



OBI2012

Caderno de Tarefas

Modalidade **Programação • Nível 1, Fase 1**

14 de abril de 2012

A PROVA TEM DURAÇÃO DE 4 HORAS

Promoção:



Sociedade Brasileira de Computação

Patrocínio:



Fundação Carlos Chagas

Instruções

LEIA ATENTAMENTE ESTAS INSTRUÇÕES ANTES DE INICIAR A PROVA

- Este caderno de tarefas é composto por 4 páginas (não contando a folha de rosto), numeradas de 1 a 4. 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.
- 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 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; e soluções na linguagem Python devem ser arquivos com sufixo *.py*. Para problemas diferentes você pode escolher trabalhar com linguagens diferentes, mas apenas uma solução, em uma única linguagem, deve ser submetida para cada problema.
- Ao final da prova, para cada solução que você queira submeter para correção, copie o arquivo fonte para o seu diretório de trabalho ou disquete, 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*, *print*, *write*
- Procure resolver o problema 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.

Vice-campeão

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

A OBI (Organização de Bocha Internacional) é responsável por organizar a competição mundial de bocha. Infelizmente esse esporte não é muito popular, e numa tentativa de aumentar a sua popularidade, ficou decidido que seriam chamados, para a Grande Final Mundial, o **campeão** e o **vice-campeão** de cada sede nacional, ao invés de apenas o **primeiro lugar**.

Tumbólia é um país pequeno que já havia realizado a sua **competição** nacional quando a nova regra foi instituída, e o comitê local não armazenou quem foi o **segundo classificado**. Felizmente eles armazenaram a **pontuação** de todos **competidores** – que foram apenas três, devido ao tamanho diminuto do país. Sabe-se também que as **pontuações** de todos jogadores foram diferentes, de forma que não ocorreu **empate** entre nenhum deles.

Resta agora descobrir quem foi o vice-campeão e para isso o comitê precisa de ajuda.

Entrada

A primeira e única linha da entrada consiste de três inteiros separados por espaços, A , B e C , as pontuações dos 3 competidores.

Saída

Imprima uma única linha na saída, contendo apenas um número inteiro, a pontuação do vice-campeão.

Restrições

- $1 \leq A \leq 100$
- $1 \leq B \leq 100$
- $1 \leq C \leq 100$

Exemplos

Entrada	Saída
4 5 6	5
10 5 9	9

Corrida

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

Leonardo é um corredor profissional que participa de diversos campeonatos de atletismo pelo mundo. O tamanho das pistas ao redor do mundo não é padronizado. Por isso, Leonardo, que treina em um clube que possui uma pista circular, resolveu fixar seu treinamento em C metros, ao invés de um número fixo de voltas na pista. Após cada treinamento, Leonardo deve tomar meio litro de água antes de fazer qualquer esforço, e por isso quer deixar sua garrafa de água exatamente no ponto da pista onde ele termina o seu treinamento.

Sabendo o comprimento da pista de corrida que Leonardo pretende treinar, ele resolveu pedir sua ajuda para calcular o local do ponto de término do treinamento. O ponto de término é o local da pista onde ele termina o percurso de C metros considerando que ele parte do ponto de partida e se movimenta sempre na mesma direção. O ponto de término é dado pelo número de metros entre o ponto de partida e o local onde Leonardo termina seu treinamento, contados na direção do percurso. Leonardo quer deixar sua garrafa de água neste ponto.

Por exemplo, se a pista tem 12 metros e Leonardo fixou seu treinamento em 22 metros, o ponto de término é 10.

Sua tarefa é, dado o número C de metros que Leonardo pretende correr e o comprimento N em metros da pista circular, determinar o ponto de término de seu treinamento.

Entrada

A entrada consiste em apenas uma linha contendo dois inteiros C e N que indicam, respectivamente, o número de metros que Leonardo pretende correr e o comprimento da pista.

Saída

Seu programa deve imprimir apenas uma linha, contendo apenas um inteiro, indicando o ponto de término do treinamento de Leonardo.

Restrições

- $1 \leq C \leq 10^8$
- $1 \leq N \leq 100$

Exemplos

Entrada	Saída
7000 100	0

Entrada	Saída
918 76	6

Consecutivos

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

Num sorteio que distribui prêmios, um participante inicialmente sorteia um inteiro N e depois N valores. O número de **pontos** do participante é o **tamanho da maior sequência de valores consecutivos iguais**. Por exemplo, suponhamos que um participante sorteia $N = 11$ e, nesta ordem, os valores

30, 30, 30, 30, 40, 40, 40, 40, 40, 30, 30

Então, o participante ganha 5 **pontos**, correspondentes aos 5 **valores** 40 **consecutivos**. Note que o participante sorteou 6 valores iguais a 30, mas nem todos são **consecutivos**.

Sua tarefa é ajudar a organização do evento, escrevendo um programa que determina o número de **pontos** de um participante.

Entrada

A primeira linha da entrada contém um inteiro N , o número de valores sorteados. A segunda linha contém N valores, V_1, V_2, \dots, V_N , na ordem de sorteio, separados por um espaço em branco.

Saída

Seu programa deve imprimir apenas uma linha, contendo apenas um inteiro, indicando o número de pontos do participante.

Restrições

- $1 \leq N \leq 10^4$
- $-2^{31} \leq V_I \leq 2^{31} - 1$, para $I = 1, 2, \dots, N$

Exemplos

Entrada	Saída
11 30 30 30 40 40 40 40 40 30 30 30	5
Entrada	Saída
14 1 1 1 20 20 20 20 3 3 3 3 3 3 3	7