



OBI

Olimpíada Brasileira de Informática
Simulado 01

Caderno de Tarefas

Modalidade Programação

23 de Abril de 2026

A PROVA TEM DURAÇÃO DE 2 HORAS

Apoio institucional:



Instruções

LEIA ATENTAMENTE ESTAS INSTRUÇÕES ANTES DE INICIAR A PROVA

- Este caderno de tarefas é composto por 8 páginas (não contando a folha de rosto), numeradas de 1 a 8. 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.
- As tarefas têm o mesmo valor na correção.
- 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, nem utilize qualquer rotina para limpar a tela ou posicionar o cursor.
- As tarefas **não** estão necessariamente ordenadas por ordem de dificuldade; procure resolver primeiro as questões mais fáceis.
- Preste muita atenção no nome dos arquivos fonte indicados nas tarefas. Soluções em **Python 3** devem ser arquivos com sufixo `.py`.
- Não utilize arquivos para entrada ou saída. Todos os dados devem ser lidos da entrada padrão e escritos na saída padrão. Utilize as funções padrão:
 - em C: `scanf`, `printf`;
 - em C++: `cin`, `cout`;
 - em Java: `Scanner`, `System.out.println`;
 - em Python: `input`, `print`;
 - em Javascript: `scanf`, `printf`;
- Procure resolver a tarefa de maneira eficiente. Na correção, eficiência também será levada em conta.

Divisão do Tesouro

Nome do arquivo: “tesouro.x”, onde x deve ser *c*, *cpp*, *java*, *js* ou *py*

O Capitão Olho Roxo e seus marinheiros encontraram uma arca com uma grande quantidade de moedas de ouro idênticas. Para a divisão das moedas, todos concordaram com a seguinte sugestão do Capitão:

- cada marinheiro exceto o Capitão deveria receber exatamente o mesmo número de moedas; e
- o Capitão deveria receber o dobro de moedas que um marinheiro recebe.

Pode ser que o fato de o Capitão ser o único com uma pistola a bordo tenha contribuído para a concordância de todos, mas também contribuiu o fato de que na forma proposta a divisão era perfeita, não sobrando ou faltando moedas.

Dados o número de moedas na arca e o número de marinheiros, escreva um programa para determinar quantas moedas o Capitão Olho Roxo recebeu.

Entrada

A primeira linha da entrada contém um número inteiro A , o número de moedas na arca. A segunda linha contém um inteiro N , o número de marinheiros (não contando o Capitão).

Saída

Seu programa deve produzir na saída uma única linha, contendo um único inteiro, o número de moedas que o Capitão Olho Roxo deve receber.

Restrições

- $3 \leq A \leq 10000$
- $1 \leq N \leq 1000$

Exemplo de entrada 1 221 11	Exemplo de saída 1 34
--	---------------------------------

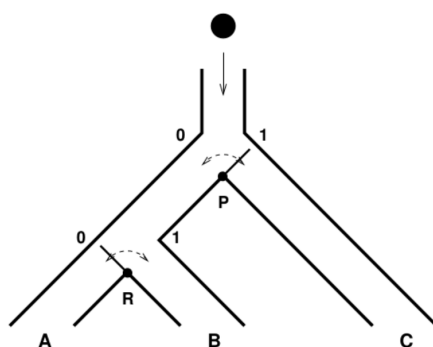
Exemplo de entrada 2 1000 8	Exemplo de saída 2 200
--	----------------------------------

Exemplo de entrada 3 3 1	Exemplo de saída 3 2
---------------------------------------	--------------------------------

Flíper

Nome do arquivo: *fliper.x*, onde *x* deve ser *c*, *cpp*, *java*, *js* ou *py*

Flíper é um tipo de jogo onde uma bolinha de metal cai por um labirinto de caminhos até chegar na parte de baixo do labirinto. A quantidade de pontos que o jogador ganha depende do caminho que a bolinha seguir. O jogador pode controlar o percurso da bolinha mudando a posição de algumas portinhas do labirinto. Cada portinha pode estar na posição 0, que significa virada para a esquerda, ou na posição 1 que quer dizer virada para a direita. Considere o flíper da figura abaixo, que tem duas portinhas. A portinha *P* está na posição 1 e a portinha *R*, na posição 0. Desse jeito, a bolinha vai cair pelo caminho *B*.



Você deve escrever um programa que, dadas as posições das portinhas *P* e *R*, neste flíper da figura, diga por qual dos três caminhos, A, B ou C, a bolinha vai cair!

Entrada

A entrada é composta por apenas uma linha contendo dois números *P* e *R*, indicando as posições das duas portinhas do flíper da figura.

Saída

A saída do seu programa deve ser também apenas uma linha, contendo uma letra maiúscula que indica o caminho por onde a bolinha vai cair: “A”, “B” ou “C”.

Restrições

- O número *P* pode ser 0 ou 1. O número *R* pode ser 0 ou 1.

Exemplo de entrada 1 0	Exemplo de saída B
Exemplo de entrada 0 0	Exemplo de saída C

Jogo de par ou ímpar

Nome do arquivo: *jogo.x*, onde *x* deve ser *c*, *cpp*, *java*, *js* ou *py*

Dois amigos, Alice e Bob, estão jogando um jogo muito simples, em que um deles grita ou “par” ou “ímpar” e o outro imediatamente responde ao contrário, respectivamente “ímpar” ou “par”. Em seguida, ambos exibem ao mesmo tempo uma mão cada um, em que alguns dedos estão estendidos e outros dobrados. Então eles contam o número total de dedos estendidos. Se a soma for par, quem gritou “par” ganha. Se a soma for ímpar, quem gritou “ímpar” ganha.

Por exemplo, suponhamos que a Alice gritou “par” e o Bob respondeu “ímpar”. Em seguida, Alice não deixou nenhum dos seus dedos estendidos, ao passo que Bob deixou três dedos estendidos. A soma então é três, que é ímpar, portanto Bob ganhou.

Seu programa deve determinar quem ganhou, tendo a informação de quem gritou par e o número de dedos estendidos de cada um.

Entrada

A entrada contém três linhas, cada uma com um número inteiro, P , D_1 e D_2 , nesta ordem. Se $P = 0$ então Alice gritou “par”, ao passo que se $P = 1$ então Bob gritou “par”. Os números D_1 e D_2 indicam, respectivamente, o número de dedos estendidos da Alice e do Bob.

Saída

Seu programa deverá imprimir uma única linha, contendo um único número inteiro, que deve ser 0 se Alice foi a ganhadora, ou 1 se Bob foi o ganhador.

Restrições

- $P = 0$ ou $P = 1$
- $0 \leq D_1 \leq 5$
- $0 \leq D_2 \leq 5$

Entrada	Saída
0	1
0	
3	

Entrada	Saída
1	0
0	
3	

Entrada	Saída
0	0
1	
5	

Prêmio

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

Uma ONG (Organização Não Governamental) oferece cursos gratuitos de programação de computadores, dança, música e culinária. Aproveitando a cozinha montada para os cursos de culinária, também vende pães integrais, doces e bolos para ajudar nas despesas.

O diretor da ONG anunciou um incentivo para a venda da produção da cozinha: considerando que cada pão vale 1 ponto, cada doce vale 2 pontos e cada bolo vale 3 pontos, os colaboradores ganharão um prêmio dependendo da soma total dos pontos dos produtos vendidos durante a semana.

Se a soma dos pontos de todos os produtos vendidos na semana for igual ou maior do que 150, cada colaborador recebe um bolo como prêmio; senão, se a soma dos pontos for maior ou igual a 120, cada colaborador recebe um doce como prêmio; senão, se a soma dos pontos for maior ou igual a 100, cada colaborador recebe um pão como prêmio. Se a soma dos pontos for menor do que 100 não há prêmio para os colaboradores.

Sabendo que você fez um curso de programação na ONG, o diretor pediu que você escreva um programa que, dados os números de pães, doces e bolos vendidos na semana, determine qual o prêmio merecido.

Entrada

A primeira linha contém um inteiro P , o número de pães vendidos na semana. A segunda linha contém um inteiro D , o número de doces vendidos na semana. A terceira e última linha contém um inteiro B , o número de bolos vendidos na semana.

Saída

Seu programa deve produzir uma única linha, contendo um único caractere, indicando o prêmio merecido: a letra maiúscula ‘P’ para pão, a letra maiúscula ‘D’ para doce, a letra maiúscula ‘B’ para bolo e a letra maiúscula ‘N’ se os colaboradores não merecem prêmio na semana.

Restrições

- $0 \leq P \leq 100$
- $0 \leq D \leq 100$
- $0 \leq B \leq 100$

Entrada	Saída
100	D
10	
4	

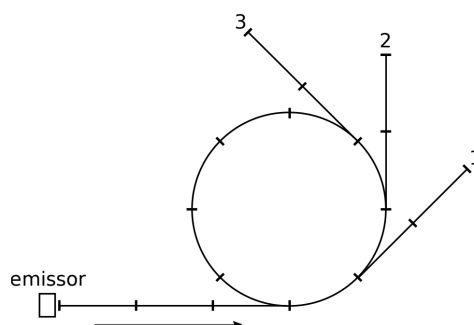
Entrada	Saída
30	B
45	
10	

Entrada	Saída
30	N
10	
5	

Acelerador de Partículas

Nome do arquivo: *acelerador.x*, onde *x* deve ser *c*, *cpp*, *java*, *js* ou *py*

A universidade está inaugurando um grande acelerador de partículas, com um emissor e três sensores, numerados 1, 2 e 3. Uma partícula, após sair do emissor, entra no acelerador onde pode dar várias voltas sendo acelerada a velocidades muito altas. Num determinado momento, a partícula sai do acelerador por uma das três saídas, atingindo um dos sensores. A figura mostra o caminho por onde as partículas trafegam, com uma graduação de 1 quilômetro. Por exemplo, do emissor até o acelerador são 3 quilômetros e a circunferência do acelerador tem 8 quilômetros.



Neste problema, será dada a distância total, em quilômetros, percorrida por uma certa partícula trafegando do emissor até algum sensor e seu programa deve determinar qual sensor foi atingido pela partícula. Por exemplo, veja que se a distância total for 23 quilômetros, então a partícula tem que ter atingido o sensor 2.

Entrada

A entrada consiste de apenas uma linha contendo um inteiro D , representando a distância total percorrida pela partícula.

Saída

Seu programa deve imprimir uma linha contendo um inteiro, representando o número do sensor que a partícula atingiu.

Exemplo de entrada 1 23	Exemplo de saída 1 2
Exemplo de entrada 2 6	Exemplo de saída 2 1
Exemplo de entrada 3 9192	Exemplo de saída 3 3