**Лабораторна робота №3**

**Тема:**ГЕНЕРАЦІЯ КОМБІНАТОРНИХ КОНФІГУРАЦІЙ

*Мета:*навчитися розв’язувати задачі з комбінаторики, засвоїти правило добутку і суми.

**Зміст роботи:**

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

***Завдання 1.1:***

Скільки різних «слів» можна скласти з слова: а) «книга»; б) «телевізор».

а) у слові «книга» усього 5 різних букв (n = 5), тому маємо перестановку:

б) слово «телевізор» містить однакові букви, тому шукані слова будуть перестановками з повтореннями. Маємо сукупність із 9 букв ( n = 9 ), що містить букву б що повторюється двічі, то кількість різних слів можна скласти:

***Завдання 1.2:*** Скількома способами можна розділити 8 різних ручок між 4 учнями, якщо кожний з них може остатися і без ручки?

***Завдання 1.3:*** У лікарні 15 палат. Лікар веде п’ять з них. Скількома способами він може підібрати собі палати для лікування?

Із 15-ти палат лікар має вибрати п’ять, причому порядок у цьому випадку не важливий. Тоді всього способів:

***Завдання 1.4:*** Скількома способами можна сформувати групу №1 з трьох учнів і одного викладача, якщо є 80 учнів і 3 викладача; чи групу №2 з п’яти учнів і двох викладачів, якщо є 20 учнів і 3 викладача?

Якщо до №1 групи увійде 3 учня, яких вибираємо з 80 учнів способами та одного викладача з 3 викладачів способами, то усього за правилом добутку маємо способів.

Якщо до №2 групи увійде 5 учнів з 20 учнів способами та 2 викладача з трьох способами, то усього за тим же правилом маємо способів.

За правилом суми маємо всього:

способів

***Завдання 1.5:*** Скількома способами можна по кругу поставити 5 різних ляльок та 3 різні м’які іграшки так, щоб при цьому м’які іграшки не стояли поруч?

З початку поставимо в круг ляльки, це можливо зробити – способами. Потім будемо ставити між ними м’які іграшки. Усього можливих міст маємо – 6, з котрих треба вибрати п’ять без повторювань та упорядковано, тому різних способів буде – . За правилом добутку усього різних способів поставити по кругу ляльок та іграшки буде – 120 \* 120 = 14 400.

***Завдання 1.6:*** Дев’ятьох студентів необхідно розподілити на три групи по 3 студента, для відправлення цих груп на різні конференції. Конференції проходять у різних п’ятьох містах, з яких необхідно вибрати три. Скількома способами можна відправити цих студентів на можливі конференції?

З початку виберемо три міста, це можливо зробити:

потім розіб’ємо студентів на три групи по 3 студента:

Далі за правилом добутку отримаємо – 1680 \* 10 = 16800 різних способів.

***Завдання 1.7:*** Лікар веде чотири палат з номерами 1, 2, 3, 4. Скільки способів обходу лікарем палат так, щоб порядок заходу лікарем до палати не відповідав її номеру?

За формулою включень та виключень обчислюємо кількість безладів:

**Завдання №2**

Написати програму, яка дає можливість вибору з 6 варіантів комбінацій, а саме розміщення без повторень та з повторенням, сполучення без повторень та з повторенням, перестановки звичайні та з повторенням елементів. При кожному виборі далі реалізувати:

1) введення натуральних чисел n та k , а в останньому випадку чисел (для перестановок з повторенням);

2) виведення обраних комбінацій, а також кількості отриманих комбінацій, тобто підрахувати числа у відповідності до зробленого вибору типу наборів на екран.

Лістинг програми:

using System;

using System.Collections.Generic;

using System.Text;

using static System.Console;

namespace Exs2

{

class Program

{

static int Fact(int n)

{

if (n == 1 || n == 0) return 1;

return n \* Fact(n - 1);

}

static int[] Distinct(int[] a)

{

HashSet<int> list = new HashSet<int>(a);

int[] b = new int[list.Count];

list.CopyTo(b);

return b;

}

static int DigitCount(int[] a, int b)

{

int count = 0;

foreach (var i in a)

if (i == b) count++;

return count;

}

static int Number(int n)

{

bool check;

int[] a = new int[n];

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

{

do

{

if ((check = int.TryParse(ReadLine(), out a[i])) != true)

WriteLine("Помилка, введіть правильне значення");

}

while (!check) ;

}

int temp = Fact(n);

foreach (var i in Distinct(a))

{

WriteLine("Число: {0} - повторюється {1} раз(ів)", i, DigitCount(a, i));

temp = temp / Fact(DigitCount(a, i));

}

return temp;

}

static int Word(string s)

{

List<char> chars = new List<char>();

for (int i = 0; i < s.Length; i++)

{

chars.Add(s[i]);

}

int count = 0, res = 1, k = s.Length;

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

{

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

{

if (j == 0)

count++;

if (j != 0)

{

if (chars[j] == chars[i] && i != j)

{

chars.RemoveAt(j);

count++;

k--;

}

}

}

res \*= Fact(count);

count = 0;

}

return Fact(s.Length) / res;

}

static void PrintMenu()

{

Write("\n- - - - - - Меню - - - - - -\n" +

"1. Звичайна перестановка\n" +

"2. Перестановка з повтореннями\n" +

"3. Розміщення без повторень\n" +

"4. Розміщення з повтореннями\n" +

"5. Сполучення без повторень\n" +

"6. Сполучення з повтореннями\n" +

"0. Вихід\n-> ");

}

static void Main(string[] args)

{

Title = "Лабораторна робота №3. Харипончук Марії, гр. ПІ-60";

SetWindowSize(50, 30);

OutputEncoding = Encoding.Unicode;

InputEncoding = Encoding.Unicode;

int menu, n, m;

string str;

bool check;

do

{

Clear();

PrintMenu();

do

{

if ((check = int.TryParse(ReadLine(), out menu)) != true)

WriteLine("!!!Помилка. Введіть коректне значення!");

}

while (!check);

switch (menu)

{

case 1:

{

Clear();

Write("\nЗвичайна перестановка.\nВведіть кількість об'єктів(n): ");

do

{

if ((check = int.TryParse(ReadLine(), out n)) != true)

WriteLine("!!!Помилка. Введіть коректне значення!");

}

while (!check);

Write("\nКількість можливих способів: " + Fact(n) + "\n\n");

ReadKey();

break;

}

case 2:

{

Clear();

Write("\nПерестановка з повтореннями.");

int num;

Write("\nВиберіть, що будете вводити: \n" +

"1. Слово \n" +

"2. Число \n");

do

{

if ((check = int.TryParse(ReadLine(), out num)) != true)

WriteLine("!!!Помилка. Введіть коректне значення!");

}

while (!check);

switch (num)

{

case 1:

Write("Введіть слово для обробки: ");

str = ReadLine();

Write("\nКількість можливих способів: " + Word(str) + "\n\n");

ReadKey();

break;

case 2:

Write("\nВведіть кількість об'єктів(n): ");

do

{

if ((check = int.TryParse(ReadLine(), out n)) != true)

WriteLine("!!!Помилка. Введіть коректне значення!");

}

while (!check);

int[] a = new int[n];

Write("Введіть числа для обробки:\n");

Write("\nКількість можливих способів: " + Number(n) + "\n\n");

ReadKey();

break;

default:

WriteLine("!!!Помилка. Введіть коректне значення!");

break;

}

break;

}

case 3:

{

Clear();

Write("\nРозміщення без повторень.\nВведіть загальну кількість об'єктів(n): ");

do

{

if ((check = int.TryParse(ReadLine(), out n)) != true)

WriteLine("!!!Помилка. Введіть коректне значення!");

}

while (!check);

Write("Введіть кількість об'єктів для розміщення(m): ");

do

{

if ((check = int.TryParse(ReadLine(), out m)) != true)

WriteLine("!!!Помилка. Введіть коректне значення!");

}

while (!check);

Write("\nКількість можливих способів: " + Fact(n) / Fact(n - m) + "\n\n");

ReadKey();

break;

}

case 4:

{

Clear();

Write("\nРозміщення з повтореннями.\nВведіть загальну кількість об'єктів(n): ");

do

{

if ((check = int.TryParse(ReadLine(), out n)) != true)

WriteLine("!!!Помилка. Введіть коректне значення!");

}

while (!check);

Write("Введіть кількість об'єктів для розміщення(m): ");

do

{

if ((check = int.TryParse(ReadLine(), out m)) != true)

WriteLine("!!!Помилка. Введіть коректне значення!");

}

while (!check);

Write("\nКількість можливих способів: " + Math.Pow(n, m \* 1.0) + "\n\n");

ReadKey();

break;

}

case 5:

{

Clear();

Write("\nСполучення без повторень.\nВведіть загальну кількість об'єктів(n): ");

do

{

if ((check = int.TryParse(ReadLine(), out n)) != true)

WriteLine("!!!Помилка. Введіть коректне значення!");

}

while (!check);

Write("Введіть кількість об'єктів для розміщення(m): ");

do

{

if ((check = int.TryParse(ReadLine(), out m)) != true)

WriteLine("!!!Помилка. Введіть коректне значення!");

}

while (!check);

Write("\nКількість можливих способів: " + Fact(n) / (Fact(n - m) \* Fact(m)) + "\n\n");

ReadKey();

break;

}

case 6:

{

Clear();

Write("\nСполучення з повтореннями.\nВведіть загальну кількість об'єктів(n): ");

do

{

if ((check = int.TryParse(ReadLine(), out n)) != true)

WriteLine("!!!Помилка. Введіть коректне значення!");

}

while (!check);

Write("Введіть кількість об'єктів для розміщення(m): ");

do

{

if ((check = int.TryParse(ReadLine(), out m)) != true)

WriteLine("!!!Помилка. Введіть коректне значення!");

}

while (!check);

Write("\nКількість можливих способів: " + Fact(n + m - 1) / (Fact(m) \* Fact(n - 1)) + "\n\n");

ReadKey();

break;

}

case 0:

WriteLine("Ви вийшли з програми!");

break;

default:

WriteLine("!!!Помилка. Введіть коректне значення!");

break;

}

} while (menu != 0);

}

}

}

Демонстрація роботи програми:

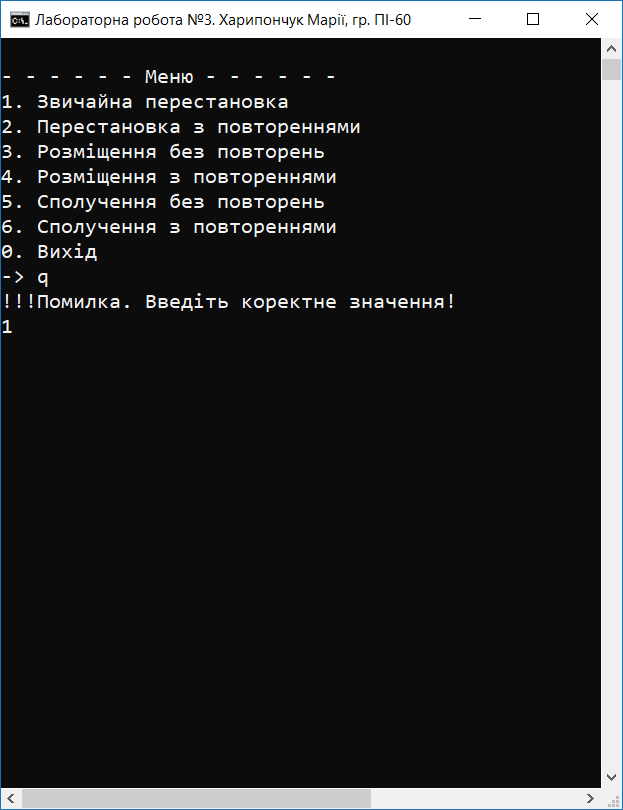


Рис 2.1 Перевірка роботи меню на некоректно введене значення.

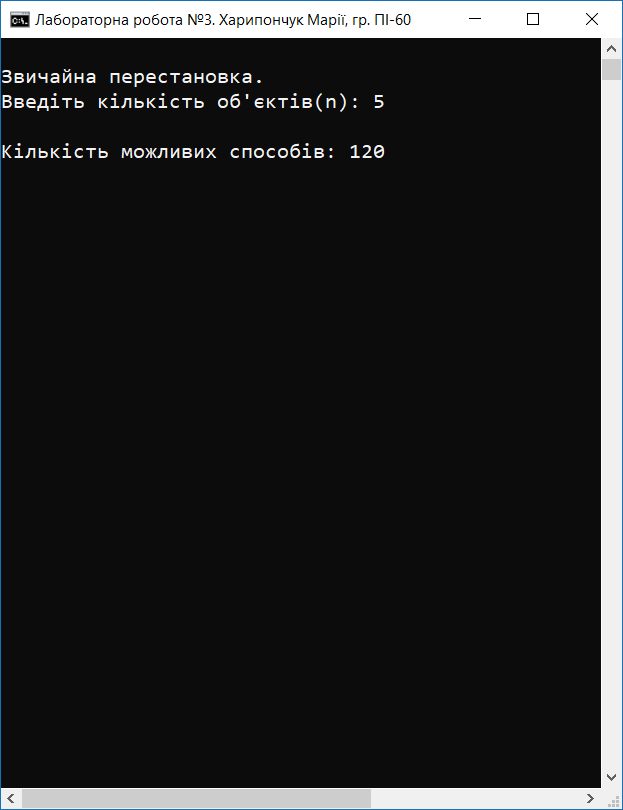


Рис 2.2 Звичайна перестановка.

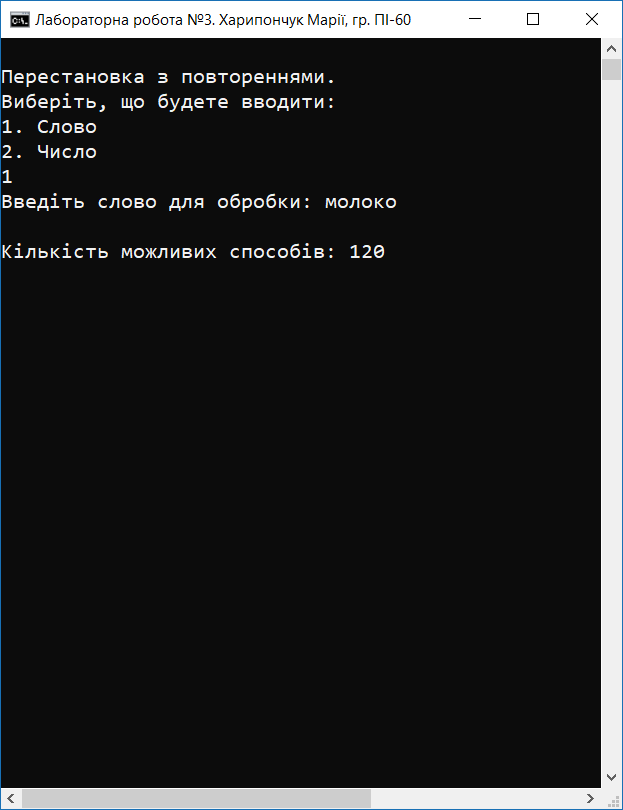


Рис 2.3 Перестановка з повтореннями (слово).

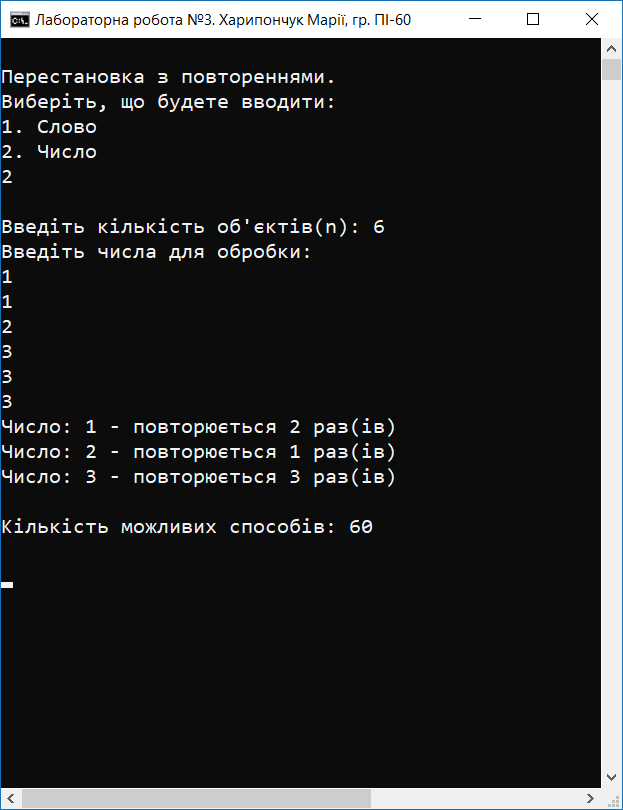


Рис 2.4 Перестановка з повтореннями (число).

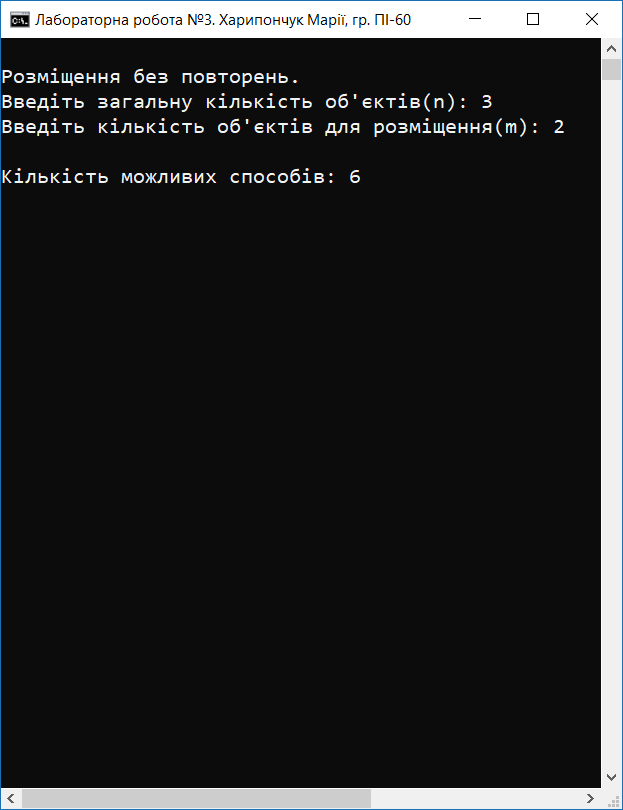


Рис 2.5 Розміщення без повторень.

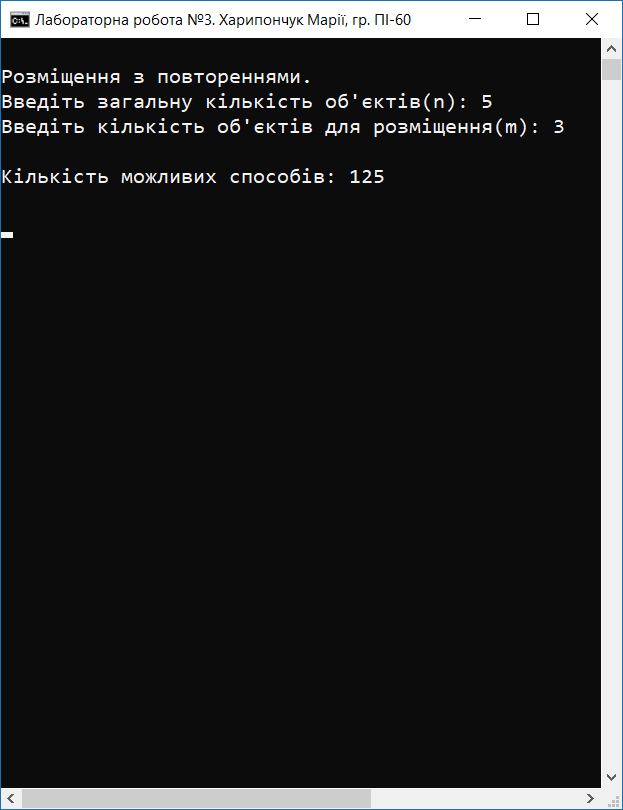


Рис 2.6 Розміщення з повтореннями.

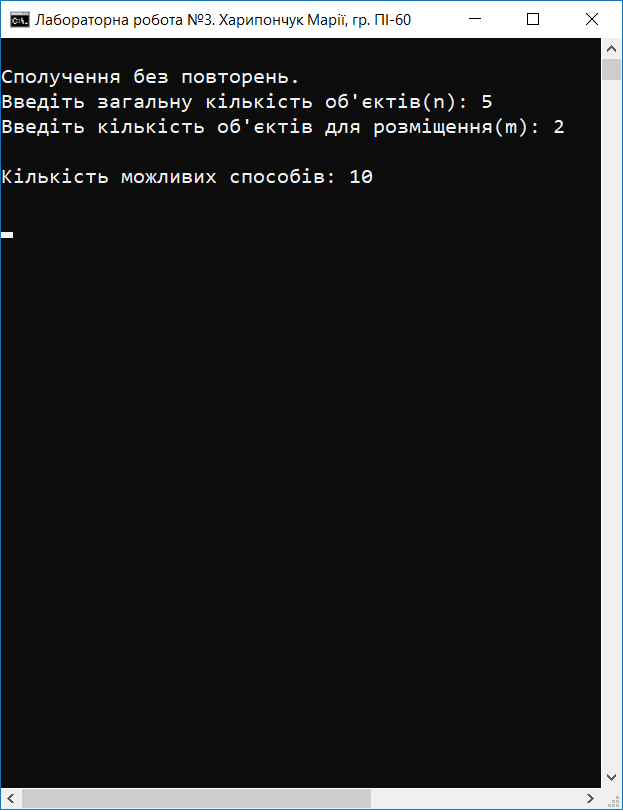


Рис 2.7 сполучення без повторень.

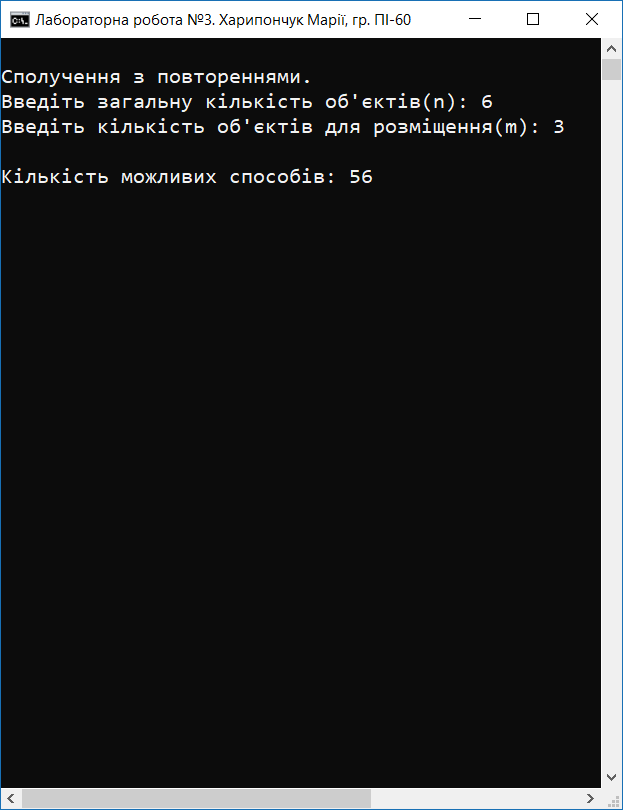


Рис 2.8 сполучення з повтореннями.

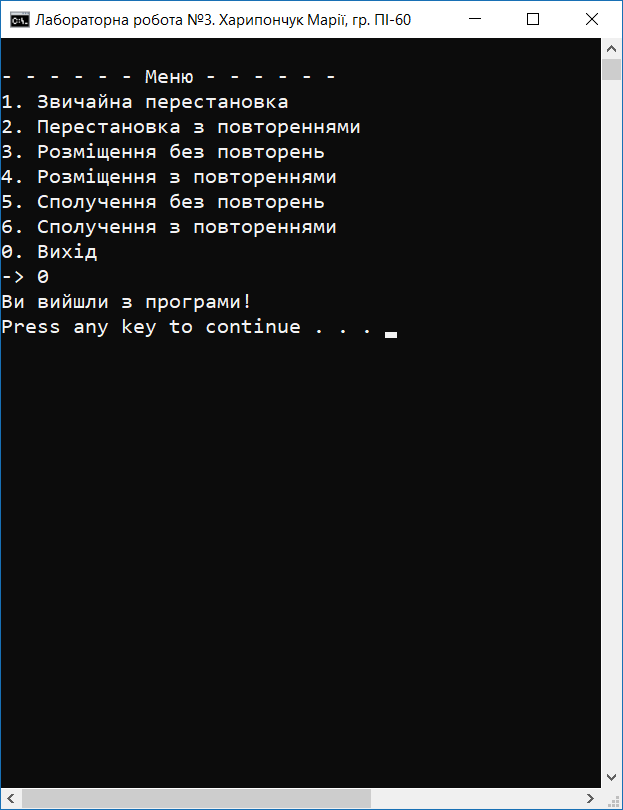


Рис 2.9 Вихід з програми.

**Блок-схема створеної програми:**

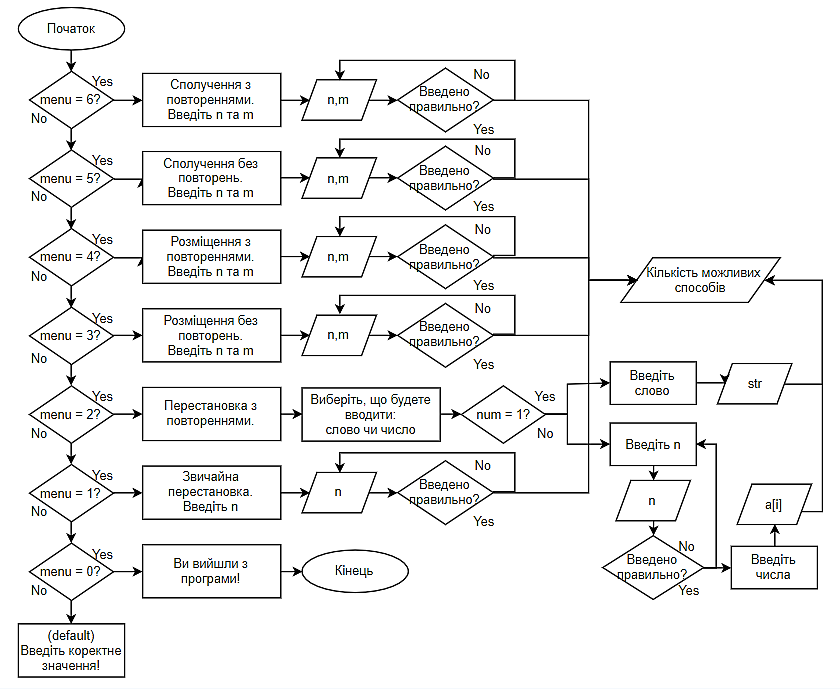


Рис 2.10 Блок-схема програми

***Висновок*:** під час виконання даної лабораторної роботи ознайомлено з областю задач комбінаторики, засвоєно комбінаторні правила суми та добутку, сформульовано вміння розв’язувати задачі, що передбачають використання цих правил.