# PROIECT NR.1:Комбинаторика

Выполнил: Adam Denis , gr. IS11Z(sub.2)

Проверил: Vitalie TICAU, lector universitar

## 1. Постановка задачи

**1. Выполнить следующие вычисления:**

1. Вычислить факториал числа, обозначающего количество элементов множества (n!)  
   Вычислить факториал числа, обозначающего количество элементов групп (k!)  
   Оба числа должны быть положительными, и k должно быть меньше n.
2. Вычислить, чему равна перемещение из n цифр (P(n)) [неповторяющихся]
3. Вычислить, чему равно размещение из n цифр по k (A(n, k)) [неповторяющихся]
4. Вычислить, чему равно сочетание из n цифр по k (C(n, k)) [неповторяющихся]
5. Вычислить, чему равна перемещение из n цифр (P(n)) [повторяющихся]
6. Вычислить, чему равно размещение из n цифр по k (A(n, k)) [повторяющихся]
7. Вычислить, чему равно сочетание из n цифр по k (C(n, k)) [повторяющихся]
8. Вычислить, чему равна перемещение из n цифр (используются 3 двойки, 1 четверка, 1 пятерка) [повторяющихся]

**2. Выбор методики**

Вычисления по теме комбинаторики были выполнены на языке программирования C++, с помощью Code Blocks.

Сперва вычисляем факториал двух введенных чисел, согласно формуле:

Далее, для каждого последующего вычисления применялась формула, исходя из первой, применяя ее в последующих вычислениях

## 3. Программа

## #include <iostream>

## #include <cmath>

## using namespace std;

## long unsigned Factorial (int n);

## long unsigned Permutari (int n);

## long unsigned Aranjamente (int n, int k);

## long unsigned Combinari (int n, int k);

## long unsigned PermutariRepetitii (int n);

## long unsigned PermutariRepetitii1 (int n, int x[]);

## long unsigned AranjamenteRepetitii (int n, int k);

## long unsigned CombinariRepetitii (int n, int k);

## int main()

## {

## 

## cout<<"proekt nr1, elaboorat de Adam Denis" << endl;

## cout<<"tema combinatorica\n";

## int n, k, ad;

## int x[20] = {3, 2, 1};

## do {

## cout <<"tastati numarul de elemente ale multimii (numar natural): -> ";

## cin >> n;

## } while (n<0);

## do {

## cout <<"tastati numarul de elemente din grupuri (numar natural): -> ";

## cin >> k;

## } while ((k<0) ||(k>n));

## cout << "tastati " << n << " cifre utilizate pentru permutari: ";

## for(int i=0; i<n; i++) x[i] = 0;

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

## {

## cin >> ad;

## x[ad-1]++;

## }

## //for(int i=0; i<n; i++) cout << x[i];

## /\*if ((k>n) || (n<0) || (k<0)) {

## cout <<"neverno";

## } else {\*/

## //for(int i=1; i<=n; i++)

## cout << "1.\t" << n << "! = " << Factorial (n) << endl;

## cout << "\t" << k << "! = " << Factorial (k) << endl;

## cout << "2.\tP(" << n << ") = " << Permutari (n) << endl;

## cout << "3.\tA(" << n <<", " << k << ") = " << Aranjamente (n, k) << endl;

## cout << "4.\tC(" << n <<", " << k << ") = " << Combinari (n, k) << endl;

## cout << "5.\tPr(" << n <<", " << k << ") = " << PermutariRepetitii (n) << endl;

## cout << "6.\tAr(" << n <<", " << k << ") = " << AranjamenteRepetitii (n, k) << endl;

## cout << "7.\tCr(" << n <<", " << k << ") = " << CombinariRepetitii (n, k) << endl;

## cout << "8.\tPr(";

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

## cout << x[i] << ", ";

## cout << "\b\b) = " << PermutariRepetitii1 (n, x) << endl;

## return 0;

## //}

## }

## long unsigned Factorial (int n)

## {

## long unsigned f=1 ;

## for(int i=1; i<=n; i++)

## f \*= i;

## return f;

## }

## long unsigned Permutari (int n)

## {

## //cisla iz 1, 2, 3, nepovtor tifri

## //123, 132, 213, 231, 312, 321,

## return Factorial(n);

## }

## long unsigned Aranjamente (int n, int k)

## {

## //cisla iz 1, 2, 3, nepovtor tifri po 2

## //12, 13, 21, 23, 31, 32

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

## 

## }

## long unsigned Combinari (int n, int k)

## {

## //cisla iz 1, 2, 3, nepovtor tifri po 2

## //1,2, 1,3, 2,3,

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

## 

## }

## long unsigned PermutariRepetitii (int n)

## {

## //cisla iz 1, 2, 3, nepovtor tifri

## //111, 112, 113, 121, 122, 123, 131, 132, 133,

## //211, 112, 113, 121, 122, 123, 131, 132, 133,

## //311, 112, 113, 121, 122, 123, 131, 132, 133,

## return pow(n,n);

## }

## long unsigned PermutariRepetitii1 (int n, int x[])

## {

## //cisla iz 3 edinit, dvux 2 i odnoi 3

## //111223, 111232

## long unsigned s=0, p=1;

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

## {

## s += x[i];

## p \*= Factorial(x[i]);

## }

## return Factorial(s)/p;

## }

## long unsigned AranjamenteRepetitii (int n, int k)

## {

## //dvuxznacnii cisla sostavleni s povtoreusimi tiframi 1, 2, 3

## //11, 12, 13, 21,22, 23, 31, 32, 33

## return pow(n,k);

## }

## long unsigned CombinariRepetitii (int n, int k)

## {

## // cisla iz 1, 2, 3 povtor tifri po 2

## // 1, 1; 1, 2; 1, 3; 2, 2; 2, 3; 3, 3;

## 

## return Combinari(n+k-1, k);

## }

## 4. Результаты и анализ

proekt nr1, elaboorat de Adam Denis

tema combinatorica

tastati numarul de elemente ale multimii (numar natural): -> 6

tastati numarul de elemente din grupuri (numar natural): -> 2

tastati 6 cifre utilizate pentru permutari: 1 1 1 2 3 3

1. 6! = 720

2! = 2

2. P(6) = 720

3. A(6, 2) = 30

4. C(6, 2) = 15

5. Pr(6, 2) = 46656

6. Ar(6, 2) = 36

7. Cr(6, 2) = 21

8. Pr(3, 1, 2, 0, 0, 0) = 60

Press any key to continue.

## 5.Выводы

С помощью комбинаторики мы можем узнать количество различных вариаций перестановки, перемещения и сочетания,

Применяя CodeBlocks и тип longunsigned можно вычислить максимум 20!