**Анализ на решението на задача  
D#. Сума от цифрите на n!**

За да намерим сумата от цифрите на *n*! очевидно трябва да изчислим *n*! ☺ Проблемът е, че факториел е много бързо растяща функция и резултатът при големи *n* (*n*>20) излиза извън диапазона дори на „най-широкия“ целочислен тип. Налага се да реализираме някакъв вариант на аритметика с „дълги“ числа.

В предложения от автора вариант, резултатът се представя чрез масив от цели числа – всяка цифра на факториела се записва в отделен елемент на масива. Операцията умножение с числото *i* се реализира като последователно умножаваме всяка цифра на дългото число (представящо факториела) с *i* с отчитане на пренос (CF) към по-старши разряд в дългото число. Съобразено е, че във факториела (за n>4) има няколко младши нули в записа му и при реализиране на умножението може да се пропусне умножаването им с числото *i*. Изследване на функцията факториел показва, че за големи *n* броят цифри на факториела клони към *n*.lg*n*, a броят последователни нули в най-младшите разряди е около 9-10 % от всички цифри.

Сорс-кодът на програмата:

#include <iostream>

**char** A[40000]={1};**/// цифрите на n! (първоначално съдържа цифрите на 1! )**

**int** len=1, **/// брой на цифрите в n!**

i=2,

j,

cf, **/// пренос при операцията умножение**

n,

tr0; **/// брой на „опашковите“ (младши) нули в n!**

**int** main () {

**for** (**std**::**cin** >> n; i<=n; i++) { **/// Изчислява n!**

**/// умножава цифрите на резултата с i и очита преноса в по-старши разряд**

**for** (j=tr0; j<len; A[j++]=cf%10, cf/=10) cf+=A[j]\*i;

**/// добавя старши цифри в резултата ако е останал пренос от последното  
 /// умножение**

**for** ( ; cf; cf/=10) A[len++]=cf%10;

**/// отчита новопоявили се опашкови нули при умножението**

**while** (A[tr0]==0) tr0++;

}

**for** (i=j=0; i<len; j+=A[i++]); **/// j = сума от цифрите на n!**

**std**::**cout** << j << '\n';

**return** 0;

}

*Автор: Евгений Василев*