**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТІ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”**

**Кафедра систем штучного інтелекту**

**Лабораторна робота №6**

З дисципліни

“Дискретна математика”

**Виконав:**

студент групи КН-112

Думич Іван

**Викладач:**

Мельникова Н.І.

Львів – 2019 р.

**Тема:** Генерація комбінаторних конфігурацій

**Мета роботи:** набути практичних вмінь та навичок при комп’ютерній реалізації комбінаторних задач.

**Варіант № 5**

**Завдання № 1**. Використовуючи теоретичні відомості, розв’язати наступні комбінаторні задачі за своїм варіантом:

1**.** Скільки різних кілець, що світяться, можна утворити, розмістивши по колу 10 різнокольорових лампочок (кільця вважати однаковими, якщо послідовність кольорів одна й та сама)?

**Розв’язання**

Якщо лампочки розложити у ряд, то всіх можливих розміщень 10!

Якщо 10 лампочок розмістити по колу, то на кожному розміщенню утвориться 10 однакових кілець. Звідси всі можливі кільця  **= 9!**

**Відповідь: 9!** кілець

2. На дев’яти картинках записані цифри 1, 2, 3, 4, 5, 6, 7, 8, 9 (на кожній картці по одній цифрі). Беруть чотири картки і складають з них чотирицифрове число. Скільки різних чисел можна отримати таким чином?

**Розв’язання**

**= =** 3024

**Відповідь:** 3024 чисел.

3. Скільки існує трикутників, довжини сторін яких мають одне з таких значень: 4, 5, 6, 7 см?

**Розв’язання**

Сторонатрикутника повинна бути менша від суми двох інших сторін. Для даних значень ця умова виконується. Тоді застосуємо сполученням з повторюваннями

**= = =** 20

**Відповідь:** існує 20 трикутників.

4. Скільки різних правильних нескоротних дробів можна скласти з чисел 2, 5, 7, 11, 15, 17, 19, 23, 25 так, щоб у кожен дріб входило два числа?

**Розв’язання**

**= =** 36 – всього правильних дробів

Правильні скоротні дроби: **;;.**

Тоді правильні нескоротні дроби: 36-3=33.

**Відповідь:** 33правильних нескоротних дробів можна скласти з даних чисел.

5. Скільки п’ятицифрових чисел можна утворити з цифр 2, 3, 6, 7, 8 (без повторення) так, щоб парні цифри не стояли поруч?

**Розв’язання**

**= 6 –** можна скласти чисел з парних цифр

Можливих місць між парними цифрами:

**= 2**

Тоді поставимо між ними непарні цифри

За правиломдобутку: 2\*6 = 12

**Відповідь:** існує 12 цифр.

6. Скількома способами можна розкласти 28 різних предметів у чотири однакові ящики так, щоб у кожному з них опинилося по 7 предметів?

**Розв’язання**

Це упорядковане розбиття, де n = 28;==== 7

**=**

**Відповідь:** можливих способів буде

7. Знайти кількість цілих додатних чисел, що не більше 1000 і не діляться на жодне з чисел 6, 7 і 15.

**Розв’язання**

* Числа, які кратні 6 : 166
* Числа, які кратні 7 : 142
* Числа, які кратні 15 : 66
* Числа, які кратні 7 і 6 : 23
* Числа, які кратні 7 і 15 : 9
* Числа, які кратні 6 і 15 : 33
* Числа, які кратні 6, 7 і 15: 4

Застосуємо формулу включень-виключень

1000 – x = 166 + 142 + 66 - 23 - 9 - 33 +4 , де x – кількість шуканих чисел

x = 1000 - 166 - 142 - 66 + 23 + 33 + 9 - 4 = 687

**Відповідь:** 687.

**Завдання №2.** Запрограмувати за варіантом обчислення кількості розміщення (перестановок, комбінацій, алгоритму визначення наступної лексикографічної сполуки, перестановки) та формулу Ньютона і побудувати за допомогою неї розклад за варіантом

**Варіант № 5**

Задані додатні цілі числа n та r. Побудувати у лексикографічному порядку всі розміщення з повтореннями із r елементів множини {1, 2, …, n}. Побудувати розклад .

**Програмна реалізація 1:**

#include <stdlib.h>

#include <iostream>

#include <fstream>

using namespace std;

int main()

{

    int n, r,C;

    cout << "n:";

    cin >> n;

    cout << "r:";

    cin >> r;

    int\* N = new int[n];

    for (int i = 0; i < r; i++)

    {

        N[i] = 1;

    }

    C = pow(n, r);

    int k = 0;

    for (int i = 0; i < C; i++) {

        for (int a = 0; a < r; a++) {

            cout << N[a];

        }

        cout << endl;

        if (N[r - 1] == n) {

            k = r - 1;

            for (int a = r-2; a < r; a--) {

                if (N[a] == n)k--;

                else break;

            }

            N[k - 1] += 1;

            for (; k < r; k++) {

                N[k] = 1;

            }

        }

        else {

            N[r - 1]++;

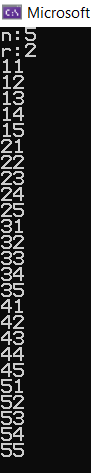
        }

    }

    return 0;

}

**Результат:**

****

**Програмна реалізація 2:**

#include <iostream>

#include <fstream>

using namespace std;

long int factorial(int N)

{

    if (N < 0) return 0;

    if (N == 0) return 1;

    else

        return N \* factorial(N - 1);

}

int main()

{

    setlocale(LC\_CTYPE, "");

    int n,m;

    cout << "(x+y)^n"<<endl;

    cout << "n: ";

    cin >> n;

    for (int k = 0; k <= n; k++) {

        m = factorial(n) / (factorial(n - k) \* factorial(k));

        if(m!=1) cout << factorial(n) / (factorial(n - k) \* factorial(k))<<"\*";

        if(k!=0){

            if(k==1)cout << "y";

            else cout << "y^" << k;

        }

        if (n - k != 0) {

            if(k!=0)cout << "\*";

            if (n - k == 1) cout << "x";

            else cout << "x^" << n - k;

        }

        if (k != n) {

            cout << " + ";

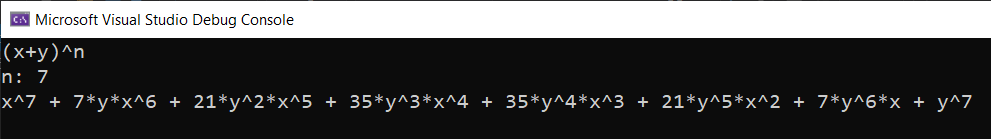
        }

    }

    return 0;

}

**Результат:**



**Висновок**: я набув практичних вмінь та навичок при комп’ютерній реалізації комбінаторних задач.