

Carteiro

Nome do arquivo fonte: `carteiro.c`, `carteiro.cpp`, `carteiro.pas`, `carteiro.java`, ou `carteiro.py`

Um carteiro é o responsável por entregar as encomendas na rua de Joãozinho. Por política da empresa, as encomendas devem ser entregues na mesma ordem que foram enviadas, mesmo que essa não seja a forma mais rápida. Cansado de subir e descer aquela rua tantas vezes, nosso amigo quer mostrar à empresa quanto tempo ele leva para entregar as encomendas, na tentativa de derrubar essa política.

A rua de Joãozinho tem N casas. Naturalmente, as casas são numeradas de forma ordenada (não necessariamente por números consecutivos). Como as casas possuem aproximadamente o mesmo tamanho, você pode assumir que o carteiro leva uma unidade de tempo para caminhar de uma casa até a casa imediatamente vizinha.

Há M encomendas para essa rua, que devem ser entregues na mesma ordem em que chegaram. Cada encomenda contém o número da casa onde deve ser entregue.

Escreva um programa que determine quanto tempo o carteiro levará para entregar todas as encomendas, assumindo que quando o tempo começa a contar, ele está na primeira casa (a de menor número), e o tempo termina de contar quando todas as encomendas foram entregues (mesmo que o carteiro não esteja de volta na primeira casa). Você pode desprezar o tempo para colocar a encomenda na caixa de correio (ou seja, se ele só tiver uma encomenda, para a primeira casa, a resposta para o problema é zero).

Entrada

A primeira linha contém dois inteiros, N e M , respectivamente o número de casas e o número de encomendas. A segunda linha contém N inteiros em ordem estritamente crescente, indicando os números das casas. A terceira linha contém M inteiros indicando os números das casas onde as encomendas devem ser entregues, na ordem dada na entrada.

Saída

Seu programa deve produzir uma única linha, contendo um único inteiro, o tempo que o carteiro levará para entregar todas as encomendas na ordem correta, assumindo que ele começa na casa de menor número.

Restrições

- $1 \leq N \leq 45.000$ e $1 \leq M \leq 45.000$
- O número de cada casa é um inteiro entre 1 e 1.000.000.000

Informações sobre a pontuação

- Para um subconjunto dos casos de teste totalizando 30 pontos, $1 \leq N \leq 1000$ e $1 \leq M \leq 1000$.

Exemplos

Entrada	Saída
5 5 1 5 10 20 40 10 20 10 40 1	10

Entrada	Saída
3 4 50 80 100 80 80 100 50	4

Ogros

Nome do arquivo fonte: `ogros.c`, `ogros.cpp` ou `ogros.pas`

Ogros marcianos, como todos sabem, são extremamente fortes. Numa feira de circo marciano, ogros são chamados para bater um martelo num aparelho que mede sua força. O ogro ganha um determinado prêmio dependendo da faixa de força que alcançou (por exemplo, se a força alcançada foi entre 0 e 5, ganha 10 pontos. Se for entre 6 e 10, ganha 30). São possíveis N premiações, para N faixas de força alcançadas.

Você deve escrever um programa que recebe quais são as faixas de forças e suas respectivas premiações. Em seguida, o programa recebe a força alcançada por M ogros, e deve calcular quanto cada ogro recebeu de premiação.

Entrada

A entrada contém um único conjunto de testes, que deve ser lido do *dispositivo de entrada padrão* (normalmente o teclado). A primeira linha contém dois inteiros N e M ($2 \leq N \leq 10.000$, $1 \leq M \leq 10.000$), representando respectivamente o número de faixas de premiações e o número de ogros cuja força foi medida.

A segunda linha de um caso de teste contém $N - 1$ inteiros A_i ($A_{i-1} < A_i < A_{i+1}$, $1 \leq A_i \leq 1.000.000.000$). A primeira faixa de premiação é dada por forças menores que A_1 . A i -ésima faixa de premiação é composta das forças que são maiores ou iguais a A_{i-1} e menores que A_i . A n -ésima pontuação é composta das forças maiores ou iguais a A_{N-1} .

A terceira linha contém N inteiros F_i ($1 \leq F_i \leq 1.000.000.000$, $F_{i-1} < F_i < F_{i+1}$) em ordem crescente, representando a premiação de cada faixa de premiação, nesta ordem.

A quarta e última linha de um caso de teste contém M inteiros O_i ($1 \leq O_i \leq 1.000.000.000$), um para cada ogro, representando qual força cada ogro alcançou.

Saída

Seu programa deve imprimir, na *saída padrão*, uma única linha, contendo M inteiros, um para cada ogro, na ordem dada pela entrada, representando qual premiação cada ogro recebeu pela sua força alcançada.

Exemplo de entrada	Exemplo de saída
<pre>3 4 3 5 1 4 7 2 3 9 4</pre>	<pre>1 4 7 4</pre>

Exemplo de entrada	Exemplo de saída
<pre>2 10 4 5 200 1 3 4 5 5 6 2 1 3 4</pre>	<pre>5 5 200 200 200 200 5 5 5 200</pre>

Exemplo de entrada	Exemplo de saída
<pre>10 1 1 2 3 4 5 6 7 8 9 1 10 100 101 102 103 104 105 106 200 5</pre>	<pre>103</pre>