



OBI2013

Caderno de Tarefas

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

18 de maio de 2013

A PROVA TEM DURAÇÃO DE 3 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.

Saldo do Vovô

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

Vovô João tem uma banca de jornais; ele tem muitos clientes, e diariamente recebe muito dinheiro, mas também faz muitos pagamentos para manter o seu estoque de jornais e revistas. Todo dia ele vai ao banco realizar um depósito ou uma retirada de dinheiro. Em alguns dias, o saldo de sua conta no banco fica negativo, mas Vovô João tem um acordo com o banco que garante que ele somente é cobrado se o saldo for menor do que um valor pré-estabelecido.

Dada a movimentação diária da conta do banco do Vovô João, você deve escrever um programa que calcule o menor saldo da conta, no período dado.

Entrada

A primeira linha da entrada contém dois números inteiros N e S que indicam respectivamente o número de dias do período de interesse e o saldo da conta no início do período. Cada uma das N linhas seguintes contém um número inteiro indicando a movimentação de um dia (valor positivo no caso de depósito, valor negativo no caso de retirada). A movimentação é dada para um período de N dias consecutivos: a primeira das N linhas corresponde ao primeiro dia do período de interesse, a segunda linha corresponde ao segundo dia, e assim por diante.

Saída

Seu programa deve imprimir uma única linha, contendo um único número inteiro, o menor valor de saldo da conta no período dado.

Restrições

- $1 \leq N \leq 30$
- $-10^3 \leq S \leq 10^3$
- $-10^3 \leq \text{cada movimentação} \leq 10^3$

Exemplos

Entrada 3 1000 100 -800 50	Saída 300
Entrada 6 -200 -100 1000 -2000 100 -50 2000	Saída -1300

Tomadas

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

A Olimpíada Internacional de Informática (IOI, no original em inglês) é a mais prestigiada competição de programação para alunos de ensino médio; seus aproximadamente 300 competidores se reúnem em um país diferente todo ano para os dois dias de prova da competição. Naturalmente, os competidores usam o tempo livre para acessar a Internet, programar e jogar em seus notebooks, mas eles se depararam com um problema: o saguão do hotel só tem uma tomada.

Felizmente, os quatro competidores da equipe brasileira da IOI trouxeram cada um uma régua de tomadas, permitindo assim ligar vários notebooks em uma tomada só; eles também podem ligar uma régua em outra para **aumentar** ainda mais o número de tomadas disponíveis. No entanto, como as réguas têm muitas tomadas, eles pediram para você escrever um programa que, dado o número de tomadas em cada régua, determina quantas tomadas podem ser **disponibilizadas** no saguão do hotel.

Entrada

A entrada consiste de uma linha com quatro inteiros positivos T_1, T_2, T_3, T_4 , indicando o número de tomadas de cada uma das quatro réguas.

Saída

Seu programa deve imprimir uma única linha contendo um único número inteiro, indicando o número máximo de notebooks que podem ser conectados num mesmo instante.

Restrições

- $2 \leq T_i \leq 6$.

Exemplos

Entrada 2 4 3 2	Saída 8
Entrada 6 6 6 6	Saída 21
Entrada 2 2 2 2	Saída 5

Capital

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

O governo do estado de Queensland está com problemas sérios de trânsito na capital Brisbane, onde estão os prédios administrativos. Para desafogar o trânsito, o prefeito de Brisbane e o governador de Queensland decidiram que uma nova capital administrativa deve ser construída em uma área fora de Brisbane. Para projetar a nova capital, o renomado arquiteto minimalista Joe Bloggs foi contratado.

Bloggs foi informado de que o terreno destinado à nova capital ainda não foi demarcado, mas será retangular. Além disso, a cidade deverá ser dividida em quatro zonas, uma delas destinada a uma reserva ambiental e cada uma das outras três receberá os novos prédios de cada um dos três poderes (Executivo, Legislativo e Judiciário). Em um arroubo de criatividade, Bloggs decidiu que duas avenidas, **perpendiculares** entre si, cada uma paralela a dois dos lados do terreno retangular, dividirão a capital nas quatro zonas.

Bloggs recebeu do governo as áreas de cada uma das zonas e, após muito esforço, encontrou um **retângulo** que pode ser dividido conforme seus planos e de forma a respeitar as áreas delimitadas. No entanto, a Fundação de Conservação dos Cangurus determinou que a área destinada à reserva ambiental era muito pequena, o que obrigou o governo a alterar as áreas das quatro zonas. Após receber as novas medidas, Bloggs tentou encontrar um novo retângulo que viabilizasse seu projeto, porém sem sucesso. Cansado de fazer testes, ele pensou que talvez tenha que abandonar sua brilhante ideia. Por isso, ele pediu para você escrever um programa que, dadas as áreas das quatro zonas, determine se ele poderá ou não manter seu projeto (ou seja, se existe um retângulo que possa ser dividido por **duas retas perpendiculares**, **cada uma paralela a dois dos lados do retângulo**, tal que as **quatro áreas formadas obedeçam às exigências** do governo).

Entrada

A entrada consiste de uma única linha contendo quatro inteiros A_1 , A_2 , A_3 , A_4 , indicando a área de cada uma das zonas.

Saída

Imprima uma única linha contendo um único caractere: ‘S’ se Bloggs pode preservar seu projeto e ‘N’ caso contrário.

Restrições

- $1 \leq A_i \leq 10^4$

Exemplos

Entrada 1 2 4 8	Saída S
Entrada 1 2 3 4	Saída N
Entrada 15 14 6 35	Saída S