



OBI2012

Caderno de Tarefas

Modalidade **Programação • Nível Júnior**, Fase **2**

26 de maio de 2012

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.

Chocolate

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

Por lei, na Nlogônia todas as barras de chocolate são quadradas. Anamaria tem uma barra quadrada de chocolate de lado L , que ela quer **compartilhar** com alguns colegas da OBI. Mas ela é uma boa cidadã e cumpre a lei. Então, ela **divide** a barra em quatro pedaços quadrados, de lado $L/2$. Depois, ela repete esse procedimento com cada pedaço gerado, sucessivamente, **enquanto** o lado for **maior do que, ou igual a** 2cm. Você deve escrever um programa que, dado o lado L da barra inicial, em centímetros, determina quantos pedaços haverá ao **final do processo**.

Entrada

A entrada consiste de uma linha, com um único inteiro, L , o número de centímetros do lado do quadrado.

Saída

Se programa deve imprimir uma única linha, contendo um único inteiro, igual ao número total de pedaços obtidos pela Anamaria.

Restrições

- $2 \leq L \leq 10^4$

Informações sobre a pontuação

Em um conjunto de casos de teste totalizando 30 pontos, $L < 64$.

Exemplos

Entrada 4	Saída 16
Entrada 9	Saída 64
Entrada 2	Saída 4

Tira-teima

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

Uma quadra de tênis tem o formato de um retângulo, cujos lados medem 36 pés por 78 pés, que correspondem a um retângulo de 432 polegadas por 936 polegadas. No último Grand Slam da Austrália, Rafael Nadal perdeu para Novak Djoković, num dos jogos mais bonitos de tênis dos últimos tempos.

Muitas vezes, uma jogada é tão rápida, e a bola tão próxima da borda da quadra, que o juiz pode tomar uma decisão que pode ser contestada por um dos jogadores. Para isso, existe o *tira-teima*, que utiliza a imagem gravada do jogo para decidir se a bola estava dentro ou fora da metade da quadra correspondente a um dos jogadores.

Considere que a semi-quadra de Rafael Nadal corresponde a um retângulo em que dois vértices têm coordenadas (0,0) e (432, 468), onde todos os números são em polegadas.

Você deve escrever um programa para, dadas as coordenadas (X, Y) do ponto de contato da bola com o solo, determinar se uma bola bateu no solo dentro ou fora da semi-quadra. Note que se a bola bate na linha divisória ela é considerada uma bola dentro.

Entrada

A entrada é dada em uma única linha, que contém dois inteiros X e Y , que correspondem às coordenadas do ponto (X, Y) de contato da bola com o solo, em polegadas.

Saída

Seu programa deve imprimir uma única linha, contendo a palavra **dentro** se a bola bateu dentro da semi-quadra, e a palavra **fora** caso contrário.

Restrições

- $-500 \leq X, Y \leq 500$

Exemplos

Entrada -2 200	Saída fora
Entrada 432 10	Saída dentro

Receita de Bolo

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

João deseja fazer bolos para seus amigos, usando uma receita que indica que devem ser usadas 2 xícaras de farinha de trigo, 3 ovos e 5 colheres de sopa de leite. Em casa ele tem A xícaras de farinha de trigo, B ovos e C colheres de sopa de leite.

João não tem muita prática com a cozinha, e portanto ele só se arriscará a fazer medidas exatas da receita de bolo (por exemplo, se ele tiver material **suficiente** para fazer mais do que 2 e menos do que 3 bolos, ele fará somente 2 bolos). Sabendo disto, ajude João escrevendo um programa que determine qual a **quantidade máxima** de bolos que ele consegue fazer.

Entrada

A entrada é dada em uma única linha, que contém três números inteiros A , B e C , indicando respectivamente o número de xícaras de farinha de trigo, o número de ovos e o número de colheres de sopa de leite que João tem em casa.

Saída

Seu programa deve imprimir uma única linha, contendo um único inteiro, a **quantidade máxima** de bolos que João consegue fazer.

Restrições

- $1 \leq A \leq 100$
- $1 \leq B \leq 100$
- $1 \leq C \leq 100$

Exemplos

Entrada 4 6 10	Saída 2
Entrada 4 6 9	Saída 1