

Trabalho 1: Vetores e Funções

O objetivo deste trabalho é implementar uma máquina virtual simples que realiza a execução eficiente de duas instruções especiais de soma. O programa deve gerenciar um vetor de N posições na memória, e que são endereçadas de 1 a N pelo usuário. Cada posição do vetor armazena um inteiro maior ou igual a zero. Inicialmente, todas as posições contêm o valor zero. As instruções especiais de soma são:

- **FRENTE i V** : dados a posição i ($1 \leq i \leq N$) e um valor positivo V , a instrução 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** : dados a posição i ($1 \leq i \leq N$) e um valor positivo V , a instrução 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 a seguir:

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, a máquina virtual possui a instrução **IMPRIME i** , que deve imprimir na saída o valor atual armazenado na posição i do vetor.

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 i do vetor 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 contém, 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** $I \ V$;
- $3 \ I$, representando **IMPRIME** I .

Saída

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

Exemplos

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

Restrições

- $1 \leq N \leq 200.000$;
- $1 \leq M \leq 200.000$;
- $1 \leq I \leq N$;
- $1 \leq V \leq 200.000$;
- Ao menos uma instrução será do tipo 3 (**IMPRIME**).

Requisitos

- 1) Declare um vetor com capacidade 200.000, para respeitar a restrição a);
- 2) Cada instrução deve ser implementada por uma função. Sugestão:

```
void frente(int vet[], int n, int i, int v);  
void tras(int vet[], int n, int i, int v);  
void imprime(int vet[], int n, int i);
```
- 3) Não é necessário validar os dados de entrada.

Critérios de avaliação

- Execução correta e alinhamento com o que foi solicitado neste enunciado.

Informações importantes:

- **Trabalho individual!**
- **Entrega:** no Moodle, até o dia **18/12**.