Міністерство освіти і науки України

Нацональний університет «Львівська політехніка»

Кафедра систем штучного інтелекту

Лабораторна робота № 6

з дисципліни

**«Дискретна математика»**

Виконав:

Студент групи КН-114

**Кратко Денис**

Викладач:

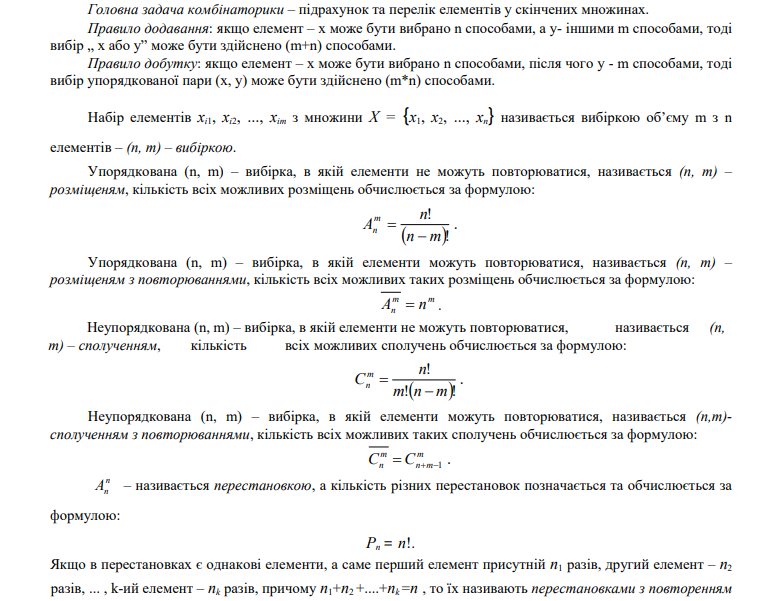
**Мельникова Н.І.**

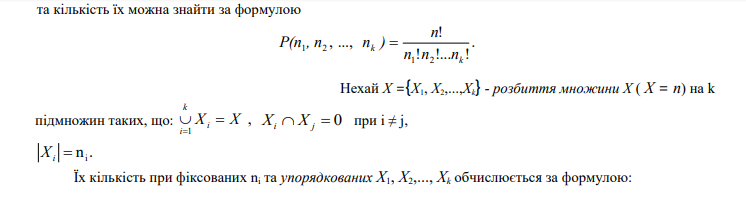
Львів – 2019

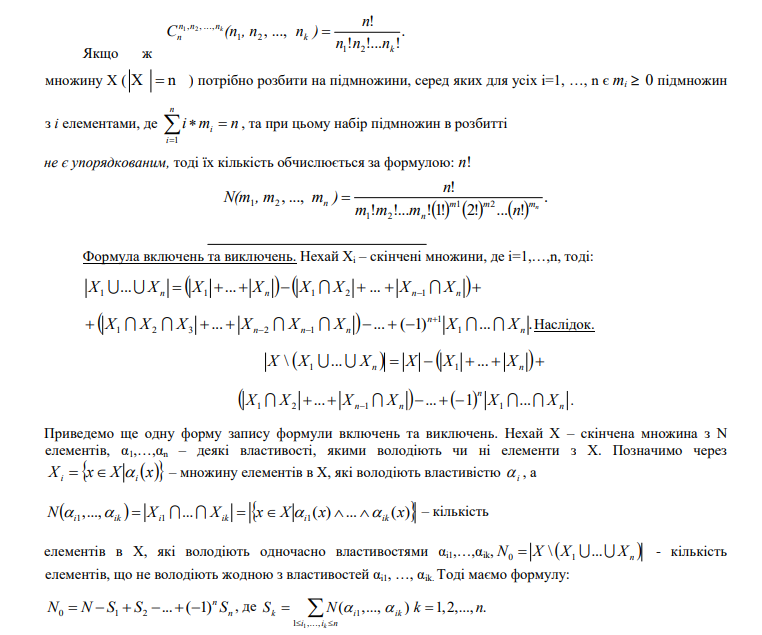
**Тема:** Генерація комбінаторних конфігурацій

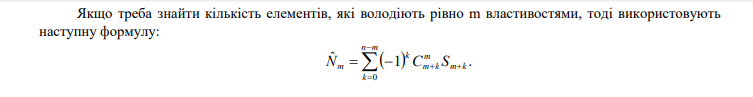
**Мета роботи:** набути практичних вмінь та навичок при комп’ютерній реалізації комбінаторних задач.

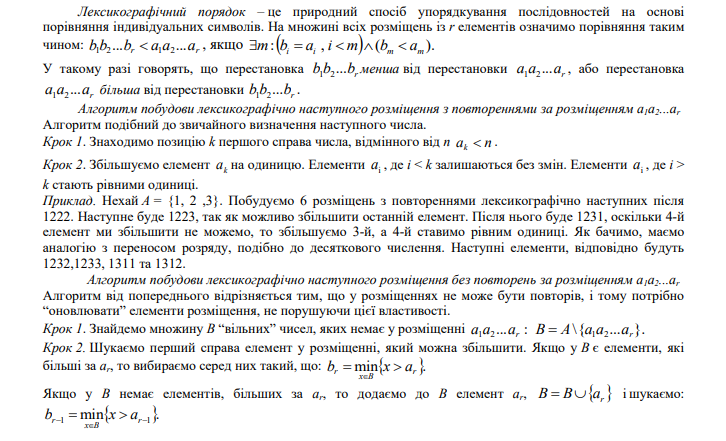
**Теоретичні відомості**

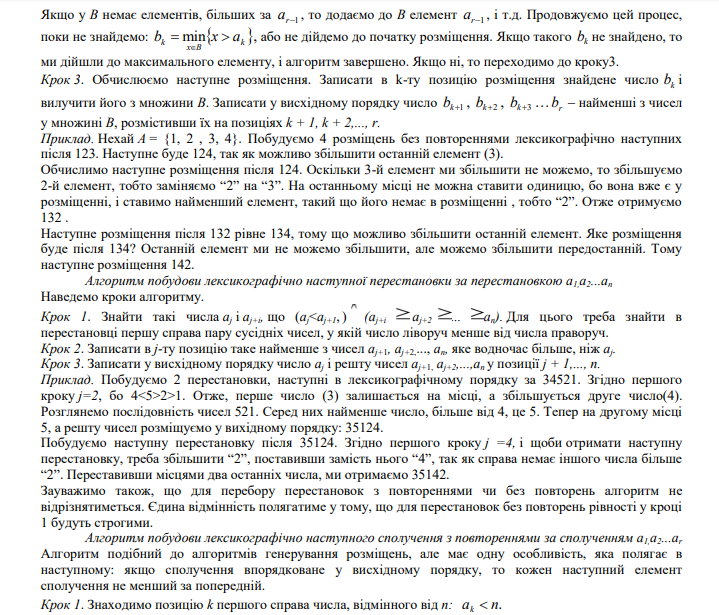


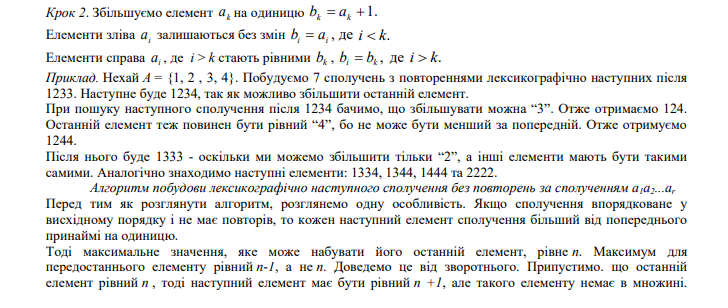


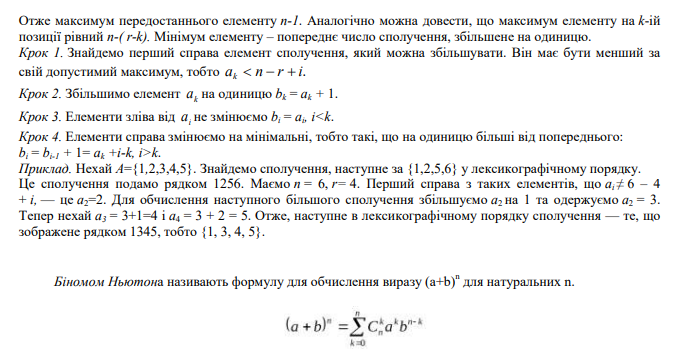










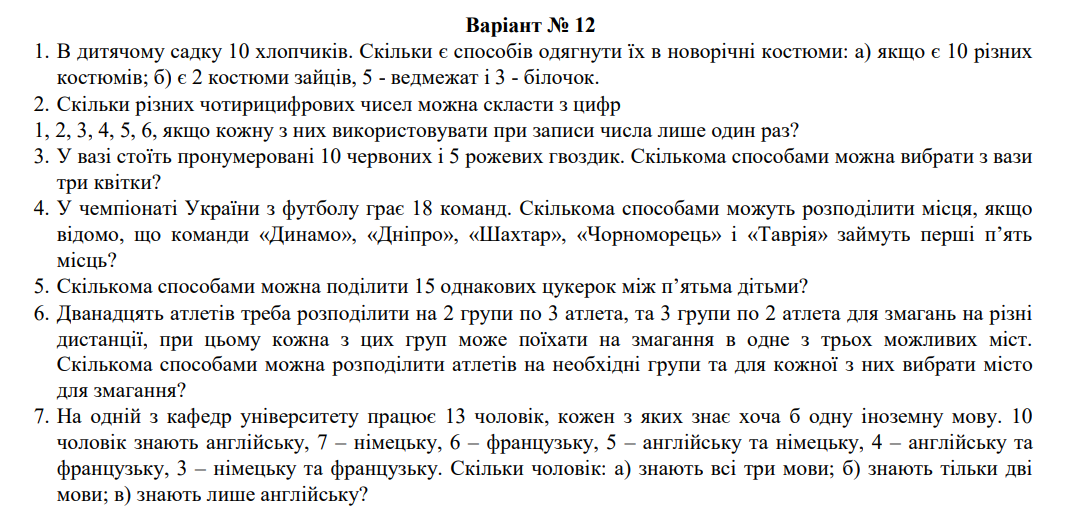


**Завдання (12 варіант)**

Додаток 1

Умова

Використовуючи теоретичні відомості, розв’язати наступні комбінаторні задачі за своїм варіантом.



Розв’язки

1. А)

Б)

2.

3.

4.

5. Кожній дитині має дістатись принаймні по цукерці. Спочатку роздамо по 1 цукерці кожному з 5 дітей, залишиться 10 цукерок. Тепер для кожної з 10 цукерок є 5 варіантів, якій дитині її дати. Отже, способів .

6.

7.

Додаток 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.

Задане додатне ціле число n. Розташувати у лексикографічному порядку всі перестановки множини {1, 2, …, n}. Побудувати розклад (a+b)10.

Розв’язок

#include <bits/stdc++.h>  
  
using namespace std;  
  
struct drib {  
 unsigned long int a;  
 unsigned long int b;  
};  
  
drib multiply\_drib\_by(drib d, unsigned long int a, unsigned long int b) {  
 d.a \*= a;  
 d.b \*= b;  
 unsigned long int m = max(d.a, d.b);  
 while (m > 0) {  
 if (!(d.a % m) && !(d.b % m)) {  
 d.a = d.a / m;  
 d.b = d.b / m;  
 }  
 m--;  
 };  
 return d;  
};  
  
void swap(unsigned long int\* a, unsigned long int\* b) {  
 unsigned long int temp = \*a;  
 \*a = \*b;  
 \*b = temp;  
};  
  
void my\_sort(unsigned long int\* arr, unsigned long int n) {  
 // bubblesort from https://www.geeksforgeeks.org/bubble-sort/  
 unsigned long int i, j;  
 for (i = 0; i < n-1; i++) {  
 // Last i elements are already in place  
 for (j = 0; j < n-i-1; j++) {  
 if (arr[j] > arr[j+1]) {  
 swap(arr + j, arr + (j+1));  
 };  
 };  
 };  
};  
  
bool next\_permutation(unsigned long int\* arr, unsigned long int n) {  
 if (n > 1) {  
 unsigned long int j, i, min\_i;  
 for (j = n - 2; (arr[j] >= arr[j + 1]); j--) {  
 if (j == 0) {  
 return false;  
 };  
 };  
 min\_i = j + 1;  
 for (i = j + 1; i < n; i++) {  
 if ((arr[i] < arr[min\_i]) && (arr[i] > arr[j])) {  
 min\_i = i;  
 };  
 };  
 swap(arr + j, arr + min\_i);  
 my\_sort(arr + (j + 1), n - (j + 1));  
 return true;  
 }  
 else {  
 return false;  
 };  
};  
  
void print\_array(unsigned long int\* arr, unsigned long int n, unsigned long int counter) {  
 cout << "Permutation " << counter << ": [";  
 if (n > 0) {  
 cout << arr[0];  
 for (unsigned long int i = 1; i < n; i++) {  
 cout << ", " << arr[i];  
 };  
 }  
 cout << "]" << endl;  
};  
  
void print\_binom(unsigned long int n) {  
 unsigned long int k;  
 drib c = {1, 1};  
 for (k = 0; k < n; k++) {  
 cout << "(" << c.a << "/" << c.b << ")";  
 cout << "\*(a^" << k << ")\*(b^" << (n - k) << ")";  
 cout << " + ";  
  
 c = multiply\_drib\_by(c, n - k, k + 1);  
 };  
  
 cout << "(" << c.a << "/" << c.b << ")";  
 cout << "\*(a^" << k << ")\*(b^" << (n - k) << ")";  
 cout << endl;  
};  
  
int main() {  
 unsigned long int i, n;  
 do {  
 cout << "Enter n=";  
 cin >> n;  
 } while (n == 0);  
 // 1 2 3 4 5 6 7 8 9 10  
 // ...  
 // 10 9 8 7 6 5 4 3 2 1  
 unsigned long int\* a = (unsigned long int\*)calloc(n, sizeof(unsigned long int));  
 for (i = 0; i < n; i++) {  
 a[i] = i + 1;  
 };  
  
 unsigned long int counter = 0;  
 do {  
 counter++;  
 print\_array(a, n, counter);  
 } while (next\_permutation(a, n));  
  
 free(a);  
  
 cout << endl << "(a + b)^10 = ";  
 print\_binom(10);  
  
 return 0;  
};

Результат виконання програми

1 випадок

"D:\Dyskretna laboratorni\6\cmake-build-debug\6.exe"  
Enter n=0  
Enter n=0  
Enter n=5  
Permutation 1: [1, 2, 3, 4, 5]  
Permutation 2: [1, 2, 3, 5, 4]  
Permutation 3: [1, 2, 4, 3, 5]  
Permutation 4: [1, 2, 4, 5, 3]  
Permutation 5: [1, 2, 5, 3, 4]  
Permutation 6: [1, 2, 5, 4, 3]  
Permutation 7: [1, 3, 2, 4, 5]  
Permutation 8: [1, 3, 2, 5, 4]  
Permutation 9: [1, 3, 4, 2, 5]  
Permutation 10: [1, 3, 4, 5, 2]  
Permutation 11: [1, 3, 5, 2, 4]  
Permutation 12: [1, 3, 5, 4, 2]  
Permutation 13: [1, 4, 2, 3, 5]  
Permutation 14: [1, 4, 2, 5, 3]  
Permutation 15: [1, 4, 3, 2, 5]  
Permutation 16: [1, 4, 3, 5, 2]  
Permutation 17: [1, 4, 5, 2, 3]  
Permutation 18: [1, 4, 5, 3, 2]  
Permutation 19: [1, 5, 2, 3, 4]  
Permutation 20: [1, 5, 2, 4, 3]  
Permutation 21: [1, 5, 3, 2, 4]  
Permutation 22: [1, 5, 3, 4, 2]  
Permutation 23: [1, 5, 4, 2, 3]  
Permutation 24: [1, 5, 4, 3, 2]  
Permutation 25: [2, 1, 3, 4, 5]  
Permutation 26: [2, 1, 3, 5, 4]  
Permutation 27: [2, 1, 4, 3, 5]  
Permutation 28: [2, 1, 4, 5, 3]  
Permutation 29: [2, 1, 5, 3, 4]  
Permutation 30: [2, 1, 5, 4, 3]  
Permutation 31: [2, 3, 1, 4, 5]  
Permutation 32: [2, 3, 1, 5, 4]  
Permutation 33: [2, 3, 4, 1, 5]  
Permutation 34: [2, 3, 4, 5, 1]  
Permutation 35: [2, 3, 5, 1, 4]  
Permutation 36: [2, 3, 5, 4, 1]  
Permutation 37: [2, 4, 1, 3, 5]  
Permutation 38: [2, 4, 1, 5, 3]  
Permutation 39: [2, 4, 3, 1, 5]  
Permutation 40: [2, 4, 3, 5, 1]  
Permutation 41: [2, 4, 5, 1, 3]  
Permutation 42: [2, 4, 5, 3, 1]  
Permutation 43: [2, 5, 1, 3, 4]  
Permutation 44: [2, 5, 1, 4, 3]  
Permutation 45: [2, 5, 3, 1, 4]  
Permutation 46: [2, 5, 3, 4, 1]  
Permutation 47: [2, 5, 4, 1, 3]  
Permutation 48: [2, 5, 4, 3, 1]  
Permutation 49: [3, 1, 2, 4, 5]  
Permutation 50: [3, 1, 2, 5, 4]  
Permutation 51: [3, 1, 4, 2, 5]  
Permutation 52: [3, 1, 4, 5, 2]  
Permutation 53: [3, 1, 5, 2, 4]  
Permutation 54: [3, 1, 5, 4, 2]  
Permutation 55: [3, 2, 1, 4, 5]  
Permutation 56: [3, 2, 1, 5, 4]  
Permutation 57: [3, 2, 4, 1, 5]  
Permutation 58: [3, 2, 4, 5, 1]  
Permutation 59: [3, 2, 5, 1, 4]  
Permutation 60: [3, 2, 5, 4, 1]  
Permutation 61: [3, 4, 1, 2, 5]  
Permutation 62: [3, 4, 1, 5, 2]  
Permutation 63: [3, 4, 2, 1, 5]  
Permutation 64: [3, 4, 2, 5, 1]  
Permutation 65: [3, 4, 5, 1, 2]  
Permutation 66: [3, 4, 5, 2, 1]  
Permutation 67: [3, 5, 1, 2, 4]  
Permutation 68: [3, 5, 1, 4, 2]  
Permutation 69: [3, 5, 2, 1, 4]  
Permutation 70: [3, 5, 2, 4, 1]  
Permutation 71: [3, 5, 4, 1, 2]  
Permutation 72: [3, 5, 4, 2, 1]  
Permutation 73: [4, 1, 2, 3, 5]  
Permutation 74: [4, 1, 2, 5, 3]  
Permutation 75: [4, 1, 3, 2, 5]  
Permutation 76: [4, 1, 3, 5, 2]  
Permutation 77: [4, 1, 5, 2, 3]  
Permutation 78: [4, 1, 5, 3, 2]  
Permutation 79: [4, 2, 1, 3, 5]  
Permutation 80: [4, 2, 1, 5, 3]  
Permutation 81: [4, 2, 3, 1, 5]  
Permutation 82: [4, 2, 3, 5, 1]  
Permutation 83: [4, 2, 5, 1, 3]  
Permutation 84: [4, 2, 5, 3, 1]  
Permutation 85: [4, 3, 1, 2, 5]  
Permutation 86: [4, 3, 1, 5, 2]  
Permutation 87: [4, 3, 2, 1, 5]  
Permutation 88: [4, 3, 2, 5, 1]  
Permutation 89: [4, 3, 5, 1, 2]  
Permutation 90: [4, 3, 5, 2, 1]  
Permutation 91: [4, 5, 1, 2, 3]  
Permutation 92: [4, 5, 1, 3, 2]  
Permutation 93: [4, 5, 2, 1, 3]  
Permutation 94: [4, 5, 2, 3, 1]  
Permutation 95: [4, 5, 3, 1, 2]  
Permutation 96: [4, 5, 3, 2, 1]  
Permutation 97: [5, 1, 2, 3, 4]  
Permutation 98: [5, 1, 2, 4, 3]  
Permutation 99: [5, 1, 3, 2, 4]  
Permutation 100: [5, 1, 3, 4, 2]  
Permutation 101: [5, 1, 4, 2, 3]  
Permutation 102: [5, 1, 4, 3, 2]  
Permutation 103: [5, 2, 1, 3, 4]  
Permutation 104: [5, 2, 1, 4, 3]  
Permutation 105: [5, 2, 3, 1, 4]  
Permutation 106: [5, 2, 3, 4, 1]  
Permutation 107: [5, 2, 4, 1, 3]  
Permutation 108: [5, 2, 4, 3, 1]  
Permutation 109: [5, 3, 1, 2, 4]  
Permutation 110: [5, 3, 1, 4, 2]  
Permutation 111: [5, 3, 2, 1, 4]  
Permutation 112: [5, 3, 2, 4, 1]  
Permutation 113: [5, 3, 4, 1, 2]  
Permutation 114: [5, 3, 4, 2, 1]  
Permutation 115: [5, 4, 1, 2, 3]  
Permutation 116: [5, 4, 1, 3, 2]  
Permutation 117: [5, 4, 2, 1, 3]  
Permutation 118: [5, 4, 2, 3, 1]  
Permutation 119: [5, 4, 3, 1, 2]  
Permutation 120: [5, 4, 3, 2, 1]  
  
(a + b)^10 = (1/1)\*(a^0)\*(b^10) + (10/1)\*(a^1)\*(b^9) + (45/1)\*(a^2)\*(b^8) + (120/1)\*(a^3)\*(b^7) + (210/1)\*(a^4)\*(b^6) +  
(252/1)\*(a^5)\*(b^5) + (210/1)\*(a^6)\*(b^4) + (120/1)\*(a^7)\*(b^3) + (45/1)\*(a^8)\*(b^2) + (10/1)\*(a^9)\*(b^1) + (1/1)\*(a^10)  
\*(b^0)  
  
Process finished with exit code 0

2 випадок

"D:\Dyskretna laboratorni\6\cmake-build-debug\6.exe"  
Enter n=1  
Permutation 1: [1]  
  
(a + b)^10 = (1/1)\*(a^0)\*(b^10) + (10/1)\*(a^1)\*(b^9) + (45/1)\*(a^2)\*(b^8) + (120/1)\*(a^3)\*(b^7) + (210/1)\*(a^4)\*(b^6) +  
(252/1)\*(a^5)\*(b^5) + (210/1)\*(a^6)\*(b^4) + (120/1)\*(a^7)\*(b^3) + (45/1)\*(a^8)\*(b^2) + (10/1)\*(a^9)\*(b^1) + (1/1)\*(a^10)  
\*(b^0)  
  
Process finished with exit code 0

3 випадок

"D:\Dyskretna laboratorni\6\cmake-build-debug\6.exe"  
Enter n=2  
Permutation 1: [1, 2]  
Permutation 2: [2, 1]  
  
(a + b)^10 = (1/1)\*(a^0)\*(b^10) + (10/1)\*(a^1)\*(b^9) + (45/1)\*(a^2)\*(b^8) + (120/1)\*(a^3)\*(b^7) + (210/1)\*(a^4)\*(b^6) +  
(252/1)\*(a^5)\*(b^5) + (210/1)\*(a^6)\*(b^4) + (120/1)\*(a^7)\*(b^3) + (45/1)\*(a^8)\*(b^2) + (10/1)\*(a^9)\*(b^1) + (1/1)\*(a^10)  
\*(b^0)  
  
Process finished with exit code 0

4 випадок

"D:\Dyskretna laboratorni\6\cmake-build-debug\6.exe"  
Enter n=4  
Permutation 1: [1, 2, 3, 4]  
Permutation 2: [1, 2, 4, 3]  
Permutation 3: [1, 3, 2, 4]  
Permutation 4: [1, 3, 4, 2]  
Permutation 5: [1, 4, 2, 3]  
Permutation 6: [1, 4, 3, 2]  
Permutation 7: [2, 1, 3, 4]  
Permutation 8: [2, 1, 4, 3]  
Permutation 9: [2, 3, 1, 4]  
Permutation 10: [2, 3, 4, 1]  
Permutation 11: [2, 4, 1, 3]  
Permutation 12: [2, 4, 3, 1]  
Permutation 13: [3, 1, 2, 4]  
Permutation 14: [3, 1, 4, 2]  
Permutation 15: [3, 2, 1, 4]  
Permutation 16: [3, 2, 4, 1]  
Permutation 17: [3, 4, 1, 2]  
Permutation 18: [3, 4, 2, 1]  
Permutation 19: [4, 1, 2, 3]  
Permutation 20: [4, 1, 3, 2]  
Permutation 21: [4, 2, 1, 3]  
Permutation 22: [4, 2, 3, 1]  
Permutation 23: [4, 3, 1, 2]  
Permutation 24: [4, 3, 2, 1]  
  
(a + b)^10 = (1/1)\*(a^0)\*(b^10) + (10/1)\*(a^1)\*(b^9) + (45/1)\*(a^2)\*(b^8) + (120/1)\*(a^3)\*(b^7) + (210/1)\*(a^4)\*(b^6) +  
(252/1)\*(a^5)\*(b^5) + (210/1)\*(a^6)\*(b^4) + (120/1)\*(a^7)\*(b^3) + (45/1)\*(a^8)\*(b^2) + (10/1)\*(a^9)\*(b^1) + (1/1)\*(a^10)  
\*(b^0)  
  
Process finished with exit code 0