



OBI2012

Caderno de Tarefas

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

14 de abril 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.

Campeonato

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

Dois times, *Cormengo* e *Flaminthians*, participam de um campeonato de futebol, juntamente com outros times. Cada vitória conta três pontos, cada empate um ponto. Fica melhor classificado no campeonato um time que tenha mais pontos. Em caso de empate no número de pontos, fica melhor classificado o time que tiver maior saldo de gols. Se o número de pontos e o saldo de gols forem os mesmos para os dois times então os dois times estão empatados no campeonato.

Dados os números de vitórias e empates, e os saldos de gols dos dois times, sua tarefa é determinar qual dos dois está melhor classificado, ou se eles estão empatados no campeonato.

Entrada

A entrada é descrita em uma única linha, que contém seis inteiros, separados por um espaço em branco: $C_v, C_e, C_s, F_v, F_e, F_s$, que são, respectivamente, o número de vitórias do Cormengo, o número de empates do Cormengo, o saldo de gols do Cormengo, o número de vitórias do Flaminthians, o número de empates do Flaminthians e o saldo de gols do Flaminthians.

Saída

Seu programa deve imprimir uma única linha. Se Cormengo é melhor classificado que Flaminthians, a linha deve conter apenas a letra 'C'; se Flaminthians é melhor classificado que Cormengo, a linha deve conter apenas a letra 'F'; e se os dois times estão empatados a linha deve conter apenas o caractere '='.

Restrições

- $0 \leq C_v, C_e, F_v, F_e \leq 100$
- $-1000 \leq C_s, F_s \leq 1000$

Exemplos

Entrada 10 5 18 11 1 18	Saída C
Entrada 10 5 18 11 2 18	Saída =
Entrada 9 5 -1 10 2 10	Saída F

Busca na Internet

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

João fez uma pesquisa em seu site de busca predileto, e encontrou a resposta que estava procurando no terceiro link listado. Além disso, ele viu, pelo site, que t pessoas já haviam clicado neste link antes. João havia lido anteriormente, também na Internet, que o número de pessoas que clicam no segundo link listado é o dobro de número de pessoas que clicam no terceiro link listado. Nessa leitura, ele também descobriu que o número de pessoas que clicam no segundo link é a metade do número de pessoas que clicam no primeiro link.

João está intrigado para saber quantas pessoas clicaram no primeiro link da busca, e, como você é amigo dele, quer sua ajuda nesta tarefa.

Entrada

Cada caso de teste possui apenas um número, t , que representa o número de pessoas que clicaram no terceiro link da busca.

Saída

Para cada caso de teste imprima apenas uma linha, contendo apenas um inteiro, indicando quantas pessoas clicaram no primeiro link, nessa busca.

Restrições

- $1 \leq t \leq 1000$

Exemplos

Entrada	Saída
2	8

Entrada	Saída
25	100

Desafio do maior número

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

Leonardo é um garoto muito criativo. Ele adora criar desafios para seus colegas da escola. Seu último desafio é o seguinte: diversos números são ditos em voz alta, quando o número 0 (zero) é dito então o desafio termina e seus colegas devem dizer imediatamente qual foi o maior número. Leonardo tem muita dificuldade de verificar se a resposta dada pelos colegas é correta ou não, pois a sequência de números costuma ser longa. Por este motivo, ele resolveu pedir sua ajuda.

Sua tarefa é escrever um programa que dada uma sequência de números inteiros positivos terminada por 0 (zero), imprime o maior número da sequência.

Entrada

A entrada é dada em uma única linha contendo uma sequência de números inteiros positivos. O último número da linha é 0 (zero).

Saída

Seu programa deve imprimir o maior número dentre os números da entrada.

Restrições

- $1 \leq \text{tamanho da sequência} \leq 100$.
- $1 \leq \text{número da sequência} \leq 1000$.

Exemplos

Entrada	Saída
10 30 20 5 0	30

Entrada	Saída
99 1000 55 1 2 9 0	1000