**Міністерство освіти і науки України**

**Національний університет  
“Львівська політехніка”**

**Кафедра систем штучного інтелекту**

**Лабораторна робота № 6**

з дисципліни

«Дискретна математика»

**Виконав**:

студент групи КН-115  
Поставка Маркіян

**Викладач:**

Мельникова Н. І.

Львів – 2019р.

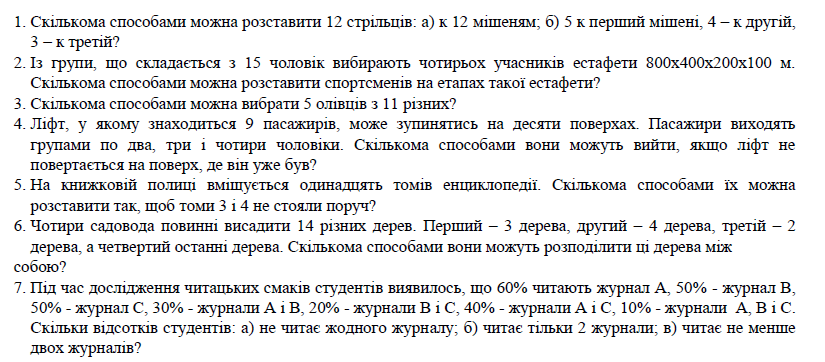
**Тема**: “Генерація комбінаторних конфігурацій”

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

**Додаток 1:**

Варіант 11.

Постановка задачі:



Рішення:

1.

а)

б) P(5,4,3) =

2.

3.

Олівці різні , тому розміщення(порядок важливий)

4.

Маємо 9 пасажирів і групи по 2, 3, 4.

Кількість розподілів пасажирів по групам: P(2,3,4) =

Кожна з груп може вийти лише на одному поверсі з 10, тобто можна вийти

різними способами.

Відповідь: P(2,3,4)\* = =

5.

способами можна розмістити всі томи.

Знайдемо скількома способами можна розмістити томи так щоб наші обрані томи були разом.  
Беремо 2 томи як один елемент. Ці 2 томи можуть бути розміщені також по різному, тобто 3,4 або 4,3.  
Отже, 2\* способами ми можемо розмістити 2 томи разом.

Відповідь: 11! - 2\*9!

6.

Обрахували, що останній садівник має висадити 5 дерев, бо 14 - 3 - 4 - 2 = 5.

Знаходимо перестановки з повторами.

Відповідь: способами.

7.   
Розв’яжемо задачу методом множин.  
Розпишемо потужності множин.

|A|=60; |B| = 50; |C| = 50;

|A∩B| = 30; |B∩C| = 20; |A∩C| = 40; |A∩B∩C| = 10;

1. Не читає жодного:

Знайдемо кількість тих, хто читає хоча б 1. Користуємося формулою включення-виключення для 3х множин.  
| A⋃B⋃C | = |A|+|B|+|C|-|A∩B| - |B∩C| - |A∩C| + |A∩B∩C|= 60+50+50-30-20-40+10=80;

Кількість тих хто не читає жодного = 100 - 80 = 20%.

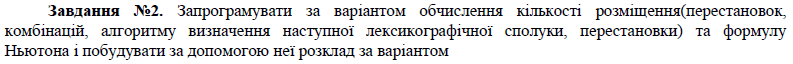
1. Читає 2 журнали:  
   Складаємо всіх студентів які читають по 2 журнали і віднімаємо з кожної множини, тобто 3 рази загалом, кількість тих які читають 3 журнали.

|A∩B| + |B∩C| + |A∩C| - 3\*|A∩B∩C|= 30+20+40-3\*10 = 60%.

1. Читає не менше 2х журналів:  
   Туди входять ті студенти, які читають лише 2 і ті студенти, які читають 3.  
   60% + 10% = 70%; - читає не менше 2х журналів

**Додаток 2:**

Постановка задачі:





Код програми:

// Laba 6.cpp: определяет точку входа для консольного приложения.

//

#include "stdafx.h"

#include <iostream>

#include <conio.h>

using namespace std;

unsigned long long factorial(unsigned long long n)

{

return (n == 1 || n == 0) ? 1 : factorial(n - 1) \* n;

}

unsigned long long coef(int n, int k)

{

return factorial(n) / (factorial(k)\*factorial(n - k));

}

void newton()

{

cout << "Newton's binominal expansion" << endl;

cout << "==================\n";

int n;

cout << "Equation: (a-b)^n" << endl; // My power == 10

cout << "Enter n" << endl;

cin >> n;

if (n == 0)

{

cout << "Binomial expansion: " << endl;

cout << 1 << endl;

return;

}

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

{

unsigned long long result;

if ((n - k) % 2 == 1)

{

if (coef(n, k) == (-1))

{

cout << "-" << "a^" << k << "b^" << n - k;

}

else cout << "-" << coef(n, k) << "a^" << k << "b^" << n - k;

}

else

{

if (coef(n, k) == (1))

{

cout << "+" << "a^" << k << "b^" << n - k;

}

else cout << "+" << coef(n, k) << "a^" << k << "b^" << n - k;

}

}

\_getch();

cout << endl;

}

void swap(int &a, int &b)

{

int temp = a;

a = b;

b = temp;

}

// Reverse by swapping elements from both edges of array

void reverse(int \*array, int i, int j)

{

int temp = 0;

while (i < j) {

temp = array[i];

array[i] = array[j];

array[j] = temp;

i++;

j--;

}

}

bool next\_permutation(int array[], int length) {

if (length == 0)

return false;

int i = length - 1; // Index of target element

while (i > 0 && array[i - 1] >= array[i]) // Going through mas from end to first decreasing element

i--;

if (i == 0) return false; // This is final permutation if target is first element

// Looking for element just bigger than target from end

int j = length - 1;

while (array[j] <= array[i - 1])

j--;

// Swapping

swap(array[i - 1], array[j]);

// Reverse elements after target

j = length - 1;

reverse(array, i, j); // Reverse by swapping elements from both edges of array

return true; // There are more permutations

}

void permutation()

{

cout << "Permutations: " << endl;

cout << "==================\n";

int n = 0;

cout << "Enter amount of numbers" << endl;

cin >> n;

int \*a = new int[n];

cout << "\nPermutations of sequence: " << endl;

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

{

a[i] = i + 1;

cout << a[i] << " ";

}

cout << endl;

while (true)

{

if (!next\_permutation(a, n)) break;

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

{

cout << a[i] << " ";

}

cout << endl;

}

delete[] a;

\_getch();

cout << endl;

}

int main()

{

permutation();

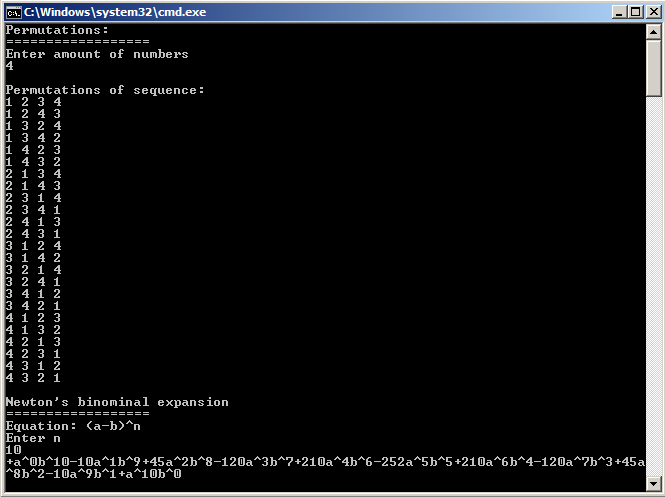
newton();

\_getch();

return 0;

}

Результат роботи програми:



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