



OBI2015

Caderno de Tarefas

Modalidade **Programação** • **Nível Júnior** • **Fase 1**

29 de maio de 2015

A PROVA TEM DURAÇÃO DE 3 HORAS

Promoção:



Sociedade Brasileira de Computação

Apoio:



Instruções

LEIA ATENTAMENTE ESTAS INSTRUÇÕES ANTES DE INICIAR A PROVA

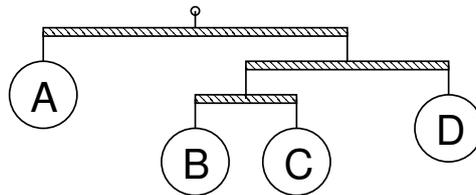
- Este caderno de tarefas é composto por 4 páginas (não contando a folha de rosto), numeradas de 1 a 4. Verifique se o caderno está completo.
- A prova deve ser feita individualmente.
- É proibido consultar a Internet, livros, anotações ou qualquer outro material durante a prova. É permitida a consulta 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; soluções na linguagem Python devem ser arquivos com sufixo *.py*; e soluções na linguagem Javascript devem ter arquivos com sufixo *.js*. 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 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 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.

MóBILE

Nome do arquivo: `mobile.c`, `mobile.cpp`, `mobile.pas`, `mobile.java`, `mobile.js` ou `mobile.py`

O móbile na sala da Maria é composto de três hastes exatamente como na figura abaixo. Para que ele esteja completamente equilibrado, com todas as hastes na horizontal, os pesos das quatro bolas A , B , C e D têm que satisfazer todas as seguintes três condições:

1. $A = B + C + D$; e
2. $B + C = D$; e
3. $B = C$.



Nesta tarefa, dados os pesos das quatro bolas, seu programa deve decidir se o móbile está ou não completamente equilibrado.

Entrada

A entrada consiste de quatro linhas contendo, cada uma, um número inteiro, indicando os pesos das bolas. Os números são dados na ordem: A , B , C e D .

Saída

Seu programa deve escrever uma única linha na saída, contendo o caractere “S” se o móbile estiver equilibrado, ou o caractere “N” se não estiver equilibrado.

Restrições

- $1 \leq A, B, C, D \leq 1000$

Exemplos

<p>Entrada</p> <p>12 3 3 6</p>	<p>Saída</p> <p>S</p>
<p>Entrada</p> <p>2002 560 560 882</p>	<p>Saída</p> <p>N</p>

Fita Colorida

Nome do arquivo: `fita.c`, `fita.cpp`, `fita.pas`, `fita.java`, `fita.js` ou `fita.py`

Roberto tem um conjunto de lápis com 10 tons diferentes de uma mesma cor, numerados de 0 a 9. Numa fita quadriculada, alguns quadrados foram coloridos inicialmente com o tom 0. Roberto precisa determinar, para cada quadrado Q não colorido, qual é a distância dele para o quadrado mais próximo de tom 0. A distância entre dois quadrados é definida com o número mínimo de movimentos para a esquerda, ou para a direita, para ir de um quadrado para o outro. O quadrado Q , então, deve ser colorido com o tom cuja numeração corresponde à distância determinada. Se a distância for maior ou igual a 9, o quadrado deve ser colorido com o tom 9. Seu programa deve colorir e imprimir a fita quadriculada dada na entrada.

Entrada

A primeira linha da entrada contém apenas um inteiro N , indicando o número de quadrados da fita. A segunda linha contém N números inteiros: “-1” se o quadrado não está colorido, e “0” se está colorido com o tom 0.

Saída

Seu programa deve escrever na saída a fita totalmente colorida, de acordo com a regra definida acima.

Restrições

- $3 \leq N \leq 10000$;
- Sempre existe pelo menos um “0” inicialmente na fita.

Informações sobre a pontuação

- Em um conjunto de casos de teste somando 80 pontos, $N \leq 1000$

Exemplos

Entrada 8 -1 -1 0 -1 -1 -1 0 -1	Saída 2 1 0 1 2 1 0 1
Entrada 13 -1 0 -1 -1 -1 -1 -1 -1 -1 -1 0 -1 -1	Saída 1 0 1 2 3 4 4 3 2 1 0 1 2
Entrada 6 0 -1 -1 -1 -1 -1	Saída 0 1 2 3 4 5

Prêmio do Milhão

Nome do arquivo: premio.c, premio.cpp, premio.pas, premio.java, premio.js ou premio.py

Alice e Bia criaram uma página na Internet com informações sobre o Macaco-prego-de-peito-amarelo, uma espécie em extinção. A página mostra como todos podem ajudar a manter o habitat natural para evitar que a espécie seja extinta.

Uma empresa gostou tanto da iniciativa de Alice e Bia que prometeu doar um prêmio para que as duas amigas possam realizar outras iniciativas semelhantes. A empresa decidiu que o prêmio seria dado quando a soma do número de acessos à página chegasse a 1 milhão.

Dada a lista de acessos diários que ocorreram à página de Alice e Bia, escreva um programa para determinar quantos dias foram necessários para a soma dos acessos chegar a 1 milhão e as amigas ganharem o prêmio.

Entrada

A primeira linha da entrada contém um número inteiro N , que indica o número de dias que a lista contém. Cada uma das linhas seguintes contém um único inteiro A , o número de acessos em um dia. O primeiro número dado indica o número de acessos no primeiro dia, o segundo número dado indica o número de acessos no segundo dia, e assim por diante.

Saída

Seu programa deve escrever na saída uma única linha, contendo um único número inteiro, o número de dias que foram necessários para a soma dos acessos à página de Alice e Bia chegar a 1000000.

Restrições

- $1 \leq N \leq 10^3$, ou seja, a lista tem no máximo 1000 números
- $0 < A \leq 10^6$, ou seja, cada inteiro A da lista é positivo e menor do que ou igual a 1 milhão.
- A soma de todos os valores A da lista é maior do que ou igual a 1 milhão (ou seja, Alice e Bia certamente ganham o prêmio).

Exemplos

Entrada	Saída
5 100 99900 400000 500000 600000	4

Entrada	Saída
1 1000000	1