

# Computador

Nome do arquivo: `computador.c`, `computador.cpp`, `computador.pas`, `computador.java`, `computador.js`,  
`computador_py2.py` ou `computador_py3.py`

Uma grande empresa está construindo uma nova arquitetura de computadores que permita a execução eficiente de duas instruções especiais de soma. O computador possui  $N$  posições de memória, endereçadas de 1 a  $N$ , e cada posição pode guardar um inteiro maior ou igual a zero. Inicialmente, todas as posições contêm o valor zero. As instruções especiais de soma são:

- **FRENTE  $i$   $V$ :** Dado o endereço  $i$ ,  $1 \leq i \leq N$ , e um valor positivo  $V$ , o computador deve somar  $V$  na posição  $i$ ,  $V - 1$  em  $i + 1$ ,  $V - 2$  em  $i + 2$ , etc, enquanto o valor a ser somando for maior do que zero e a posição for menor ou igual a  $N$ ;
- **TRÁS  $i$   $V$ :** Dado o endereço  $i$ ,  $1 \leq i \leq N$ , e um valor positivo  $V$ , o computador deve somar  $V$  na posição  $i$ ,  $V - 1$  em  $i - 1$ ,  $V - 2$  em  $i - 2$ , etc, enquanto o valor a ser somando for maior do que zero e a posição for maior ou igual a 1.

Por exemplo, para  $N = 16$ , uma possível sequência de instruções é dada abaixo:

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

FRENTE 4 8

0	0	0	8	7	6	5	4	3	2	1	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

TRÁS 16 3

0	0	0	8	7	6	5	4	3	2	1	0	0	1	2	3
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

TRÁS 2 12

11	12	0	8	7	6	5	4	3	2	1	0	0	1	2	3
----	----	---	---	---	---	---	---	---	---	---	---	---	---	---	---

FRENTE 8 7

11	12	0	8	7	6	5	11	9	7	5	3	2	2	2	3
----	----	---	---	---	---	---	----	---	---	---	---	---	---	---	---

Além disso, o computador possui a instrução **IMPRIME  $i$** , que deve imprimir na saída o valor atual armazenado na posição  $i$  da memória.

Dados  $N$  e uma sequência de  $M$  instruções, seu programa deve imprimir, para cada instrução do tipo **IMPRIME  $i$** , uma linha contendo o valor armazenado na posição de memória  $i$  no instante da execução da instrução.

## Entrada

A primeira linha da entrada contém dois inteiros  $N$  e  $M$ , representando o número de posições de memória e o número de instruções, respectivamente. As  $M$  linhas seguintes contêm, cada uma, a descrição de uma instrução em uma de três formas possíveis: 1  $I$   $V$ , representando **FRENTE  $I$   $V$** ; 2  $I$   $V$ , representando **TRÁS  $I$   $V$** ; e 3  $I$ , representando **IMPRIME  $I$** .

## Saída

Para cada instrução do tipo **IMPRIME  $i$** , seu programa deve imprimir uma linha contendo um inteiro representando o valor armazenado na posição de memória  $i$  no instante da execução da instrução.

**Restrições**

- $1 \leq N \leq 200000$ ;
- $1 \leq M \leq 200000$ ;
- $1 \leq I \leq N$ ;
- $1 \leq V \leq 200000$ ;
- Ao menos uma instrução será do tipo 3.

**Informações sobre a pontuação**

- Em um conjunto de casos de teste somando 20 pontos,  $N \leq 10000$ ,  $M \leq 10000$  e  $V \leq 10000$ ;

**Exemplos**

<b>Exemplo de entrada 1</b>  16 7 1 4 8 2 16 3 3 14 2 2 12 1 8 7 3 10 3 14	<b>Exemplo de saída 1</b>  1 7 2
<b>Exemplo de entrada 2</b>  200000 2 1 2345 193290 3 112230	<b>Exemplo de saída 2</b>  83405