

Cálculo de complexidade sem loops aninhados

Analisar a complexidade de tempo de execução de um algoritmo é uma tarefa muito importante para a criação de programas eficientes. Um algoritmo que executa em tempo linear é geralmente muito mais rápido que um algoritmo que necessita de tempo exponencial para a mesma tarefa.

Geralmente, o tempo de execução de um algoritmo é definido de acordo com o tamanho N da entrada, onde N pode ser a quantidade de elementos a serem ordenados, o número de pontos de determinado polígono e assim por diante.

Desta forma, determinar uma fórmula para o tempo de execução em função de N não é uma tarefa simples e seria muito bom se isso fosse automatizado. Infelizmente, isto em geral não é possível, mas neste exercício você irá considerar programas de natureza simples e poderá determinar a quantidade de operações de um programa com precisão. Considere que um analista efetuou anotações referentes à quantidade de repetições dos *loops* e à quantidade de tempo que cada operação (instrução) necessita. Seu programa deverá analisar estas anotações de acordo com a descrição a seguir e calcular a complexidade total em unidades de tempo:

- A entrada deverá iniciar com a palavra INICIO e terminar com a palavra FIM;
- A entrada poderá ter instruções que consomem determinada quantidade de tempo T , identificadas por **OP T** onde T é um número inteiro;
- A entrada poderá usar estruturas de repetição (*loop*) para repetir um bloco de comandos internos por determinada quantidade de vezes, mas não será permitido que este bloco de comandos contenha outro *loop* interno (isto é, não haverá *loop* aninhado). O *loop* deverá iniciar com a palavra LOOP seguida de um número inteiro indicando a quantidade de vezes que ele repetirá o bloco de comandos interno. Portanto, após o número haverá um bloco de comandos e o *loop* será finalizado com a palavra FIMLOOP. Observe o exemplo abaixo:

```
INICIO
  LOOP 10
    OP 7
    OP 3
  FIMLOOP
FIM
```

- O programa pode ter vários *loops*, desde que estejam no mesmo nível (não haverá *loop* aninhado).

REGRA GERAL: Você deverá utilizar python 3. Sempre que um novo *loop* for fechado, a quantidade de operações que foi calculada dentro dele deverá ser multiplicada pela quantidade de vezes que o *loop* executou.

Entrada

A entrada contém um programa que é construído de acordo com as regras acima. Espaços em branco, tabulações e quebras de linha (`\r` e `\n`) podem aparecer em qualquer local do programa e devem ser desconsiderados, mas não aparecerão dentro das palavras INICIO, FIM, LOOP, FIMLOOP, OP e nem dentro de um número inteiro. Por questões técnicas da plataforma de testes, toda a entrada será lida em uma única linha sem enter.

Saída

A saída deverá exibir o tempo de execução como um número inteiro.

Exemplos:

Exemplo de entrada 1: (Obs.: código identado para fins didáticos; a entrada será dada em uma única linha sem enter).

INICIO

 LOOP 10

 OP 4

 OP 1

 FIMLOOP

 OP 17

 LOOP 3

 OP 5

 FIMLOOP

 OP 2

FIM

Exemplo de saída 1:

84

Exemplo de entrada 2:

INICIO OP 199 LOOP 10 OP 1 FIMLOOP FIM

Exemplo de saída 2:

209