Competidor(a): _____

Número de inscrição: _____–_____ *(opcional)*

OBI2019

Caderno de Tarefas

Modalidade **Programação** • **Nível 1** • Fase **Local**

30 de maio de 2019

A PROVA TEM DURAÇÃO DE **2** HORAS

Promoção:

Sociedade Brasileira de Computação

Apoio:

CNPq
Conselho Nacional de Desenvolvimento Científico e Tecnológico

Instruções

LEIA ATENTAMENTE ESTAS INSTRUÇÕES ANTES DE INICIAR A PROVA

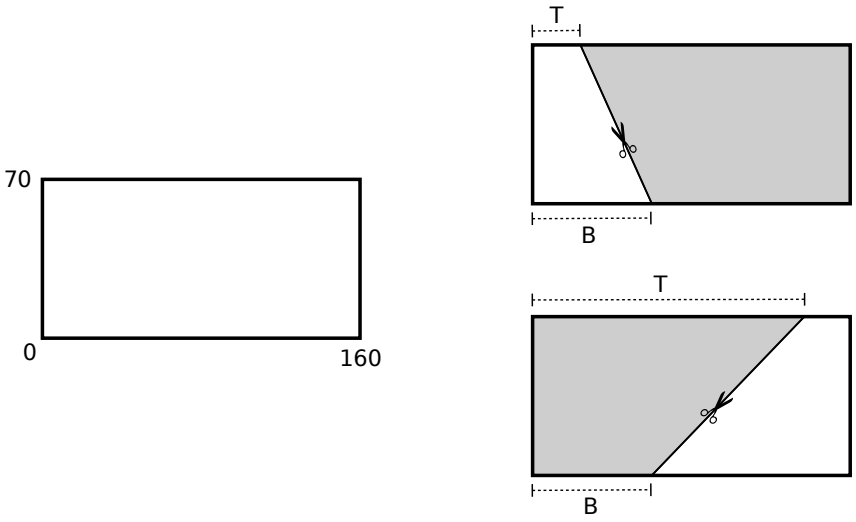
- Este caderno de tarefas é composto por 7 páginas (não contando a folha de rosto), numeradas de 1 a 7. Verifique se o caderno está completo.
- A prova deve ser feita individualmente.
- É proibido consultar a Internet, livros, anotações ou qualquer outro material durante a prova. É permitida a consulta ao *help* do ambiente de programação se este estiver disponível.
- As tarefas têm o mesmo valor na correção.
- A correção é automatizada, portanto siga atentamente as exigências da tarefa quanto ao formato da entrada e saída de seu programa; em particular, seu programa não deve escrever frases como "Digite o dado de entrada:" ou similares.
- Não implemente nenhum recurso gráfico nas suas soluções (janelas, menus, etc.), nem utilize qualquer rotina para limpar a tela ou posicionar o cursor.
- As tarefas **não** estão necessariamente ordenadas, neste caderno, por ordem de dificuldade; procure resolver primeiro as questões mais fáceis.
- Preste muita atenção no nome dos arquivos fonte indicados nas tarefas. Soluções na linguagem C devem ser arquivos com sufixo *.c*; soluções na linguagem C++ devem ser arquivos com sufixo *.cc* ou *.cpp*; soluções na linguagem Pascal devem ser arquivos com sufixo *.pas*; soluções na linguagem Java devem ser arquivos com sufixo *.java* e a classe principal deve ter o mesmo nome do arquivo fonte; soluções na linguagem Python 2 devem ser arquivos com sufixo *.py2*; soluções na linguagem Python 3 devem ser arquivos com sufixo *.py3*; e soluções na linguagem Javascript devem ter arquivos com sufixo *.js*.
- Na linguagem Java, **não** use o comando *package*, e note que o nome de sua classe principal deve usar somente letras minúsculas (o mesmo nome do arquivo indicado nas tarefas).
- Para tarefas diferentes você pode escolher trabalhar com linguagens diferentes, mas apenas uma solução, em uma única linguagem, deve ser submetida para cada tarefa.
- Ao final da prova, para cada solução que você queira submeter para correção, copie o arquivo fonte para o seu diretório de trabalho ou pen-drive, conforme especificado pelo seu professor.
- Não utilize arquivos para entrada ou saída. Todos os dados devem ser lidos da entrada padrão (normalmente é o teclado) e escritos na saída padrão (normalmente é a tela). Utilize as funções padrão para entrada e saída de dados:
  - em Pascal: *readln, read, writeln, write*;
  - em C: *scanf, getchar, printf, putchar*;
  - em C++: as mesmas de C ou os objetos *cout* e *cin*.
  - em Java: qualquer classe ou função padrão, como por exemplo *Scanner, BufferedReader, BufferedWriter* e *System.out.println*
  - em Python: *read,readline,readlines,input,print,write*
  - em Javascript: *scanf, printf*
- Procure resolver a tarefa de maneira eficiente. Na correção, eficiência também será levada em conta. As soluções serão testadas com outras entradas além das apresentadas como exemplo nas tarefas.

Nota cortada

Nome do arquivo: "`nota.x`", *onde* `x` *deve ser* `c`, `cpp`, `pas`, `java`, `js`, `py2` *ou* `py3`

Se pegarmos uma nota de 100 reais e a cortarmos, usando uma tesoura, em dois pedaços, quanto vale cada um dos pedaços? A regra é simples: se um dos pedaços possuir estritamente mais da metade da área da nota original, então ele vale 100 reais; e o outro pedaço não vale nada. Veja que se cada pedaço possuir exatamente metade da área original, então nenhum dos dois tem valor.

Felix e Marzia decidiram fazer um corte, em linha reta, que comece no lado inferior da nota, a base, e termine no lado superior, o topo. A nota é um retângulo de comprimento 160 centímetros e altura 70 centímetros, como mostrado na parte esquerda da figura abaixo. Felix sempre vai ficar com o pedaço mais à esquerda da nota e Marzia com o pedaço mais à direita. A parte direita da figura ilustra dois possíveis cortes. No de cima, Marzia ficaria claramente com o maior pedaço, que vale 100 reais; e no de baixo, dá para ver que Felix é quem ficaria com o maior pedaço.



O corte reto vai começar na base a uma distância de B centímetros a partir do lado esquerdo da nota; e terminar no topo a uma distância de T centímetros também a partir do lado esquerdo da nota. Veja a indicação na parte direita da figura.

Neste problema, dados os valores B e T, seu programa deve computar quem vai ficar com o pedaço que vale 100 reais, ou se o valor da nota se perdeu.

Entrada

A primeira linha da entrada contém um inteiro B representando a distância do ponto inicial do corte, na base, para o lado esquerdo da nota. A segunda linha da entrada contém um inteiro T representando a distância do ponto final do corte, no topo, para o lado esquerdo da nota.

Saída

Seu programa deve imprimir uma linha contendo um número inteiro: 1, se Felix ficou com o pedaço que vale 100 reais; 2, se Marzia ficou com o pedaço que vale 100 reais; ou 0, se o valor da nota se perdeu.

Restrições

- $0 < B < 160$
- $0 < T < 160$

Informações sobre a pontuação

- Em um conjunto de casos de teste somando 25 pontos, $B = T$

Exemplos

Exemplo de entrada 1	**Exemplo de saída 1**
50 86	2

Exemplo de entrada 2	**Exemplo de saída 2**
70 90	0

Exemplo de entrada 3	**Exemplo de saída 3**
130 138	1

A idade de Dona Mônica

Nome do arquivo: "`idade.x`", *onde* `x` *deve ser* `c`, `cpp`, `pas`, `java`, `js`, `py2` *ou* `py3`

Dona Mônica é mãe de três filhos que têm idades diferentes. Ela notou que, neste ano, a soma das idades dos seus três filhos é igual à idade dela. Neste problema, dada a idade de dona Mônica e as idades de dois dos filhos, seu programa deve computar e imprimir a idade do filho mais velho.

Por exemplo, se sabemos que dona Mônica tem 52 anos e as idades conhecidas de dois dos filhos são 14 e 18 anos, então a idade do outro filho, que não era conhecida, tem que ser 20 anos, pois a soma das três idades tem que ser 52. Portanto, a idade do filho mais velho é 20. Em mais um exemplo, se dona Mônica tem 47 anos e as idades de dois dos filhos são 21 e 9 anos, então o outro filho tem que ter 17 anos e, portanto, a idade do filho mais velho é 21.

Entrada

A primeira linha da entrada contém um inteiro M representando a idade de dona Mônica. A segunda linha da entrada contém um inteiro A representando a idade de um dos filhos. A terceira linha da entrada contém um inteiro B representando a idade de outro filho.

Saída

Seu programa deve imprimir uma linha, contendo um número inteiro, representando a idade do filho mais velho de dona Mônica.

Restrições

- $40 \leq M \leq 110$
- $1 \leq A < M$
- $1 \leq B < M$
- $A \neq B$

Exemplos

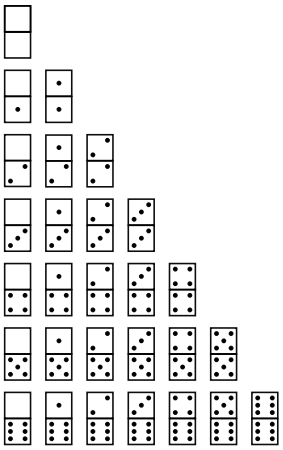
Exemplo de entrada 1	**Exemplo de saída 1**
52 14 18	20

Exemplo de entrada 2	**Exemplo de saída 2**
47 21 9	21

Jogo de Dominós

Nome do arquivo: "`jogo.x`", *onde* `x` *deve ser* `c`, `cpp`, `pas`, `java`, `js`, `py2` *ou* `py3`

O jogo de dominó tradicional, conhecido como duplo-6, possui 28 peças. Cada peça está dividida em dois quadrados e dentro de cada quadrado há entre 0 e 6 círculos. O jogo é chamado de duplo-6 justamente porque esse é o maior número de círculos que aparece num quadrado de uma peça. A figura ao lado mostra uma forma de organizar as 28 peças do jogo duplo-6 em 7 linhas. Essa figura permite ver claramente quantas peças haveria num jogo de dominó, por exemplo, do tipo duplo-4: seriam todas as peças das 5 primeiras linhas, 15 peças no total. Também poderíamos ver, seguindo o padrão da figura, quantas peças possui o jogo de dominó conhecido como mexicano, que é o duplo-12. Seriam 91 peças, correspondendo a 13 linhas.



Neste problema, estamos precisando da sua ajuda para escrever um programa que, dado o valor N, calcule e imprima quantas peças existem num jogo de dominó do tipo duplo-N.

Entrada

A primeira linha da entrada contém um número natural N representando o tipo do jogo de dominó: duplo-N.

Saída

Seu programa deve imprimir uma linha contendo um número natural representando quantas peças existem num jogo de dominó do tipo duplo-N.

Restrições

- $0 \leq N \leq 10000$

Exemplos

Exemplo de entrada 1	**Exemplo de saída 1**
6	28

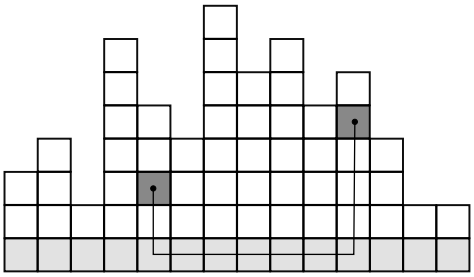
Exemplo de entrada 2	**Exemplo de saída 2**
12	91

Distância entre amigos

Nome do arquivo: "`amigos.x`", *onde* `x` *deve ser* `c`, `cpp`, `pas`, `java`, `js`, `py2` *ou* `py3`

Ao longo da rua existem N prédios de largura igual, mas com número de andares diferentes. Quase toda a turma do colégio mora em algum apartamento desses prédios e eles resolveram definir a distância entre dois apartamentos quaisquer da rua para saber, ao final, qual par de colegas da turma mora mais longe um do outro.

Funciona assim: para um colega A visitar um colega B, que mora num prédio diferente, ele deve descer a andares até o térreo do seu prédio; depois andar para a esquerda ou direita, dependendo do lado para o qual seu colega mora, por p prédios; depois subir b andares até o apartamento do colega B. A distância entre A e B, então, será $a+p+b$. A figura mostra um exemplo, para $N = 14$, onde estão marcados dois andares de prédios diferentes para os quais a distância é 12.



Dado um número de andares de cada prédio ao longo da rua, seu programa deve computar a distância máxima possível entre dois apartamentos quaisquer na rua.

Entrada

A primeira linha da entrada contém um inteiro N representando o número de prédios na rua. A segunda linha contém N inteiros $A_i, 1 \leq i \leq N$, representando o número de andares de cada prédio, sem contar o térreo. Quer dizer, por exemplo, se $A_i = 19$, então quem mora no último andar precisa descer 19 andares até o térreo. Veja a figura, que corresponde ao primeiro exemplo de entrada abaixo.

Saída

Seu programa deve imprimir uma linha contendo um número inteiro representando a distância máxima possível entre dois apartamentos na rua.

Restrições

- $2 \leq N \leq 200000 (2 \times 10^5)$;
- $1 \leq A_i \leq 10^9$ para todo $1 \leq i \leq N$.

Informações sobre a pontuação

- Em um conjunto de casos de teste somando 25 pontos, $N \leq 10^4$ e $A_i \leq 10^4$
- Em um conjunto de casos de teste somando 25 pontos, $A_i \leq 100$
- Em um conjunto de casos de teste somando 50 pontos, nenhuma restrição adicional

Exemplos

Exemplo de entrada 1	**Exemplo de saída 1**
14 2 3 1 6 4 3 7 5 6 4 5 3 1 1	18

Exemplo de entrada 2	**Exemplo de saída 2**
6 1 1 4 3 1 2	9

Exemplo de entrada 3	**Exemplo de saída 3**
2 1 1	3