

НАЦИОНАЛНА ОЛИМПИАДА ПО ИНФОРМАТИКА

Общински кръг, 27 януари 2007 г.

Автори на задачите

A1 – Красимир Манев A2 – Красимир Манев A3 – Красимир Манев	D1 – Венета Богданова D2 – Венета Богданова D3 – Галина Момчева
B1 – Младен Манев B2 – Николина Николова B2 – Емил Келеведжиев	E1 – Бисерка Йовчева E2 – Велислава Христова E3 – Велислава Христова
C1 – Павлин Пеев C2 – Павлин Пеев C3 – Каталина Григорова	

Задача A1. РАЗЛИКА

На стандартния вход е зададено цяло положително число R , не по-голямо от 4100000000. Напишете програма **DIF**, която намира и извежда на стандартния изход две цели неотрицателни числа N и M , такива че сумата на целите числа от 1 до N , минус сумата на целите числа от 1 до M да е равна на R . Ако съществуват няколко двойки N и M с исканото свойство, програмата трябва да изведе онази от тях, за която N е най-малко. Входните данни са такива, че сумата на целите числа от 1 до N не надхвърля 4100000000.

ПРИМЕР 1

Вход	Изход
6	3 0

ПРИМЕР 2

Вход	Изход
61	31 29

Решение:

Да означим с S_K сумата на целите числа от 1 до K . Погледнато математически, задачата винаги има решение защото $S_R - S_{R-1} = R$. Тривиалната идея да се използва формула за сумите S_N (на числата от 1 до N) и S_M (на числата от 1 до M) и да се сравнява $S_N - S_M$ с R , очевидно е приложима само за неголеми стойности на N . А за тези, за които е приложима, тази идея ще ни даде алгоритъм със сложност $O(N^2)$.

За да получим алгоритъм със сложност $O(N)$ прилагаме достатъчно общата схема, която ще наричаме „плъзгащ се прозорец”. В началото образуваме прозореца, като пресмятаме сумата на първите K числа, където K е най-малкото цяло за което $S_K \geq R$. Ако $S_K = R$, тогава $N = K$, $M = 0$ е търсеното решение. В противен случай започваме „плъзгане” на получения прозорец $[M, N]$. Докато $S_N - S_M > R$, увеличаваме M с 1. Когато $S_N - S_M$ стане по-малко от R , докато $S_N - S_M < R$, увеличаваме N с 1. Когато за пръв път $S_N - S_M$ стане равно на R , текущите стойности на N и M са търсеното решение. При зададените параметри, стойностите на S_N и S_M трябва да се съхраняват в подходящ тип (най-малко `unsigned int` за пишещите на C/C++).

```

#include <stdio.h>
main()
{
    unsigned N=0,M=0,SN=0,SM=0,R;
    scanf("%d",&R);
    while(SN+N<R){SN+=N;N++;}SN+=N;
    if(SN>R)
        while(SM+M<SN-R){SM+=M;M++;}
    SM+=M;
    if(SN-SM==R) printf("%d %d\n",N,M);
    else
    {
        while(N<=R)
        {
            N++;
            //if(SN+N<SN){printf("Not calculable!\n");break;}
            SN+=N;M++;
            while(SM+M<SN-R){SM+=M;M++;}
            SM+=M;
            if(SN-SM==R){printf("%d %d\n",N,M);break;}
        }
    }
}

```

Решението има един недостатък. То няма да даде искания резултат, ако някоя от сумите S_N и S_M надхвърли максималната стойност на използвания тип. Затова е ограничението, всяка една от сумите да не надхвърля 4100000000. Тази стойност, обаче, е условна, за да се избегне явното указване на необходимия тип. Забележете, че в решението (поставения като коментар ред) се използва максимално възможностите на типа `unsigned int`. Така в множеството от тестове се е „промъкнал” тест при който $S_N > 4100000000$, което не е проблем. Ако някои състезатели са следвали буквално условието и затова не са получили решение на този тест, редно е да се пусне допълнителен тест, като за да се намери такъв, трябва да се промени условието $SN+N<SN$ на $SN+N>4100000000$.