**Анализ на решението на задача  
C#. Пермутации**

Още с прочитането на поредното число, разместваме цифрите му така, че да са подредени в намаляващ ред. В тази връзка е удобно числото да се прочете като низ и за сортиране на знаците в низа да се използват стандартни средства за сортиране (функциата sort от библиотеката algorithm). Низа преобразуваме в число и го записваме в на поредно място в масив. Сортираме елементите на масива по големина. По този начин еднаквите числа, т.е. тези които чрез разместване на цифрите си се трансформират до някое от другите прочетени, се оказват съседни. За да достигнем до отговора на задачата преброяваме еднаквите съседни числа в масива

Сорс-кодът на програмата:

#include <iostream>

#include <cstring>

#include <algorithm>

**using namespace std**;

**typedef unsigned long long int** ull;

**int** main () {

**int** n,i,j,count\_equals=0;

**char** num[22];

**cin** >> n;

ull \*A = **new** ull[n];

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

**cin** >> num;

**sort**(num, num+(j=strlen(num)));

**for** (A[i]=num[--j]-'0'; --j>=0; A[i]=A[i]\*10+(num[j]-'0'));

}

**sort**(A,A+n);

**for** (j=0,i=1; i<n; i++)

**if** (A[i]==A[i-1]) count\_equals++;

**else if** (count\_equals) j+=++count\_equals,count\_equals=0;

**if** (count\_equals) j+=++count\_equals;

**delete** [] A;

**cout** << j << **endl**;

**return** 0;

}

*Автор: Евгений Василев*