# PROIECT NR.2:ЭЛЕМЕНТЫ КОМБИНАТОРИКИ

Выполнила: Ilona FRASINIUC, gr. IS11Z

Проверил: Vitalie TICAU, lector universitar

## 1. Постановка задачи:

Вывести таблицы операций:

1. Факториал
2. Перестановки
3. Размещения
4. Сочетания
5. Степень
6. Степень рекурсивной функции
7. Перестановки с повторениями
8. Размещения с повторениями
9. Сочетания с повторениями
10. Перестановки с повторениями (2 способ)

## 2. Выбор методики:

Для выполнения данного проекта был использовал язык C++, в программе Visual Studio Code.

**3. Предварительные исследования:**

1. Элементы комбинаторики:

***Факториал / перестановки:***

С точки зрения комбинаторики, перестановка - это способ последовательного расположения его элементов. Другими словами, количество изменений, которые могут быть сделаны между элементами набора.

***P(n) = n!***

***Размещения:***

Упорядоченные множества, которые формируются из элементов любого подмножества конечного множества А, называются упорядоченными подмножествами А или размещениями.

***Сочетания:***

Сочетания означает количество неупорядоченных подмножеств множества из n элементов, в каждом из этих подмножеств по k элементов.

2. Повторяющиеся элементы комбинаторики:

***Перестановки с повторениями:***

***‾P(n) = nn***

***Размещения с повторениями:***

***Сочетания с повторениями:***

**4. Программа**

*#include* <iostream>

*#include* <stdio.h>

*#include* <string.h>

using namespace std;

unsigned long long Factorial(unsigned);

unsigned long long Permutari(unsigned);

unsigned long long Aranjamente(unsigned, unsigned);

unsigned long long Combinari(unsigned, unsigned);

unsigned long long Power(unsigned, unsigned);

unsigned long long PowerRec(unsigned, unsigned);

unsigned long long PermutariCuRepetitii(unsigned);

unsigned long long AranjamenteCuRepetitii(unsigned, unsigned);

unsigned long long CombinariCuRepetitii(unsigned, unsigned);

unsigned long long PermutariCuRepetitii2(unsigned, unsigned[]);

int main()

{

    cout << "Proiect nr.2" << endl;

    cout << "Tema: Elemente de combinatorica" << endl;

    cout << "Elaborat de Smoleac Mihai" << endl;

    unsigned a[10] = {1, 3, 2, 0, 0};

    char sir[100] = "Pr(", Sir[100];

    unsigned n = 6, k = 2, m;

    cout << "Numarul total de elemente al multimii: ";

    cin >> n;

*do*

    {

        cout << "Numarul de elemente al grupului (cate elemente luam in grup): ";

        cin >> k;

    } *while* (k > n || k < 0);

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

    {

        a[i] = 0;

    }

    cout << "Introduceti " << n << " elemente: " << endl;

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

    {

        cin >> m;

        a[m - 1]++;

    }

    cout << n << "^" << k << "=" << PowerRec(n, k) << endl;

    cout << n << "!=" << Factorial(n) << endl;

    cout << "P(" << n << ")"

         << "=" << Permutari(n) << endl;

    cout << "A(" << n << "," << k << ")"

         << "=" << Aranjamente(n, k) << endl;

    cout << "C(" << n << "," << k << ")"

         << "=" << Combinari(n, k) << endl;

    cout << "Pr(" << n << ")"

         << "=" << PermutariCuRepetitii(n) << endl;

    cout << "Ar(" << n << ")"

         << "=" << AranjamenteCuRepetitii(n, k) << endl;

    cout << "Cr(" << n << ")"

         << "=" << CombinariCuRepetitii(n, k) << endl;

    int d = 0;

*while* (a[d] == 0)

    {

        d++;

    }

    sprintf(Sir, "%s%u", sir, a[d]);

    strcpy(sir, Sir);

*for* (int i = d + 1; i < n; i++)

    {

*if* (a[i] != 0)

        {

            sprintf(Sir, "%s%c%u", sir, ',', a[i]);

        }

        strcpy(sir, Sir);

    }

    m = strlen(sir);

    strcpy(&sir[1], &sir[m]);

*//sir[strlen(sir) - 1] = '\0';*

    cout << Sir << ") = " << PermutariCuRepetitii2(n, a);

*return* 0;

}

unsigned long long Factorial(unsigned m)

{

    unsigned long long f = 1;

*for* (int i = 2; i <= m; i++)

    {

        f \*= i;

    }

*return* f;

}

unsigned long long Permutari(unsigned n)

{

*return* Factorial(n);

}

unsigned long long Aranjamente(unsigned n, unsigned k)

{

*return* (Factorial(n) / Factorial(n - k));

}

unsigned long long Combinari(unsigned n, unsigned k)

{

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

}

unsigned long long Power(unsigned n, unsigned k)

{

    unsigned long long p = 1;

*for* (int i = 1; i <= k; i++)

    {

        p = p \* n;

    }

*return* p;

}

unsigned long long PowerRec(unsigned n, unsigned k)

{

    unsigned long long p = 1;

*if* (k <= 0)

*return* 1;

*return* (PowerRec(n, k - 1) \* n);

}

unsigned long long PermutariCuRepetitii(unsigned n)

{

*return* Power(n, n);

}

unsigned long long AranjamenteCuRepetitii(unsigned n, unsigned k)

{

*return* Power(n, k);

}

unsigned long long CombinariCuRepetitii(unsigned n, unsigned k)

{

*return* (Factorial(n + k - 1) / (Factorial(k) \* Factorial(n - 1)));

}

unsigned long long PermutariCuRepetitii2(unsigned size, unsigned a[])

{

    unsigned long long p = 1;

    unsigned s = 0;

*for* (int i = 0; i < size; i++)

    {

        s += a[i];

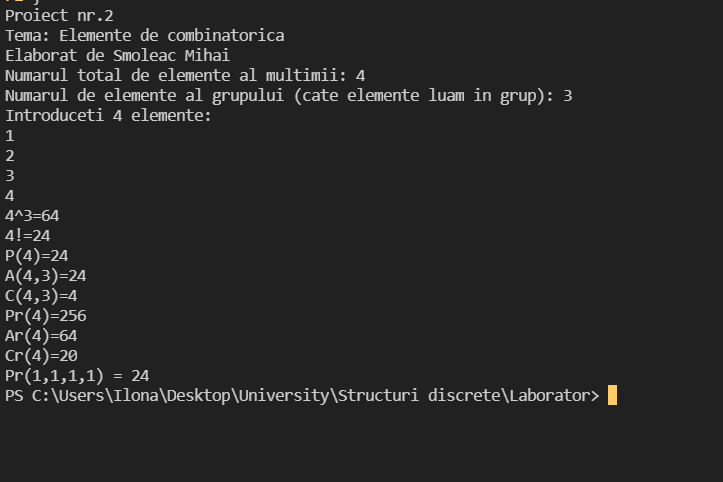
        p \*= Factorial(a[i]);

    }

*return* Factorial(s) / p;

}

**5. Результаты:**



**6. Вывод:**

В результате этой лабораторной работы, я заметила, что результаты факториала и перестановок идентичны.