

Описание на програмата

Програмата започва с приема на вход от клавиатурата, който се присвоява на *целочислена променлива N*.

N има **смисъла** на *броя на триъгълниците, които ще се обработват от програмата*.

При въвеждане на число, *което надвишава 24 (или е отрицателно)*, се извежда съобщение за грешка и се очаква повторно въвеждане от потребителя. Такива стойности са лишени от смисъл. При въвеждане на стойност 0, програмата излиза със съобщение за това.

Въвеждането на стойност нула също няма голям смисъл, *но е начин програмата да се пркъсне коректно от потребителя*.

След това се въвеждат данни за съответния брой триъгълници.

При въвеждането на страните на всеки триъгълник се прави проверка *дали такъв триъгълник съществува*.

Ако триъгълник с въведените страни **НЕ съществува**, се извежда съобщение и се излиза от програмата със статус 200, което има смисъл на грешка с код 200:

```
if (not(triangle_exists(a,b,c)))
{
    cout << "Triugulnik s vuvdenite strani ne sushtestvuwa. Izlizane ot
programata. ";
    return 200;
}
```

Очевидно тази част от програмата *може да бъде подобрена*, ако проверката води до очакване потребителят да въведе нови стойности за страните (*вместо грубо да се излиза от нея*). Но за момента оставяме тази възможност.

След приключване на въвеждането се изчисляват **медианите на триъгълника и радиусът на вписаната в него окръжност**:

```
triangleset[i][3]=ma(a,b,c);
triangleset[i][4]=mb(a,b,c);
triangleset[i][5]=mc(a,b,c);
triangleset[i][6]=r(a,b,c);
```

Така пресметнатите стойности се съхраняват в отделни масиви (i-тият триъгълник е в i-тия масив, където i=1, 2, ..., N. Тук N е въведения в началото брой триъгълници).

След приключване на изчисленията се пристъпва към намиране на средноаретметичната стойност на радиусите на вписаните окръжности и изчисляване на това колко от тях имат радиус, по-малък от нея:

* Средноаретметичната стойност на радиусите на вписаните окръжности:

```
double sum=0;
    for (i=1; i<=N; i++)
        sum += triangleset[i][6];

double r_average=sum/N;
```

Тя е сумата (изчислена в цикъла и присвоена на реалната променлива 'sum'), разделена на броя N, т.е. sum/N.

* Брой триъгълници с радиуси, по-малки от нея:

```
int j=0;           // променливата съдържа броя триъгълници с радиус, по-малък
от средния. В началото той е 0.
```

```
    for (i=1; i<=N; i++)
    {
        if (triangleset[i][6]<=r_average)
            j++;    // Ако съответният радиус не надвишава средния, стойността на j
расте с единица
    }              // В края на цикъла променливата съдържа информация за
триъгълниците с радиус,
                // ненадвишаващ средния за всички триъгълници.
```

След това се преминава към пресмятане на броя триъгълници, надвишаващи по радиус средния.

Тук използваме един простичък трик:

Броят на тези триъгълници може да се получи като от N извадим j. Наистина, ако търсеният брой триъгълници е k, то имаме:

$$j + k = N \Rightarrow k = N - j$$

Така пресмятането и извеждането се свеждат до един ред:

```
cout << "Имаме " << N-j << " триъгълници с радиус, по-малък от средния. ";
```

С това програмата е постигнала целите си. Последният фрагмент от програмата осигурява "меко" излизане.

Очаква се натискането на Ентер два пъти за излизане от програмата. Един вид програмата не излиза, докато потребителят не предприеме това действие. Дотогава той може да разглежда резултатите от програмата, ако се налага.

Но дори и да няма нужда от много време за това, директното излизане води до скриване на резултатите и е най-малкото неприятно.

Списък променливи, използвани в програмата

Променливите, които са използвани, са: N, i, a, b, c, sum, j, a, r_average

В програмата се срещат и параметри на функциите, участващи в нея. Но те не ще бъдат коментирани тук.

N - Брой на триъгълниците

i - Брояч за циклите с управляваща променлива

a, b и c - Страните на триъгълника

sum - Променлива, в която записваме сумата от дължините на радиусите на вписаните окръжности във всеки триъгълник

j - Брой триъгълници, ненадвишаващи по радиус (на вписаната в тях окръжност) средния

a - ОНОВО! Предефинирана в края на програмата за въвеждане на символ преди излизане

r_average - средно аритметично на дължините на радиусите на вписаните окръжности

Използвани библиотеки

В програмата не са използвани дефинирани библиотеки. Използвани са само стандартните библиотеки `iostream` и `cmath`:

#include <iostream> - Стандартна библиотека за управление на потоците вход и изход.
Използваме `cin` и `cout`, които са част от именно тази библиотека;

#include <cmath> - Необходима ни е само за пресмятането на квадратен корен от реален израз:
`return sqrt(ss);`