

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ НАЦІОНАЛЬНИЙ
УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”**

Кафедра систем штучного інтелекту

Лабораторна робота №6

з дисципліни

“Дискретна математика”

Виконав:

студент групи КН-109

Коржов Володимир

Викладач:

Мельникова Н.І.

Генерація комбінаторних конфігурацій

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

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

Варіант №7

1. Учасники шахового турніру грають у залі, де є 8 столів. Скількома способами можна розмістити 16 шахістів, якщо учасники всіх партій відомі?

Якщо учасники всіх партій нам відомі, то в нас є 8 пар, які потрібно розмістити за 8 столів, а це $8! = 40320$ способів.

2. Скільки трицифрових чисел можна утворити з дев'яти цифр 1, 2, 3, 4, 5, 6, 7, 8, 9?

Це є розміщення з повторенням, тобто є $9^3 = 729$ чисел.

3. Скільки можна побудувати різних прямокутних паралелепіпедів, довжини ребер яких виражають натуральними числами від 1 до 10?

В прямокутному паралелепіпеді є всього 12 ребер, 3 з них можуть мати різні значення. Тоді буде $\frac{3^{10} \cdot 4}{12} = 19\,683$ прямокутних паралелепіпедів.

4. У вищій лізі чемпіонату України з футболу грають 16 команд. Скільки існує способів розподілення I, II, та III місця та вибору двох команд які перейдуть у першу лігу (дві останні команди)?

На I місце можуть поставити 16 команд, на II – 15, на III – 14.

Тобто ми маємо $16 \cdot 15 \cdot 14 = 3360$ способів для розподілення 3 перших місць. Передостаннє місце можуть посісти 13, а останнє 12 команд, тобто

$13 \cdot 12 = 156$ способів. І разом ми маємо $3360 \cdot 156 = 524160$ способів розподілення трьох перших та двох останніх команд.

5. З цифр 1, 2, 3, 4, 5, 6, 7, 8, 9 утворюють різні п'ятицифрові числа, що не мають однакових цифр. Визначити кількість чисел, у яких зустрічається цифри 5, 3, 4 одночасно, якщо вони не стоять поруч?

Якщо 5, 3, 4 не стоять поруч, то вони стоять на індексах 0, 2 і 4. На 0 може бути 3 цифри(5, 3, 4), на 2 – дві цифри і на 4 одна. Це залишає нам 1 та 3 індекси. На індексу 1 може бути $9 - 3 = 6$ цифр і на 3 – $6 - 1 = 5$. Отож, в нас виходить $3 \cdot 6 \cdot 2 \cdot 5 \cdot 1 = 180$ цифр.

6. У шаховому турнірі беруть участь 18 шахістів. Визначити кількість різних розкладів першого туру (розклади вважаються різними, якщо вони відрізняються учасниками, колір та номер столу не враховується).

18! різних розкладів турніру.

7. Знайти кількість цілих додатних чисел, які змінюються від 101 до 1000 та діляться рівно на два з чисел 3, 6 і 7.

Таких чисел є 151.

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 № 7

Визначити лексикографічно наступну перестановку для кожної з перестановок: 1432, 54123, 12453, 45231, 6714235, і 31528764. Побудувати розклад $(x - y)^8$.

Код програми :

```
#include <stdio.h>
```

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

```

int fact(int n)
{
    if(n == 0 || n == 1)
    {
        return 1;
    }
    else
    {
        return n*fact(n-1);
    }
}

int F(int k, int l)
{
    return fact(k)/(fact(l)*fact(k-l));
}

void swap(int *x, int *y)
{
    int temp;
    temp = *x;
    *x = *y;
    *y = temp;
}

void lexSwap(int len, int a[])
{
    int i, k, j, counter = 0;
    for(i = len-1; i >= 0; i--)
    {
        if(a[i] > a[i-1])
        {
            for(k = i+1; k < len; k++)
            {
                if(a[i] > a[k])
                {
                    counter++;
                }
            }
        }
    }
}

```

```

        }
    }
    if(counter != len - i - 1)
    {
        counter = 0;
    }
    else
    {
        break;
    }
}
}
}
if(counter == len - i - 1)
{
    swap(&a[i], &a[i-1]);
    for(k = i; k < len; k++)
    {
        for( j = k+1; j < len; j++)
        {
            if(a[k] > a[j])
            {
                swap(&a[k], &a[j]);
            }
        }
    }
}
}
}

int main()
{
    int len1 = 4, len2 = 5, len3 = 5, len4 = 5, len5 = 7, len6 = 8, n = 8, result =
0, m, j, i, x, y;
    int a1[] = {1, 4, 3, 2};
    int a2[] = {5, 4, 1, 2, 3};
    int a3[] = {1, 2, 4, 5, 3};
    int a4[] = {4, 5, 2, 3, 1};

```

```
int a5[] = {6, 7, 1, 4, 2, 3, 5};
int a6[] = {3, 1, 5, 2, 8, 7, 6, 4};
printf("Set 1 is: \n");
for(int k = 0; k < len1; k++)
{
printf(" %d ", a1[k]);
}printf("\n");
printf("Set 2 is: \n");
for(int k = 0; k < len2; k++)
{
printf(" %d ", a2[k]);
}printf("\n");
printf("Set 3 is: \n");
for(int k = 0; k < len3; k++)
{
printf(" %d ", a3[k]);
}printf("\n");
printf("Set 4 is: \n");
for(int k = 0; k < len4; k++)
{
printf(" %d ", a4[k]);
}printf("\n");
printf("Set 5 is: \n");
for(int k = 0; k < len5; k++)
{
printf(" %d ", a5[k]);
}printf("\n");
printf("Set 6 is: \n");
for(int k = 0; k < len6; k++)
{
printf(" %d ", a6[k]);
}printf("\n");
lexSwap(len1, a1);
lexSwap(len2, a2);
lexSwap(len3, a3);
```

```

lexSwap(len4, a4);
lexSwap(len5, a5);
lexSwap(len6, a6);
printf("The next lexicographic permutation for set 1 is:\n");
for(i = 0; i < len1; i++)
{
printf(" %d ", a1[i]);
}printf("\n");
printf("The next lexicographic permutation for set 2 is:\n");
for(i = 0; i < len2; i++)
{
printf(" %d ", a2[i]);
}printf("\n");
printf("The next lexicographic permutation for set 3 is:\n");
for(i = 0; i < len3; i++)
{
printf(" %d ", a3[i]);
}printf("\n");
printf("The next lexicographic permutation for set 4 is:\n");
for(i = 0; i < len4; i++)
{
printf(" %d ", a4[i]);
}printf("\n");
printf("The next lexicographic permutation for set 5 is:\n");
for(i = 0; i < len5; i++)
{
printf(" %d ", a5[i]);
}printf("\n");
printf("The next lexicographic permutation for set 6 is:\n");
for(i = 0; i < len6; i++)
{
printf(" %d ", a6[i]);
}printf("\n");
//this is for the binomial theorem
printf("Enter a value for x: \n");

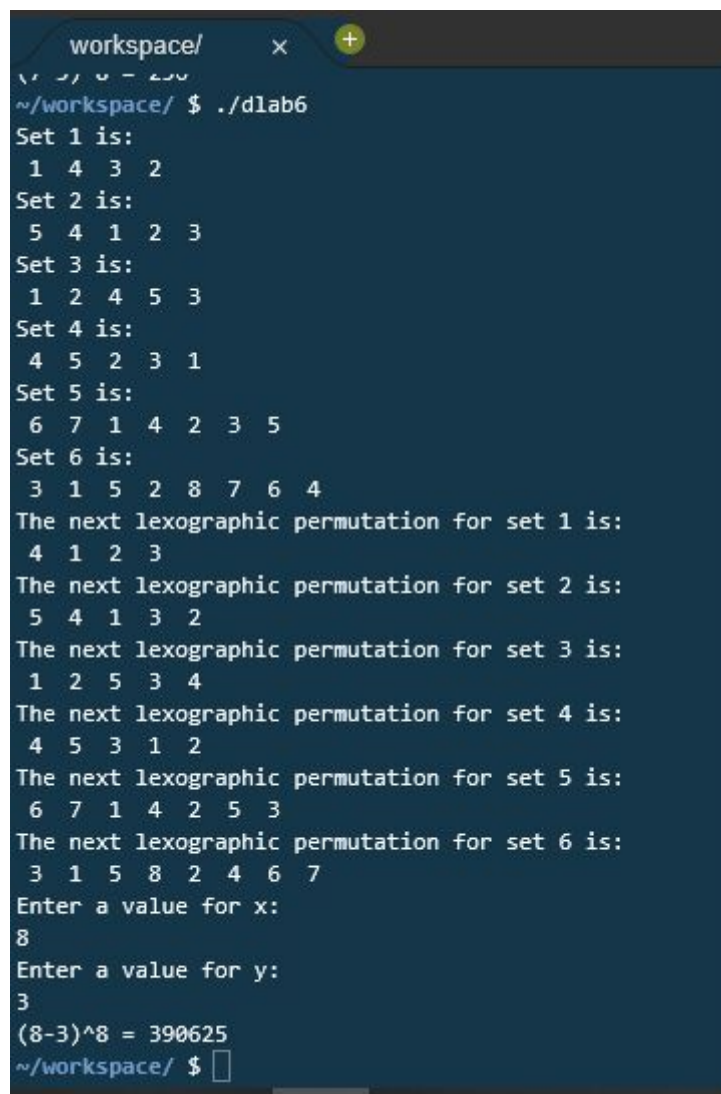
```

```

scanf("%d", &x);
printf("Enter a value for y: \n");
scanf("%d", &y);
y *= -1;
for(j = 0, m = n, i = 0; i <= n; i++, m--, j++)
{
result += F(n, j)*pow(x, m)*pow(y, j);
}
printf("(%d%d)^8 = %d\n", x, y, result);
return 0;
}

```

Скріншот роботи програми:



```

workspace/ x +
~/workspace/ $ ./dlab6
Set 1 is:
1 4 3 2
Set 2 is:
5 4 1 2 3
Set 3 is:
1 2 4 5 3
Set 4 is:
4 5 2 3 1
Set 5 is:
6 7 1 4 2 3 5
Set 6 is:
3 1 5 2 8 7 6 4
The next lexicographic permutation for set 1 is:
4 1 2 3
The next lexicographic permutation for set 2 is:
5 4 1 3 2
The next lexicographic permutation for set 3 is:
1 2 5 3 4
The next lexicographic permutation for set 4 is:
4 5 3 1 2
The next lexicographic permutation for set 5 is:
6 7 1 4 2 5 3
The next lexicographic permutation for set 6 is:
3 1 5 8 2 4 6 7
Enter a value for x:
8
Enter a value for y:
3
(8-3)^8 = 390625
~/workspace/ $

```


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