадав тему комбінаторики та виконав поставлене перед мною індивідуальне завдання.

**Додаток**

#include <iostream>

#include <string>

using namespace std;

int factorial(int num) {

int result = 1;

for (int i = 2; i <= num; i++) result \*= i;

return result;

}

int placements(int total, int chosen) // Розміщення без повторень

{

return factorial(total) / factorial(total - chosen);

}

string binom(int power, char operation = '+') {

string result = "";

int coefficient;

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

// coefficient

coefficient = placements(power, i);

if (operation == '-' && (power - i) % 2 == 1)

result += " - ";

else if (i > 0)

result += " + ";

if (coefficient > 1)

result += to\_string(coefficient);

// x

if (power - i == 1)

result += "x";

else if (power - i > 1)

result += "(x^" + to\_string(power - i) + ")";

// y

if (i == 1)

result += "y";

else if (i > 1)

result += "(y^" + to\_string(i) + ")";

}

return result;

}

string show(int\* set, int\* placements, int placementsSize) {

string result = "";

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

result += to\_string(set[placements[i]]) + " ";

}

return result;

}

bool next(int setSize, int\* placements, int placementsSize) {

bool no\_last = false;

for (int i = placementsSize - 1; i >= 0; i--) {

if (placements[i] != setSize - placementsSize + i) {

no\_last = true;

placements[i]++;

for (int j = i + 1; j < placementsSize; j++) {

placements[j] = placements[i] + j - i;

}

break;

}

}

return no\_last;

}

void showAll(int\* set, int setSize, int placementsSize) {

int\* placements = new int[placementsSize];

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

placements[i] = i;

cout << show(set, placements, placementsSize) << "\n";

while (next(setSize, placements, placementsSize))

cout << show(set, placements, placementsSize) <<

"\n";

}

int main() {

cout << "(x-y)^6 = " << binom(6, '-') << "\n";

cout << "\nEnter set size: ";

int setSize;

cin >> setSize;

cout << "Set: ";

int\* set = new int[setSize];

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

set[i] = i + 1;

cout << set[i] << " ";

}

cout << "\nEnter placements size: ";

int placementsSize;

cin >> placementsSize;

showAll(set, setSize, placementsSize);

return 0;

}