

# НАЦИОНАЛНА ОЛИМПИАДА ПО ИНФОРМАТИКА

Национален кръг, 29-30 април 2006 г.

Тема за група В (10-11 клас)

## Задача В1. RISC-ПРОЦЕСОР

Съвременните RISC-процесори са невероятно бързи, но както подсказва името им (**Reduced Instruction Set Computer**), имат ограничен брой инструкции. На вашия нов компютър дизайнерите са изгубили всякаква мярка! Осигурили са само три *регистъра* (64-битови целочислени променливи): **A**, **B** и **C** и следните *операции*:

- всеки от регистрите **A** и **B** може да увеличава стойността си с 1. Такава операция ще означаваме с името на регистъра, върху който се прилага, т. е. имаме операциите **A** и **B**;

- към съдържанието на регистър **A** може да се добави съдържанието на регистър **B**, като съдържането на **B** не се променя. Тази операция ще означаваме просто с **+**;

- регистрите **A** и **B** могат да копират съдържанието на единия в другия, а **A** може да копира и съдържанието на **C**. Това действие ще означаваме с **=** между участващите операнди. Т. е. имаме операциите **A=B**, **B=A** и **A=C**. Левият операнд при тези операции копира стойността на десния, десният не се променя;

- има конструкция за цикъл, която започва със сравняване на съдържанието на **B** с това на **C**. Операцията ще означаваме с **?**. Тялото на цикъла започва (или продължава) да се изпълнява само ако при проверката съдържанието на **B** е строго по-малко от това на **C**. Краят на цикъла ще означаваме с инструкцията **LOOP**, при която управлението безусловно се предава към началото на цикъла, където отново се прави проверката. **ВНИМАНИЕ** – цикли не могат да се влагат – възниква системна грешка!

Подпрограмите за този процесор са функции с не повече от един входен параметър, задължително подаван в **C**, и един изходен параметър, задължително връщан в **A**. При влизането в подпрограма регистър **A** автоматично се нулира, а в **B** има стойност 1. Всяка инструкция ще записваме на нов ред. Ето три примера за подпрограми:

1. Подпрограма за изчисляване на числото 6, независимо от съдържанието на **C**:

A  
A  
A  
A  
A  
A

2. Подпрограма за утрояване на цялото число **n** (в **C** се подава **n**):

A=C  
B=A  
+  
+

3. Подпрограма, която намира сумата на естествените числа от 1 до **n** (в **C** се подава **n**):

A=B  
?  
B  
+  
LOOP

Описаните подпрограми, разбира се, не са единствените, които решават съответната задача.

Съвсем очевидно е, че първата подпрограма, например, има следната еквивалентна (и с по-малък брой редове):

B  
B  
+  
+

В конкретните си изследвания в момента вие се занимавате със сумата от квадратите на последователни естествени числа и ви трябва функция, които да намират тази сума. Ще се наложи да поемете риска за написване на програма **RISC**, която за зададено естествено число **n** създава нужната подпрограма за вашия RISC-процесор. При вход в регистър **C** ще е числото **n**, а накрая в **A** трябва да се получи числото  $n^2 + (n+1)^2$ . А тъй като после ще преписвате получените подпрограми, предпочитате те да имат **колкото може по-малък брой** редове-инструкции.

От стандартния вход се въвежда един ред с естественото число **n** ( $1 \leq n \leq 10000$ ). Запишете на стандартния изход:

- ред 1: брой редове **m**, които Вашата RISC-подпрограма съдържа;
- на останалите **m** реда запишете една от най-късите намерени от Вас RISC-подпрограми за изчисляване на  $n^2 + (n+1)^2$  – по една инструкция на ред, последователно, както ще се въвежда и изпълнява.

## ПРИМЕР

### Вход

3

### Изход

8

?

В

+

+

+

+

+

LOOP

## Задача В2. СТОЛИЦА

В една държава има  $N$  града ( $5 \leq N \leq 10000$ ), като градовете са номерирани с числата от 1 до  $N$ . Някои от тях са свързани директно с шосета, като от всеки град излизат максимум по **4 различни шосета**. Всяко шосе е двупосочно и е известно разстоянието между градовете, които то свързва директно. Всеки два града, между които има шосе, отстоят минимум на **10 километра** един от друг. Между всеки два града в държавата има път.

„Отдалечени” ще наричаме всички градове, които се намират на минимално разстояние от столицата, **строго по-голямо** от  $K$  километра ( $1 \leq K \leq 66$ ). Напишете програма **CAPITAL**, която намира броя на „отдалечените” градове по зададена пътна мрежа и стойност на  $K$ .

Входните данни се четат от стандартния вход. На първи ред стои  $N$  – броя на градовете в държавата. Столицата винаги е град с номер 1. На втория ред се намира числото  $K$  – критерий за „отдалеченост”. Следват редове от тройки числа  $A B C$ , които задават връзка между два града.  $A$  и  $B$  са номерата на градовете, а  $C$  е дължината на директното шосе между тях в километри. Входните данни завършват с тройката 0 0 0.

На стандартния изход изведете едно единствено число – броя на намерените „отдалечени” градове.

## ПРИМЕР

### Вход

8

65

1 2 80

1 3 20

5 3 30

5 2 10

2 4 20

7 6 10

1 6 65

8 1 70

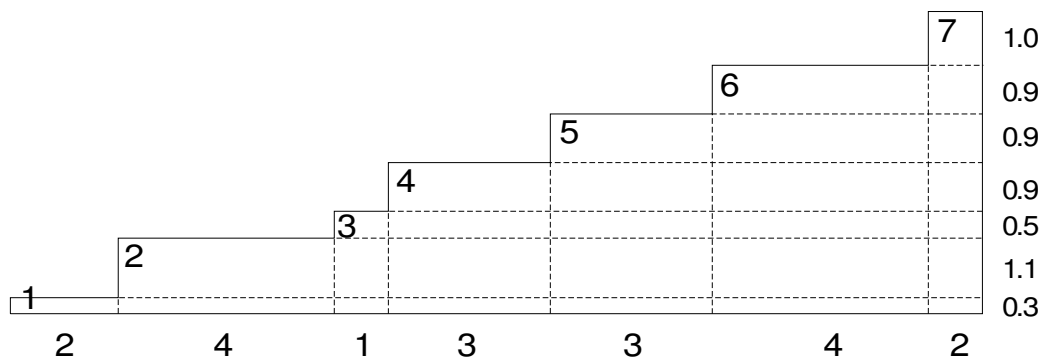
0 0 0

### Изход

3

### Задача В3. СТЬЛБАТА НА ФАРАОНА

Златна стълба с  $N$  стъпала ( $2 \leq N \leq 100$ ) води към трона на фараона. По време на реконструкция на двореца било решено броят на стъпалата да бъде намален до  $K$  ( $1 \leq K \leq N$ ) поради естетически съображения. Поради същите съображения било забранено на работещите по стълбата да секат златното изделие и единственият начин да се редуцира количеството стъпала е да се долива злато, така че две или повече съседни стъпала да се сливат в едно. Напишете програма **STAIRS**, която определя минималното количество злато, необходимо за планираната реконструкция.



Входните данни се четат от стандартния вход. На първи ред са  $N$  и  $K$  – броя стъпала, които има първоначално стълбата, и броя стъпала, които трябва да има след реконструкцията. Следват  $N$  реда от двойки положителни числа, разделени с интервал – височина и дълбочина на поредното стъпало в дециметри (ширината на стълбата е постоянна – за целите на задачата ще предпологаеме, че това е една двумерна стълба).

На стандартния изход изведете едно число, закръглено с точност до третия знак – минималното количество злато, което е необходимо за изпълнение на задачата, в квадратни дециметри (напомняме, че разглеждаме стълбата като двумерен обект)

#### ПРИМЕР

##### Вход

```
7 6
0.3 2
1.1 4
0.5 1
0.9 3
0.9 3
0.9 4
1 2
```

##### Изход

```
0.900
```