



Olimpíada Brasileira de Informática

OBI2025

Caderno de Tarefas

Modalidade Programação • Nível 1 • Fase 1

12 a 14 de Junho de 2025

A PROVA TEM DURAÇÃO DE 2 horas

Promoção:



Sociedade Brasileira de Computação

Apoio:



Coordenação:



Instruções

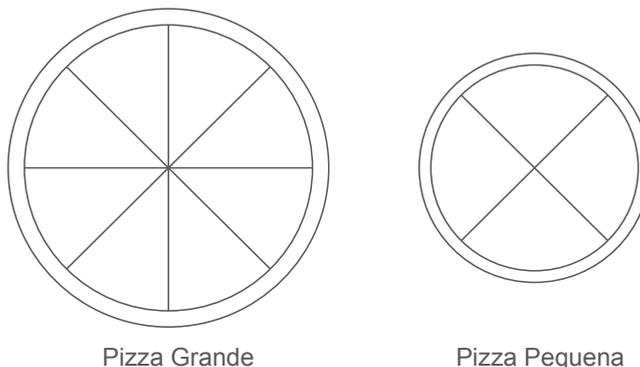
LEIA ATENTAMENTE ESTAS INSTRUÇÕES ANTES DE INICIAR A PROVA

- 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 Java devem ser arquivos com sufixo *.java* e a classe principal deve ter o mesmo nome do arquivo fonte; soluções na linguagem Python 3 devem ser arquivos com sufixo *.py*; 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).
- Você pode submeter até 50 soluções para cada tarefa. A pontuação total de cada tarefa é a melhor pontuação entre todas as submissões. Se a tarefa tem sub-tarefas, para cada sub-tarefa é considerada a melhor pontuação entre todas as submissões.
- 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 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.

Pizzaria

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

A mãe de Larissa decidiu comemorar o aniversário da filha em uma pizzaria. Nessa pizzaria, existem pizzas de dois tamanhos: uma pizza grande possui 8 fatias, enquanto que uma pizza pequena possui 4 fatias.



A pizzaria informou que os pedidos devem ser feitos com antecedência. Por isso, a mãe de Larissa já fez um pedido indicando quantas pizzas de cada tamanho ela vai comprar.

Larissa está empolgada em convidar seus amigos para a festa. Porém, como ela tem muitos amigos, é possível que o pedido feito não possua fatias de pizza suficientes para servir ela, sua mãe e todos os seus amigos. Assim, Larissa gostaria de saber, no máximo, quantos amigos ela pode convidar de modo que todos os convidados (ela, sua mãe e seus amigos) recebam uma fatia de pizza.

Por exemplo, suponha que a mãe de Larissa pediu 2 pizzas grandes e 3 pizzas pequenas. No total, haverão $2 \times 8 + 3 \times 4 = 28$ fatias de pizza. Após reservar uma fatia para Larissa e uma para sua mãe, sobram $28 - 2 = 26$ fatias de pizza. Assim, Larissa poderá convidar 26 amigos para sua festa.

Sua tarefa é: dadas a quantidade de pizzas grandes e a quantidade de pizzas pequenas que a mãe de Larissa pediu, calcule o número máximo de **amigos** que Larissa pode convidar para a festa, ou seja, o número de fatias que sobram após reservar uma fatia para Larissa e uma para sua mãe.

Entrada

A entrada possui duas linhas. A primeira linha contém um inteiro G , indicando a quantidade de pizzas grandes que a mãe de Larissa pediu. A segunda linha contém um inteiro P , indicando a quantidade de pizzas pequenas que ela pediu.

Saída

Seu programa deverá imprimir uma única linha contendo um único inteiro: o número máximo de amigos que Larissa pode convidar para a festa, excluindo ela mesma e sua mãe.

Restrições

É garantido que todo caso de teste satisfaz as restrições abaixo.

- $1 \leq G \leq 10$
- $1 \leq P \leq 10$

Informações sobre a pontuação

A tarefa vale 100 pontos.

Exemplos

Exemplo de entrada 1 2 3	Exemplo de saída 1 26
Exemplo de entrada 2 7 4	Exemplo de saída 2 70
Exemplo de entrada 3 1 2	Exemplo de saída 3 14

Café com Leite

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

Felipe trabalha em uma cafeteria especializada em café com leite. O chefe dele criou uma promoção na qual os clientes recebem um desconto caso tragam suas próprias xícaras, evitando o uso de materiais descartáveis. A promoção se tornou muito popular, o que é ótimo para o meio ambiente mas dificultou o trabalho de Felipe, pois cada cliente possui uma xícara de um tamanho diferente. Além disso, cada cliente prefere quantidades diferentes de leite na bebida.

Ao fazer um pedido, o cliente indica para Felipe dois números: o volume mínimo A (em mililitros) e o volume máximo B (em mililitros) de leite que ele deseja em sua bebida. O cliente indica também a capacidade C (também em mililitros) de sua xícara.

Para preparar o pedido, Felipe insere a xícara na máquina de espresso, que prepara D mililitros de café. Ele usa a máquina somente uma vez, isto é, o volume de café na xícara sempre será exatamente D . Depois de remover a xícara da máquina, Felipe adiciona leite de modo a enchê-la completamente, ou seja, o volume total de café com leite é exatamente C .

Felipe gostaria de saber se o volume de leite na xícara está dentro das preferências do cliente. Por exemplo, suponha que a máquina prepare $D = 30$ ml de café e considere dois clientes:

- O cliente 1 possui uma xícara com capacidade $C = 170$ ml e deseja que sua bebida possua entre $A = 130$ ml e $B = 150$ ml de leite. Neste caso, a bebida preparada por Felipe possui 30 ml de café e $170 - 30 = 140$ ml de leite. O volume de leite está entre 130 e 150 ml, ou seja, dentro das preferências do cliente.
- O cliente 2 possui uma xícara com capacidade $C = 240$ ml e deseja que sua bebida possua entre $A = 220$ ml e $B = 230$ ml de leite. Neste caso, a bebida preparada por Felipe possui 30 ml de café e $240 - 30 = 210$ ml de leite. Portanto, o volume de leite está abaixo do volume mínimo especificado pelo cliente.

Escreva um programa para ajudar Felipe: dados os volumes A , B e C especificados por um cliente e o volume D de café preparado pela máquina, determine se o volume de leite na bebida atenderá às preferências do cliente.

Entrada

A entrada possui quatro linhas, cada uma contendo um único inteiro:

- a primeira linha contém o volume mínimo A de leite (em ml) que o cliente deseja;
- a segunda linha contém o volume máximo B de leite (em ml) que o cliente deseja;
- a terceira linha contém a capacidade C (em ml) da xícara;
- a quarta linha contém o volume D (em ml) de café preparado pela máquina.

Saída

Seu programa deverá imprimir uma única linha contendo um único caractere: caso Felipe consiga satisfazer as preferências do cliente, imprima o caractere **S** (a letra S maiúscula). Caso contrário, imprima o caractere **N** (a letra N maiúscula).

Restrições

É garantido que todo caso de teste satisfaz as restrições abaixo.

- $100 \leq C \leq 500$
- $0 \leq A \leq B < C$
- $10 \leq D \leq 100$

Informações sobre a pontuação

A tarefa vale 100 pontos.

Exemplos

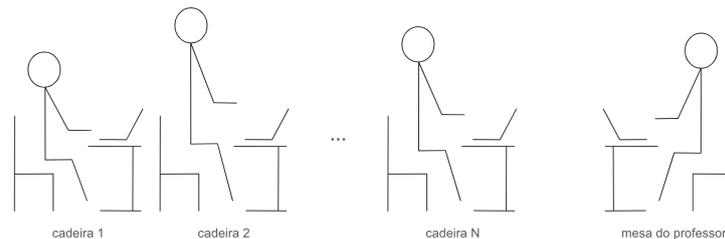
Exemplo de entrada 1 130 150 170 30	Exemplo de saída 1 S
Exemplo de entrada 2 220 230 240 30	Exemplo de saída 2 N
Exemplo de entrada 3 0 200 300 45	Exemplo de saída 3 N
Exemplo de entrada 4 120 120 200 80	Exemplo de saída 4 S

Fila

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

Após um longo semestre de aulas de informática, chegou a hora do professor Ricardo avaliar o conhecimento de seus alunos com uma prova surpresa. No entanto, ele conhece seus alunos e sabe que nem todos estão com a matéria em dia. Por isso, ele suspeita que alguns deles tentarão colar na prova, algo que ele não pode permitir!

O professor é muito bom em identificar trapagens e consegue reconhecê-las com apenas um olhar. No entanto, sua sala de aula é um tanto incomum. Nessa sala, as cadeiras dos N alunos estão dispostas em uma única fila, de forma que a cadeira 1 fica no fundo da sala e a cadeira N fica logo em frente à mesa de Ricardo, de onde ele vigia os alunos.



Os alunos possuem tamanhos variados, sendo que o aluno sentado na cadeira i tem a_i centímetros de altura. Por causa da organização estranha das cadeiras, essas alturas podem atrapalhar a visão do professor, de modo que Ricardo não consegue enxergar um aluno caso exista outro aluno com altura maior ou igual à dele que está mais perto da mesa do professor. Formalmente, Ricardo **não** consegue enxergar o aluno sentado na cadeira i se existe algum aluno em uma cadeira j tal que $j > i$ e $a_j \geq a_i$. Observe que o aluno na cadeira N sempre pode ser observado, pois está logo em frente ao professor.

Ricardo consegue monitorar os alunos que ele enxerga, mas está preocupado com possíveis colas entre os demais. Dadas as alturas dos alunos, ajude-o a saber qual o máximo de alunos que poderão colar na prova sem que ele perceba, isto é, a quantidade de alunos que ele não consegue enxergar.

Entrada

A primeira linha da entrada contém um único inteiro positivo N , a quantidade de alunos. A segunda linha da entrada contém N inteiros positivos a_1, a_2, \dots, a_N , as alturas dos alunos na ordem em que eles estão dispostos na sala.

Saída

A saída deve conter um único inteiro, a quantidade de alunos que Ricardo **não** consegue enxergar.

Restrições

É garantido que todo caso de teste satisfaz as restrições abaixo.

- $1 \leq N \leq 100\,000$
- $1 \leq a_i \leq 1\,000\,000\,000$ para todo $1 \leq i \leq N$

Informações sobre a pontuação

A tarefa vale 100 pontos. Estes pontos estão distribuídos em subtarefas, cada uma com suas **restrições adicionais** às definidas acima.

- **Subtarefa 1 (0 pontos):** Esta subtarefa é composta apenas pelos exemplos mostrados abaixo. Ela não vale pontos, serve apenas para que você verifique se o seu programa imprime o resultado correto para os exemplos.
- **Subtarefa 2 (60 pontos):** $N \leq 1\,000$.
- **Subtarefa 3 (40 pontos):** Sem restrições adicionais.

Exemplos

Exemplo de entrada 1 5 200 180 190 140 160	Exemplo de saída 1 2
Exemplo de entrada 2 5 180 180 170 150 190	Exemplo de saída 2 4
Exemplo de entrada 3 2 150 150	Exemplo de saída 3 1