

Competidor(a): _____

Número de inscrição: _____ (opcional)

Este Caderno de Tarefas não pode ser levado para casa após a prova. Após a prova entregue este Caderno de Tarefas para seu professor guardar. Os professores poderão devolver os Cadernos de Tarefas aos competidores após o término do período de aplicação das provas (20 e 21 de agosto de 2026).



Olimpíada Brasileira de Informática

OBI2026

Caderno de Tarefas

Modalidade Programação • Nível Mirim • Fase 2

20 e 21 de agosto de 2026

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.

Festas de Aniversário

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

Alice e Bruno são melhores amigos e estão muito animados com seus aniversários que se aproximam. Como cada um faz aniversário em um dia diferente, eles querem descobrir quem vai comemorar primeiro para começar logo os preparativos da festa.

Dados o dia e o mês do aniversário de Alice e o dia e o mês do aniversário de Bruno, calcule quem vai festejar primeiro no ano.

Entrada

A primeira linha da entrada contém um único inteiro D_A , o dia do aniversário de Alice.

A segunda linha da entrada contém um único inteiro M_A , o mês do aniversário de Alice.

A terceira linha da entrada contém um único inteiro D_B , o dia do aniversário de Bruno.

A quarta linha da entrada contém um único inteiro M_B , o mês do aniversário de Bruno.

Saída

A saída deve conter uma única letra: **A** se o aniversário de Alice acontece primeiro, ou **B** se o aniversário de Bruno acontece primeiro. Não coloque nada além disso na saída, senão será considerado errado.

Restrições

- $1 \leq D_A \leq 31$.
- $1 \leq M_A \leq 12$.
- $1 \leq D_B \leq 31$.
- $1 \leq M_B \leq 12$.
- É garantido que as datas são válidas (por exemplo, não haverá 31 de fevereiro).
- É garantido que as festas serão em datas diferentes.

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 (20 pontos):** $M_A = M_B$ (os aniversários são no mesmo mês).
- **Subtarefa 3 (20 pontos):** $D_A = D_B$ (os aniversários são no mesmo dia do mês, porém em meses diferentes).
- **Subtarefa 4 (60 pontos):** Sem restrições adicionais.

Exemplos

Exemplo de entrada 1	Exemplo de saída 1
7 9 2 11	A

Explicação do exemplo 1: Alice faz aniversário em 7 de setembro e Bruno faz aniversário em 2 de novembro. Nesse caso, a resposta é A.

Exemplo de entrada 2	Exemplo de saída 2
29 12 15 12	B

Explicação do exemplo 2: Alice faz aniversário em 29 de dezembro e Bruno faz aniversário em 15 de dezembro. Nesse caso, a resposta é B. Este exemplo satisfaz as restrições da subtarefa 2.

Exemplo de entrada 3	Exemplo de saída 3
10 3 10 8	A

Explicação do exemplo 3: Alice faz aniversário em 10 de março e Bruno faz aniversário em 10 de agosto. Nesse caso, a resposta é A. Este exemplo satisfaz as restrições da subtarefa 3.

Adivinha o Número

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

Na aula de matemática, os alunos estavam tão agitados que a professora Lúcia decidiu propor uma brincadeira para acalmá-los: **Adivinha o Número**

As regras são simples:

- A professora escolhe um número secreto P , sem revelar para ninguém.
- Os alunos Ana e Beto, ao mesmo tempo, chutam um número: Ana chuta A e Beto chuta B .
- A professora então revela o número P .
- Ganha quem chegou mais perto do número da professora. Se os dois ficarem à mesma distância, é empate.

Dados os chutes de Ana e Beto e o número escolhido pela professora, descubra quem ganhou a brincadeira.

Entrada

A primeira linha da entrada contém um único inteiro A , o chute de Ana.

A segunda linha da entrada contém um único inteiro B , o chute de Beto.

A terceira linha da entrada contém um único inteiro P , o número escolhido pela professora.

Saída

A saída deve conter uma única letra: **A** se Ana ganhou, **B** se Beto ganhou, ou **E** se houve empate. Não coloque nada além disso na saída, senão será considerado errado.

Restrições

- $1 \leq A \leq 100$.
- $1 \leq B \leq 100$.
- $1 \leq P \leq 100$.

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 (20 pontos):** $A \leq P \leq B$.
- **Subtarefa 3 (20 pontos):** $A \leq B$.
- **Subtarefa 4 (60 pontos):** Sem restrições adicionais.

Exemplos

Exemplo de entrada 1	Exemplo de saída 1
10 5 8	A

Explicação do exemplo 1: Ana chutou 10, Beto chutou 5 e a professora revelou o número 8. Nesse caso, a resposta é A.

Exemplo de entrada 2	Exemplo de saída 2
1 9 5	E

Explicação do exemplo 2: Ana chutou 1, Beto chutou 9 e a professora revelou o número 5. Nesse caso, a resposta é E. Este exemplo satisfaz as restrições das subtarefas 2 e 3.

Exemplo de entrada 3	Exemplo de saída 3
2 3 20	B

Explicação do exemplo 3: Ana chutou 2, Beto chutou 3 e a professora revelou o número 20. Nesse caso, a resposta é B. Este exemplo satisfaz as restrições da subtarefa 3.

Passes

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

Durante uma partida de futebol, uma câmera registra, a cada segundo, o número do jogador que está com a bola. Os jogadores do time 1 são identificados por números de 1 a 100, e os jogadores do time 2 são identificados por números de -1 a -100 .

Um passe acontece quando, em um segundo, a bola está com um jogador de determinado time e, no segundo seguinte, a bola passa para um **outro** jogador **do mesmo time**. Quando a bola vai de um jogador de um time para um jogador do outro time, isso **não** é considerado passe de nenhum dos dois times.

Por exemplo, se a sequência de posse de bola é 10, 9, 9, -3 , -2 , -2 , -1 , -7 , 3, 3, então:

- O time 1 realizou 1 passe: do jogador 10 para o jogador 9.
- O time 2 realizou 3 passes: de -3 para -2 , de -2 para -1 , e de -1 para -7 .

Note que os momentos em que a posse passou de um time para o outro (como de 9 para -3 , ou de -7 para 3) não contam como passe de nenhum time.

Dado o registro de N segundos da partida, determine quantos passes realizou cada time.

Entrada

A primeira linha da entrada contém um único inteiro N , a quantidade de segundos registrados.

A segunda linha da entrada contém N inteiros $J_1, J_2, \dots, J_N$, onde J_i é o número do jogador que está com a bola no segundo i . Números positivos identificam jogadores do time 1, e números negativos identificam jogadores do time 2.

Saída

A saída deve conter exatamente duas linhas. A primeira linha deve conter um inteiro com a quantidade de passes realizados pelo time 1. A segunda linha deve conter um inteiro com a quantidade de passes realizados pelo time 2. Imprima 0 caso um time não tenha realizado nenhum passe.

Restrições

- $3 \leq N \leq 10000$.
- $1 \leq J_i \leq 100$, para todo segundo i em que a bola está com um jogador do time 1.
- $-100 \leq J_i \leq -1$, para todo segundo i em que a bola está com um jogador do time 2.

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 (20 pontos):** $N = 3$.
- **Subtarefa 3 (20 pontos):** Todos os valores de J_i são positivos, ou seja, a bola ficou sempre com jogadores do time 1 durante todo o tempo registrado.
- **Subtarefa 4 (60 pontos):** Sem restrições adicionais.

Exemplos

Exemplo de entrada 1	Exemplo de saída 1
10 10 9 9 -3 -2 -2 -1 -7 3 3	1 3

Explicação do exemplo 1: Note que este é o exemplo mostrado no enunciado.

Exemplo de entrada 2	Exemplo de saída 2
3 5 -2 7	0 0

Explicação do exemplo 2: A bola passou de um time para o outro em todos os momentos, portanto nenhum time realizou passes. Este exemplo satisfaz as restrições da subtarefa 2.

Exemplo de entrada 3	Exemplo de saída 3
5 3 3 7 7 1	2 0

Explicação do exemplo 3: O time 1 realizou 2 passes: do jogador 3 para o jogador 7, e do jogador 7 para o jogador 1. Note que, nos momentos em que o mesmo jogador manteve a posse da bola por dois segundos consecutivos (3 e 3; 7 e 7), não houve passe. Este exemplo satisfaz as restrições da subtarefa 3.

Exemplo de entrada 4	Exemplo de saída 4
6 1 2 -1 -2 5 -3	1 1