**СЛУЧАЙНИ ЧИСЛА**

**СЪЗДАВАНЕ НА ТЕСТОВИ ДАННИ**

**1. Стартиране в конзолен прозорец**

**а/ отваряне на конзолен прозорец**

Отваряме конзолния прозорец по следния начин: Start -> Run -> cmd

С команда cd правим текуща директорията, в която е изпълнимия файл. Например, ако изпълнимия файл е в папка Test на диск С:, описваме пътя така: cd \Test.

**б/ стартиране на изпълним файл**

Стартираме изпълнимия файл – въвеждаме име и Enter/отбелязваме го с ↵/, например:

Sum ↵

Въвеждаме входните данни от клавиатурата.

**в/ вход от файл**

Ако входните данни са много, можем да ги запишем във файл in.txt и да тестваме така:

Sum < in.txt ↵

**г/ изход във файл**

Ако искаме резултатите да бъдат записани в файл out.txt, тестваме така:

sum < in.txt > out.txt

д/ създаване на EXE в конзолен прозорец – описваме пътя до компилатора, пътя до сорс-файла /ако е текущата директория, пътя може да се пропусне/, аргумент –o, който казва какво ще е името на изпълнимия файл, последван от самото име:

C:\Dev-Cpp\bin\g++ sum.cpp –o sum.exe

**2. Тестване на задачи с тестови данни**

**a/ тестови данни**

Тестовете на задачите, давани по състезания и олимпиади, се предоставят от авторите в папката с условието или отделно в архив. Входните файлове имат име във вида: name.01.in, където 01 е номера на теста, обикновено от 01 /или 00/ до 10.

Изходните данни на съответния тест са във файл със следното име: name.01.sol

**б/ ръчно тестване**

Можем да тестваме в конзолен прозорец, като посочим входния файл и сравним резултата на екрана с този в съответния изходен файл.

Можем да създадем наш изходен файл и после да го сравним с този на автора – отваряме в Notepad и сравняваме.

Можем да сравним двата файла /означаваме нашия с name.01.out, a на автора – name.01.sol/ и в конзолен прозорец с команда fc по следния начин:

fc name.01.out name.01.sol

**в/ тестване на решението чрез ВАТ – файлове**

Тестването на изпълнимия файл е автоматизирано чрез използването на три BAT файла, които ще намерите в папка TESTER.

* Файлът DO се извиква за всеки тест и извежда резултата на участника и верния резултат
* Файлът DOTEST съдържа номерата на тестовете и за всеки тест стартира файл DO
* Файлът TESTER съдържа името на изпълнимия файл, без разширението. За всяка задача трябва в този файл да променим името на изпълнимия файл, който ще тестваме.

**Процедурата по тестването е:**

Създаваме изпълнимия файл. Копираме тестовете в папката на изпълнимия файл.

Копираме файловете Do.bat, DoTest.bat и Tester.bat в същата папка.

Отваряме Tester.bat и променете името на изпълнимия файл.

Стартираме Tester.bat и сравняваме двата резултата.

**Пример: НОИ 2010 – група С, задача C1 – KPERMS.**

В папката са сорс файла и тестовeте.

* Отворете и компилирайте. Създава се изпълнимия файл.
* Копирайте файловете Do.bat, DoTest.bat и Tester.bat в същата папка.
* Отворете Tester.bat с д.б. и Edit и променете името на файла да е kperms
* Стартирайте Tester.bat

**Задача: Да се напише програма, която намира сумата на n цели числа.**

int n, a;

cin>>n;

long long S = 0;

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

{

cin>>a;

S = S + a ;

}

cout << S <<endl;

**3. Създаване на входни файлове с тестове – name.00.in**

**Задача: Да се създадат входни файлове за намиране на сумата на n цели числа.**

На първия ред се въвежда броя на числата. На втория ред се въвеждат последователно n цели числа, разделени с интервал.

Входните файлове ще съдържат случайни числа – броят им зависи от това колко входни данни очаква програмата. Файловете имат едно и също име – името на програмата, но се различават по номера – обикновено са от 00 до 10, например name.00.in.

**a/ генериране на случайни числа**

Генерирането на случайните числа става с функция rand(), която връща цяло положително случайно число до 32 767. Изисква да включим файл cstdlib /или stdlib.h/.

Ако искаме случайните числа да са в някакъв интервал, например до 100, използваме операция %, например rand()%100. За да бъдат случайните числа всеки път различни, използваме еднократно функция srand (time(NULL)). Аргумент на функцията е функция time осигурява произволни случайни числа, която връща системното време в секунди от 1.1.1970 г. Тези функции изискват файлове ctime /или time.h/ и cstdlib.

n = rand() % 100;

a = rand() % 100;

**б/ запис във файл name.00.in**

За работа с файлове, ще трябва да декларираме файлова променлива, чрез която ще е обръщаме към файла. Според това дали ще четем или записваме данни задаваме типа ifstream /четене/ или ofstream /запис/. Отваряме файла – така свързваме файловата променлива с конкретен файл от диска. Четем или записваме данни и накрая затваряме файла.

Запис във файл ще стане със следния код:

ofstream ofile; // деклариране на файлова променлива за четене

ofile.open (“Sum.00.in”); //отваряме файл за запис,

//Ако не съществува – го създава, ако го има – изтрива съдържанието му

ofile << n << endl; //записваме във файла

ofile << a << endl;

След стартиране се създава указания файл и в него се записват случайни числа. Ако стартираме многократно, като сменяме името, можем да създадем всички входни файлове.

**в/ автоматизирано създаване на всички входни файлове**

За да автоматизираме създаването на всички файлове едновременно, въвеждаме тези номера /от 00 до 10/ в един текстов файл TestNumbers.txt на отделни редове.

Съдържание на файл **TestNumbers.txt** – номерата на тестовете:

00

01

02

03

04

05

06

07

08

09

10

Създаването на входните файлове ще направим с програма на С++ с име make-in.cpp. Тя ще чете номер **NN** от файл TestNumbers.txt**,** ще генерира случайни числа и ще ги записва във файл с име **name.NN.in.**

Четенето от файл TestNumbers.txt става със следния код:

ifstream ifile; //деклариране на файлова променлива за четене

ifile.open ("TestNumbers.txt"); //отваряне на файл TestNumbers.txt

ifile >>s; //четене на стойност от файла и запазване в променлива s

ifile.close(); //затваряне на файла TestNumbers.txt

Записът във файл **name.NN.in** става със следния код:

ofstream ofile; // деклариране на файлова променлива за четене

ofile.open (("Sum."+s+".in").c\_str());

//Името се генерира от името на програмата + номер на тест + разширение in

//Изисква преобразуване в с-низ чрез функция c\_str().

ofile << n << endl; //запис във файл

ofile.close(); //затваряне на файл

Окончателно файлът **make-in.cpp** има следното съдържание:

#include <iostream>

#include <fstream>

#include <string>

#include <ctime>

#include <stdlib.h>

using namespace std;

int main()

{ ifstream ifile;

ifile.open ("data.txt");

string s;

ofstream ofile;

srand (time(NULL));

while (ifile >>s)

{ ofile.open (("Sum."+s+".in").c\_str());

int n = rand()%100;

ofile << n << endl;

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

ofile << rand()<< " ";

ofile << endl;

ofile.close();

}

ifile.close();

return 0;

}

Стартираме и в текущата папка се създават файловете с входните данни.

**4. Създаване на изходни файлове – name.00.sol**

**а/ в конзолен прозорец**

Създаване на файл с резултати става чрез стартиране на програмата в прозореца на конзолата и пренасочване на входа и изхода. Това става чрез следния код:

sum < sum.00.in > sum.00.sol

Предварително трябва направим текуща директория тази, в която са изпълнимия файл и тестовете. Това трябва да се повтори за всички входни файлове, като сменяме и името на изходния файл.

**б/ автоматизирано – чрез ВАТ файл под Windows**

Този процес може да се автоматизира, като създадем един BAT файл **make-sol.bat**, в който опишем последователно създаването на всички изходни файлове. В този файл ще укажем името на изпълнимия файл и ще със следното съдържание:

@echo off

set name=sum

echo %name%.00.sol

%name%.exe <%name%.00.in >%name%.00.sol

echo %name%.01.sol

%name%.exe <%name%.01.in >%name%.01.sol

echo %name%.02.sol

%name%.exe <%name%.02.in >%name%.02.sol

echo %name%.03.sol

%name%.exe <%name%.03.in >%name%.03.sol

echo %name%.04.sol

%name%.exe <%name%.04.in >%name%.04.sol

echo %name%.05.sol

%name%.exe <%name%.05.in >%name%.05.sol

echo %name%.06.sol

%name%.exe <%name%.06.in >%name%.06.sol

echo %name%.07.sol

%name%.exe <%name%.07.in >%name%.07.sol

echo %name%.08.sol

%name%.exe <%name%.08.in >%name%.08.sol

echo %name%.09.sol

%name%.exe <%name%.09.in >%name%.09.sol

echo %name%.10.sol

%name%.exe <%name%.10.in >%name%.10.sol

Този файл се стартира в Windows. Той трябва да е в същата папка, където е изпълнимия файл и файловете с входните данни.

**в/ чрез код на С++**

Редактираме решението на задачата така, че да данните да се четат от файл и резултатите да се записват във файл. Имената на файловете да се генерират чрез слепването на името на програмата, номера на теста и разширение in.

Може да напишем и нова програма, която да върши всичко едновременно – да генерира случайни числа, да ги записва в in файл, да им намира сумата и да записва резултата във sol файл.

**Задача:** Да се напише програма, която намира сумата на числата под и над главния диагонал в квадратна матрица. Да се създадат тестови данни.

<http://arena.maycamp.com/> - Задачите от НОИ2010 – група С, D, E – І кръг, Арена Уран

<http://www.math.bas.bg/infos/>