МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТІ “ЛЬВІВСЬКА ПОЛІТЕХНІКА“

Кафедра систем штучного інтелекту

Лабораторна робота №6

З дисципліни
«Дискретна математика»

Виконала:
Студентка групи КН-115
Галік Вікторія
Викладач:
Мельникова Н. І.

Львів – 2019р.

Тема: Генерація комбінаторних конфігурацій

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

Теоритичні відомості :

Головна задача комбінаторики – підрахунок та перелік елементів у скінчених множинах.

Правило додавання: якщо елемент – x може бути вибрано n способами, а y- іншими m способами, тоді вибір „ x або y” може бути здійснено (m+n) способами.

Правило добутку: якщо елемент – x може бути вибрано n способами, після чого y - m способами, тоді вибір упорядкованої пари (x, y) може бути здійснено (m*n) способами.

Набір елементів $x_{i1}, x_{i2}, \ldots, x_{im}$ з множини $X = \{x_1, x_2, \ldots, x_n\}$ називається вибіркою об’єму m з n елементів – *(n, m) – вибіркою.*

Упорядкована (n, m) – вибірка, в якій елементи не можуть повторюватися, називається *(n, m) – розміщеням*, кількість всіх можливих розміщень обчислюється за формулою:

$$A_n^m = \frac{n!}{(n-m)!}.$$

Упорядкована (n, m) – вибірка, в якій елементи можуть повторюватися, називається *(n, m) – розміщеням з повторюваннями*, кількість всіх можливих таких розміщень обчислюється за формулою:

$$\overline{A_n^m} = n^m.$$

Неупорядкована (n, m) – вибірка, в якій елементи не можуть повторюватися, називається *(n, m) – сполученням*, кількість всіх можливих сполучень обчислюється за формулою:

$$C_n^m = \frac{n!}{m!(n-m)!}.$$

Неупорядкована (n, m) – вибірка, в якій елементи можуть повторюватися, називається *(n,m)-сполученням з повторюваннями*, кількість всіх можливих таких сполучень обчислюється за формулою:

$$\overline{C_n^m} = C_{n+m-1}^m.$$

A_n^n – називається *перестановкою*, а кількість різних перестановок позначається та обчислюється за формулою:

$$P_n = n!.$$

Якщо в перестановках є однакові елементи, а саме перший елемент присутній n_1 разів, другий елемент – n_2 разів, ... , k-ий елемент – n_k разів, причому $n_1+n_2+\ldots+n_k=n$, то їх називають *перестановками з повторенням* та кількість їх можна знайти за формулою

$$P(n_1, n_2, \ldots, n_k) = \frac{n!}{n_1!n_2!\ldots n_k!}.$$

Нехай $X = \{X_1, X_2, \ldots, X_k\}$ - *розбиття множини* X ($|X| = n$) на k підмножин таких, що: $\bigcup_{i=1}^{k} X_i = X$, $X_i \cap X_j = 0$ при $i \neq j$,

$|X_i| = n_i$.

Їх кількість при фіксованих n_i та *упорядкованих* $X_1, X_2, \ldots, X_k$ обчислюється за формулою:

$$C_n^{n_1,n_2,\dots,n_k}(n_1, n_2, \dots, n_k) = \frac{n!}{n_1!n_2!\dots n_k!}.$$

Якщо ж множину X ($|X| = n$) потрібно розбити на підмножини, серед яких для усіх i=1, …, n є $m_i \geq 0$ підмножин з i елементами, де $\sum_{i=1}^{n} i * m_i = n$, та при цьому набір підмножин в розбитті *не є упорядкованим,* тоді їх кількість обчислюється за формулою: $n!$

$$N(m_1, m_2, \dots, m_n) = \frac{n!}{m_1!m_2!\dots m_n!(1!)^{m1}(2!)^{m2}\dots(n!)^{m_n}}.$$

Лексикографічний порядок – це природний спосіб упорядкування послідовностей на основі порівняння індивідуальних символів. На множині всіх розміщень із r елементів означимо порівняння таким чином: $b_1b_2\dots b_r < a_1a_2\dots a_r$, якщо $\exists m : (b_i = a_i\ , i < m) \wedge (b_m < a_m)$.

У такому разі говорять, що перестановка $b_1b_2\dots b_r$ *менша* від перестановки $a_1a_2\dots a_r$, або перестановка $a_1a_2\dots a_r$ *більша* від перестановки $b_1b_2\dots b_r$.

Алгоритм побудови лексикографічно наступного розміщення з повтореннями за розміщенням $a_1a_2\dots a_r$

Алгоритм подібний до звичайного визначення наступного числа.

Крок 1. Знаходимо позицію k першого справа числа, відмінного від n $a_k < n$.

Крок 2. Збільшуємо елемент a_k на одиницю. Елементи a_i, де $i < k$ залишаються без змін. Елементи a_i, де $i > k$ стають рівними одиниці.

Приклад. Нехай $A = \{1, 2, 3\}$. Побудуємо 6 розміщень з повтореннями лексикографічно наступних після 1222. Наступне буде 1223, так як можливо збільшити останній елемент. Після нього буде 1231, оскільки 4-й елемент ми збільшити не можемо, то збільшуємо 3-й, а 4-й ставимо рівним одиниці. Як бачимо, маємо аналогію з переносом розряду, подібно до десяткового числення. Наступні елементи, відповідно будуть 1232,1233, 1311 та 1312.

Алгоритм побудови лексикографічно наступного розміщення без повторень за розміщенням $a_1a_2\dots a_r$

Алгоритм від попереднього відрізняється тим, що у розміщеннях не може бути повторів, і тому потрібно "оновлювати" елементи розміщення, не порушуючи цієї властивості.

Крок 1. Знайдемо множину B "вільних" чисел, яких немає у розміщенні $a_1a_2\dots a_r$: $B = A \setminus \{a_1a_2\dots a_r\}$.

Крок 2. Шукаємо перший справа елемент у розміщенні, який можна збільшити. Якщо у B є елементи, які більші за a_r, то вибираємо серед них такий, що: $b_r = \min_{x \in B}\{x > a_r\}$.

Якщо у B немає елементів, більших за a_r, то додаємо до B елемент a_r, $B = B \cup \{a_r\}$ і шукаємо: $b_{r-1} = \min_{x \in B}\{x > a_{r-1}\}$.

Якщо у B немає елементів, більших за a_{r-1}, то додаємо до B елемент a_{r-1}, і т.д. Продовжуємо цей процес, поки не знайдемо: $b_k = \min_{x \in B}\{x > a_k\}$, або не дійдемо до початку розміщення. Якщо такого b_k не знайдено, то ми дійшли до максимального елементу, і алгоритм завершено. Якщо ні, то переходимо до кроку3.

Крок 3. Обчислюємо наступне розміщення. Записати в k-ту позицію розміщення знайдене число b_k і вилучити його з множини B. Записати у висхідному порядку число b_{k+1}, b_{k+2}, $b_{k+3} \dots b_r$ – найменші з чисел у множині B, розмістивши їх на позиціях $k + 1, k + 2, \dots, r$.

Приклад. Нехай $A = \{1, 2, 3, 4\}$. Побудуємо 4 розміщень без повтореннями лексикографічно наступних після 123. Наступне буде 124, так як можливо збільшити останній елемент (3).

Обчислимо наступне розміщення після 124. Оскільки 3-й елемент ми збільшити не можемо, то збільшуємо 2-й елемент, тобто заміняємо "2" на "3". На останньому місці не можна ставити одиницю, бо вона вже є у розміщенні, і ставимо найменший елемент, такий що його немає в розміщенні , тобто "2". Отже отримуємо 132 .

Наступне розміщення після 132 рівне 134, тому що можливо збільшити останній елемент. Яке розміщення буде після 134? Останній елемент ми не можемо збільшити, але можемо збільшити передостанній. Тому наступне розміщення 142.

Алгоритм побудови лексикографічно наступної перестановки за перестановкою $a_1 a_2 \ldots a_n$

Наведемо кроки алгоритму.

Крок 1. Знайти такі числа a_j і a_{j+1}, що $(a_j<a_{j+1},) \wedge (a_{j+i} \geq a_{j+2} \geq \ldots \geq a_n)$. Для цього треба знайти в перестановці першу справа пару сусідніх чисел, у якій число ліворуч менше від числа праворуч.

Крок 2. Записати в j-ту позицію таке найменше з чисел $a_{j+1}, a_{j+2},\ldots, a_n$, яке водночас більше, ніж a_j.

Крок 3. Записати у висхідному порядку число a_j і решту чисел $a_{j+1}, a_{j+2},\ldots,a_n$ у позиції $j+1,\ldots, n$.

Приклад. Побудуємо 2 перестановки, наступні в лексикографічному порядку за 34521. Згідно першого кроку $j=2$, бо 4<5>2>1. Отже, перше число (3) залишається на місці, а збільшується друге число(4). Розглянемо послідовність чисел 521. Серед них найменше число, більше від 4, це 5. Тепер на другому місці 5, а решту чисел розміщуємо у вихідному порядку: 35124.

Побудуємо наступну перестановку після 35124. Згідно першого кроку $j=4$, і щоби отримати наступну перестановку, треба збільшити “2”, поставивши замість нього “4”, так як справа немає іншого числа більше “2”. Переставивши місцями два останніх числа, ми отримаємо 35142.

Зауважимо також, що для перебору перестановок з повтореннями чи без повторень алгоритм не відрізнятиметься. Єдина відмінність полягатиме у тому, що для перестановок без повторень рівності у кроці 1 будуть строгими.

Алгоритм побудови лексикографічно наступного сполучення з повтореннями за сполученням $a_1 a_2 \ldots a_r$

Алгоритм подібний до алгоритмів генерування розміщень, але має одну особливість, яка полягає в наступному: якщо сполучення впорядковане у висхідному порядку, то кожен наступний елемент сполучення не менший за попередній.

Крок 1. Знаходимо позицію k першого справа числа, відмінного від n: $a_k < n$.

Крок 2. Збільшуємо елемент a_k на одиницю $b_k = a_k + 1$.

Елементи зліва a_i залишаються без змін $b_i = a_i$, де $i < k$.

Елементи справа a_i, де $i > k$ стають рівними b_k, $b_i = b_k$, де $i > k$.

Алгоритм побудови лексикографічно наступного сполучення без повторень за сполученням $a_1 a_2 \ldots a_r$

Перед тим як розглянути алгоритм, розглянемо одну особливість. Якщо сполучення впорядковане у висхідному порядку і не має повторів, то кожен наступний елемент сполучення більший від попереднього принаймі на одиницю.

Тоді максимальне значення, яке може набувати його останній елемент, рівне n. Максимум для передостаннього елементу рівний $n-1$, а не n. Доведемо це від зворотнього. Припустимо. що останній елемент рівний n, тоді наступний елемент має бути рівний $n+1$, але такого елементу немає в множині.

Отже максимум передостаннього елементу $n-1$. Аналогічно можна довести, що максимум елементу на k-ій позиції рівний $n-(r-k)$. Мінімум елементу – попереднє число сполучення, збільшене на одиницю.

Крок 1. Знайдемо перший справа елемент сполучення, який можна збільшувати. Він має бути менший за свій допустимий максимум, тобто $a_k < n - r + i$.

Крок 2. Збільшимо елемент a_k на одиницю $b_k = a_k + 1$.

Крок 3. Елементи зліва від a_i не змінюємо $b_i = a_i$, $i<k$.

Крок 4. Елементи справа змінюємо на мінімальні, тобто такі, що на одиницю більші від попереднього: $b_i = b_{i-1} + 1 = a_k + i - k$, $i>k$.

Приклад. Нехай $A=\{1,2,3,4,5\}$. Знайдемо сполучення, наступне за $\{1,2,5,6\}$ у лексикографічному порядку. Це сполучення подамо рядком 1256. Маємо $n = 6$, $r = 4$. Перший справа з таких елементів, що $a_i \neq 6 - 4 + i$, — це $a_2=2$. Для обчислення наступного більшого сполучення збільшуємо a_2 на 1 та одержуємо $a_2 = 3$. Тепер нехай $a_3 = 3+1=4$ і $a_4 = 3 + 2 = 5$. Отже, наступне в лексикографічному порядку сполучення — те, що зображене рядком 1345, тобто $\{1, 3, 4, 5\}$.

*Біномом Ньютон*а називають формулу для обчислення виразу $(a+b)^n$ для натуральних n.

$$(a+b)^n = \sum_{k=0}^{n} C_n^k a^k b^{n-k}$$

ІНДИВІДУАЛЬНІ ЗАВДАННЯ

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

Варіант № 5

1. Скільки різних кілець, що світяться, можна утворити, розмістивши по колу 10 різнокольорових
лампочок (кільця вважати однаковими, якщо послідовність кольорів одна й та сама)?
2. На дев'яти картинках записані цифри 1, 2, 3, 4, 5, 6, 7, 8, 9 (на кожній картці по одній цифрі). Беруть
чотири катки і складають з них чотирицифрове число. Скільки різних чисел можна отримати таким
чином?
3. Скільки існує трикутників, довжини сторін яких мають одне з таких значень: 4, 5, 6, 7 см?
8
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?
5. Скільки п'ятицифрових чисел можна утворити з цифр 2, 3, 6, 7, 8 (без повторення) так, щоб парні
цифри не стояли поруч?
6. Скількома способами можна розкласти 28 різних предметів у чотири однакові ящики так, щоб у кожному з них опинилося по 7
предметів?
7. Знайти кількість цілих додатних чисел, що не більше 1000 і не діляться на жодне з чисел 6, 7 і 15.

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 $(x + y)^7$.

Розв'язання :

№1 Кільце не має фіксованого початку, тому візьмемо один елемент як фіксовану точку. Отже, можна скласти кільце такою кількістю способів :

$P_{10-1} = 9! = 1 * 2 * 3 * 4 * 5 * 6 * 7 * 8 * 9 = \mathbf{362880}$ **способів.**

№2 1) • не всі елементи взято

• порядок не важливий

Отже, сполучення (без повторень) :

C_9^4 = 9!/(5!*4!) = (6*7*8*9)/(1*2*3*4) = 6*7= 42 способи.

Існує 42 способи витягнути 4 картки з 9

2) • всі 4 елементи взято

• порядок важливий

Отже, перестановка :

P_4 = 4! = 1*2*3*4 = 24 способи

Існує 24 способи скласти число

За правилом добутку : n = 42*24 = **1008 способи.**

№3 • не всі елементи взято

• порядок не важливий

Отже, сполучення (без повторень):

C_4^3 = 4!/(1!*3!) = 4 - можна утворити **4 трикутники**

№4 Сполучення без повторень, оскільки мають входити 2 різні числа:

Для чисельника "2" існує 8 можливих знаменників :

2/5, 2/7, 2/11, 2/15, 2/17, 2/19, 2/23, 2/25

Для чисельника "5" існує 5 можливих знаменників :

5/7, 5/11, 5/17, /5/19, 5/23

Аналогічно :

Для "7" існує 6 знаменників

Для "11" існує 5 знаменників

Для "15" існує 3 знаменників

Для "17" існує 3 знаменників

Для "19" існує 2 знаменників

Для "23" існує 1 знаменників

За правилом суми : **n = 8+5+6+5+3+3+2+1=33**

Отже, існують 33 правильних нескоротних дробів.

№5 Оскільки, парні числа не можуть бути поруч, то розміщення парних і непарних можливе тільки таке :

Парне	Непарне	Парне	Непарне	Парне

1) Кількість способів можливих розташувань парних чисел

   • перестановка : P_3 = 3! = 1*2*3 = 6 способів.

2) Кількість способів можливих розташувань непарних чисел

   • перестановка : P_2 = 2! = 1*2 = 2 способи.

3) За правилом добутку n = 6*2 = 12 способів

   Отже, ці числа можна розмістити **дванадцятьма способами.**

№ 6 1) Заповнюємо перший ящик :

• не всі елементи взято

• порядок неважливий

Сполучення без повторень :

C_{28}^{7} = 28! / (21!*7!) = (22*23*23*25*26*27*28) / (1*2*3*4*5*6*7) = 1184040 способів.

2) Заповнюємо другий ящик (з кожним ящиком вибірка зменшується на 7)

Аналогічно :

C_{21}^{7} = 21! / (14! * 7!) = (15*16*17*18*19*20*21) / (1*2*3*4*5*6*7) = 116280 способів

3) Заповнюємо третій ящик :

Аналогічно :

C_{14}^{7} = 14! / (7! * 7!) = (8*9*10*11*12*13*14) / (1*2*3*4*5*6*7) = 6864 способи.

4) Заповнюємо четвертий ящик :

Аналогічно :

C_{7}^{7} = P_7 = 7! = 1*2*3*4*5*6*7 = 5040 способів

Оскільки ми заповнюємо всі чотири ящики, то за правилом добутку :

N = C_{28}^{7} * C_{21}^{7} * C_{14}^{7} * P_7 = 1184040*116280*6864*5040 **= 476298494e18 способів**.

№7 A = { x : x є z, 0 <= x <= 1000 }

|A| = 1000

B = { x : x є A, x - ділиться на 6 }

|B| = 1000/6 = 166 чисел

C = { x : x є A, x - ділиться на 7 }

|C| = 1000/7 = 142 чисел

D = { x : x є A, x - ділиться на 15 }

|D| = 1000/15 = 66 чисел

| B ⋂ C|= 23 числа

| C ⋂ D | = 9 чисел

|B ⋂ D | = 33 числа

| B ⋂ C ⋂ D | = 4 числа

| B ⋃ C ⋃ D | = 166+142+66-23-9-33+4 = 346 чисел, що діляться хоча б на одне із заданих

|A|\| B ⋃ C ⋃ D | =1000-346 = **654** числа, що не діляться на жодне із заданих чисел

Програмна реалізація :

```cpp
#include <iostream>
#include <math.h>

using namespace std;


void Print(int* a, int n)
{
	 static int num = 1;
	cout <<"\t\t\t"<< num++ << ":  ";
	for (int i = 0; i < n; i++) {             // вивід комбінацій
		cout << a[i] << " ";
	}
	cout << endl;
}

////////////// лексикографічний порядок ////////////////
bool Set(int* a, int n, int m)
{
	int j = m - 1;                                  // перший елемент справа
	while (j >= 0 && a[j] == n) j--;
	if (j < 0) return false;
	if (a[j] >= n) j--;
	a[j]++;
	if (j == m - 1) return true;
	for (int k = j + 1; k < m; k++) {
		a[k] = 1;
	}
	return true;
}
///////////////////////////////////////////////////////////

int factorial(int a) {                               // функція факторіалу
	if (a == 1) {
		return a;
	}
	else if (a == 0)
	{
		return 1;
	}
	return a * factorial(a - 1);
}

long long int koef(int n, int k) {                                   // значення C
	return factorial(k) / (factorial(n) * factorial(k - n));
}

int main()
{
	/////////////// first part ////////////////////

	cout << endl << "\t______________FIRST_TASK________________" << endl<<endl;
	int n, m, * a;
	cout << "\tEnter N : ";                        // із скількох елементів будемо вибирати
	cin >> n;
	cout << "\tENTER M : ";                        // по скільки елементів будемо вибирати
	cin >> m;
	int size = n > m ? n : m;                      // обирається розмір масиву
	a = new int[size];
	for (int i = 0; i < size; i++) {               // створення масиву вибірки
		a[i] = 1;
	}
	Print(a, m);
	while (Set(a, n, m)) {
		Print(a, m);
	}
	cout << endl;

	///////////////// second part //////////////////
```

```
	cout << endl << "\t_______________SECOND_TASK_________________" << endl;
	long long int x, y;
	cout << endl << "\tEnter X:";
	cin >> x;
	cout << "\tEnter Y:";
	cin >> y;
	cout << endl << "   THE RESULT : " << endl << endl;
	if (x == y) {                                          // випадок, коли х та у рівні
		cout << "(x + y)^7 = 0";
		return 0;
	}
	long long int bin = 0;
	int p = 7;                                             // степінь до якого підносимо
	for (int i = 0; i <= p; ++i)                           // обчислення бінома
	{
		bin += koef(i, p) * pow(x, i) * pow(y, p - i);
	}
	cout << "(x + y)^7 = ";
	for (int i = 0; i < p; ++i) {                          // розкладання
		if (i % 2)  cout << " - ";
		else {
			cout << " + ";
		}
		cout << koef(i, p) << " * (x^" << i << ") * (y^" << p - i << ")";
	}
	cout << " = " << bin << "\n";
	cout << endl << endl << endl;

	system("pause");
	return 0; }
```

Результат програми :

TEST #1

```
D:\Desktop\Дискретна математика\Лабораторна №6\laaaab6\Debug\laaaab6.exe

	_____________FIRST_TASK_______________

	Enter N : 4
	ENTER M : 2
			1:  1 1
			2:  1 2
			3:  1 3
			4:  1 4
			5:  2 1
			6:  2 2
			7:  2 3
			8:  2 4
			9:  3 1
			10:  3 2
			11:  3 3
			12:  3 4
			13:  4 1
			14:  4 2
			15:  4 3
			16:  4 4


	_____________SECOND_TASK_______________

	Enter X:8
	Enter Y:3

   THE RESULT :

(x + y)^7 =  + 1 * (x^0) * (y^7) - 7 * (x^1) * (y^6) + 21 * (x^2) * (y^5) - 35 * (x^3) * (y^4) + 35 * (x^4) * (y^3) - 21
 * (x^5) * (y^2) + 7 * (x^6) * (y^1) = 19487171



Press any key to continue . . .
```

TEST #2

D:\Desktop\Дискретна математика\Лабораторна №6\laaaab6\Debug\laaaab6.exe

```
______________FIRST_TASK________________

Enter N : 5
ENTER M : 3
                1:  1 1 1
                2:  1 1 2
                3:  1 1 3
                4:  1 1 4
                5:  1 1 5
                6:  1 2 1
                7:  1 2 2
                8:  1 2 3
                9:  1 2 4
                10:  1 2 5
                11:  1 3 1
                12:  1 3 2
                13:  1 3 3
                14:  1 3 4
                15:  1 3 5
                16:  1 4 1
                17:  1 4 2
                18:  1 4 3
                19:  1 4 4
                20:  1 4 5
                21:  1 5 1
                22:  1 5 2
                23:  1 5 3
                24:  1 5 4
                25:  1 5 5
                26:  2 1 1
                27:  2 1 2
                28:  2 1 3
                29:  2 1 4
                30:  2 1 5
                31:  2 2 1
                32:  2 2 2
                33:  2 2 3
```

D:\Desktop\Дискретна математика\Лабораторна №6\laaaab6\Debug\laaaab6.exe

```
32:  2 2 2
33:  2 2 3
34:  2 2 4
35:  2 2 5
36:  2 3 1
37:  2 3 2
38:  2 3 3
39:  2 3 4
40:  2 3 5
41:  2 4 1
42:  2 4 2
43:  2 4 3
44:  2 4 4
45:  2 4 5
46:  2 5 1
47:  2 5 2
48:  2 5 3
49:  2 5 4
50:  2 5 5
51:  3 1 1
52:  3 1 2
53:  3 1 3
54:  3 1 4
55:  3 1 5
56:  3 2 1
57:  3 2 2
58:  3 2 3
59:  3 2 4
60:  3 2 5
61:  3 3 1
62:  3 3 2
63:  3 3 3
64:  3 3 4
65:  3 3 5
66:  3 4 1
67:  3 4 2
68:  3 4 3
69:  3 4 4
70:  3 4 5
```

D:\Desktop\Дискретна математика\Лабораторна №6\laaaab6\Debug\laaaab6.exe

```
69:  3 4 4
70:  3 4 5
71:  3 5 1
72:  3 5 2
73:  3 5 3
74:  3 5 4
75:  3 5 5
76:  4 1 1
77:  4 1 2
78:  4 1 3
79:  4 1 4
80:  4 1 5
81:  4 2 1
82:  4 2 2
83:  4 2 3
84:  4 2 4
85:  4 2 5
86:  4 3 1
87:  4 3 2
88:  4 3 3
89:  4 3 4
90:  4 3 5
91:  4 4 1
92:  4 4 2
93:  4 4 3
94:  4 4 4
95:  4 4 5
96:  4 5 1
97:  4 5 2
98:  4 5 3
99:  4 5 4
100:  4 5 5
101:  5 1 1
102:  5 1 2
103:  5 1 3
104:  5 1 4
105:  5 1 5
106:  5 2 1
107:  5 2 2
```

```
107:  5 2 2
108:  5 2 3
109:  5 2 4
110:  5 2 5
111:  5 3 1
112:  5 3 2
113:  5 3 3
114:  5 3 4
115:  5 3 5
116:  5 4 1
117:  5 4 2
118:  5 4 3
119:  5 4 4
120:  5 4 5
121:  5 5 1
122:  5 5 2
123:  5 5 3
124:  5 5 4
125:  5 5 5
```

```
        ______________SECOND_TASK________________

        Enter X:7
        Enter Y:-3

   THE RESULT :

(x + y)^7 =  + 1 * (x^0) * (y^7) - 7 * (x^1) * (y^6) + 21 * (x^2) * (y^5) - 35 * (x^3) * (y^4) + 35 * (x^4) * (y^3) - 21
 * (x^5) * (y^2) + 7 * (x^6) * (y^1) = 16384



Press any key to continue . . .
```

Висновок : виконуючи лабораторну роботу, я набула практичних вмінь та навичок при комп'ютерній реалізації комбінаторних задач.