

13^a Maratona de Programação

29 de março de 2014

Caderno de Problemas

Regras: *ACM International Collegiate Programming Contest*

Brasil – **Desafio ALGAR TELECOM**

(Este caderno contém 12 problemas; as páginas estão numeradas de 1 a 16)

Participantes:

CESUC
FAZU
FEIT-UEMG
FPU
IFTM Uberaba
IFTM Uberlândia
Pitágoras
UNIFEI
UFU
UNIPAM
UNITRI
UNIUBE

13ª Maratona de Programação – Instruções

Linhas de compilação:

Linguagem	Linha de comando de compilação
C	gcc -lm teste.c -oteste
C++	g++ -lm teste.cpp -oteste
Java	javac teste.java

Linhas de execução:

Linguagem	Linha de comando de execução
C	./teste < entrada.in
C++	./teste < entrada.in
Java	java teste < entrada.in

Exemplo de problema:

Problema Teste

Arquivo fonte: teste.c, teste.cpp ou teste.java.

Para um dado valor n , calcular a expressão $3n + 1$.

Entrada

A entrada contém várias linhas. Cada linha contém um inteiro n ($1 \leq n \leq 100$).

Saída

Para cada caso de teste imprima uma linha contendo o resultado da expressão.

Exemplo de entrada

8
3
12
2

Exemplo de saída

25
10
37
7

Exemplo de solução em C++ (teste.cpp):

```
#include <iostream>

using namespace std;

int main()
{
    int n;
    while(cin >> n)
    {
        cout << 3*n+1 << "\n";
    }
}
```

Exemplo de solução em C (teste.c):

```
#include <stdio.h>

int main()
{
    int n;
    while(scanf("%d", &n)==1)
    {
        printf("%d\n", 3*n+1);
    }
}
```

Exemplo de solução em Java (teste.java):

```
import java.util.Scanner;

public class teste{
    public static void main(String [] args){
        int n;
        Scanner s = new Scanner(System.in);
        try{
            while(true){
                n = s.nextInt();
                System.out.println(3*n+1);
            }
        } catch(Exception e){}
    }
}
```

Exemplo do arquivo de entrada (entrada.in):

8
3
12
2

Obs.: para programas criados em Java, o nome da classe deve ser o mesmo que o nome do arquivo. No exemplo acima, como o arquivo fonte deve ser teste.java, a classe principal deve-se chamar teste.

Sites úteis para auxiliar na preparação para as próximas maratonas

Site: <http://www.urionlinejudge.com.br>

Descrição: URI Online Judge – Site para treinamento de resolução de problemas.

Site: <http://crbonilha.com>

Descrição: Editoriais e algoritmos para maratonas – Blog para estudos de algoritmos. Autor: Cristhian Bonilha.

Site: <http://maratona.algartelem.com.br>

Descrição: Site oficial das maratonas Algar Telecom.

Site: <http://maratona.ime.usp.br>

Descrição: Site oficial das maratonas nacionais.

Problema A

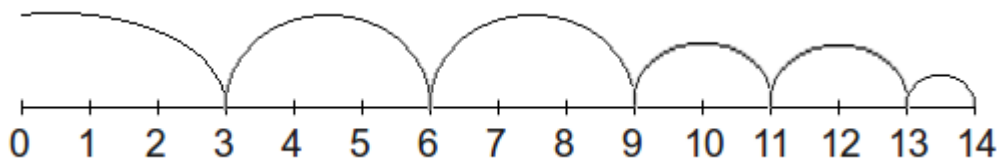
Arremesso de Bolas

Arquivo fonte: arremesso.c, arremesso.cpp ou arremesso.java

Seus amigos inventaram uma nova competição: Arremesso de bolas. O objetivo é simples, basta arremessar uma bola de forma que ela caia dentro de um buraco N metros a sua frente.

Quando a bola é arremessada, digamos que à uma velocidade inteira V , ela permanece no ar por V metros e então quica. Ela repete esse processo V vezes. Após ela quicar V vezes, ela muda sua velocidade para $V-1$, e o processo anterior se repete, até que a velocidade seja igual a 0.

Por exemplo, se a bola for arremessada a uma velocidade igual a 3, ela quicará nos seguintes pontos: 3, 6, 9, 11, 13, 14; conforme pode ser visto na imagem.



Você consegue arremessar a bola a uma velocidade inteira menor ou igual a V . Dada a distância do buraco, diga se é possível que você arremesse a bola e que ela quique exatamente no buraco, acertando-o.

Entrada

Cada caso de teste contém dois inteiros, N e V ($1 \leq N \leq 1000$, $1 \leq V \leq 30$), representando a distância do buraco e a velocidade máxima com a qual você consegue arremessar a bola.

O último caso de teste é indicado quando $N = V = 0$, o qual não deverá ser processado.

Saída

Para cada caso de teste, imprima uma linha contendo a palavra “possivel” (sem aspas), caso seja possível arremessar a bola a uma velocidade menor ou igual a V de forma que ela quique no buraco, ou “impossivel”, caso contrário.

Exemplos de Entrada

```
14 3
13 3
12 3
5 3
30 4
0 0
```

Exemplos de Saída

```
possivel
possivel
impossivel
possivel
possivel
```

Problema B

Detetive Watson

arquivo: detetive.c , detetive.cpp, detetive.java

John Watson, mesmo após anos trabalhando ao lado de Sherlock Holmes, nunca conseguiu entender como ele consegue descobrir quem é o assassino com tanta facilidade.

Em uma certa noite, Sherlock bebeu mais do que devia e acabou contando o segredo a John. “Elementar, meu caro Watson”, disse Sherlock Holmes. “Nunca é o mais suspeito, mas sim o segundo mais suspeito”.

Após descobrir o segredo, John decidiu resolver um crime por conta própria, só para testar se aquilo fazia sentido ou se era apenas conversa de bêbado.

Dada uma lista com N inteiros, representando o quanto cada pessoa é suspeita, ajude John Watson a decidir quem é o assassino, de acordo com o método citado.

Entrada

Cada caso de teste inicia com um inteiro N ($2 \leq N \leq 1000$), representando o número de suspeitos.

Em seguida haverá N inteiros distintos, onde o i -ésimo inteiro, para todo $1 \leq i \leq N$, representa o quão suspeita a i -ésima pessoa é, de acordo com a classificação dada por John Watson. Seja V o valor do i -ésimo inteiro, $1 \leq V \leq 10000$.

O último caso de teste é indicado quando $N = 0$, o qual não deverá ser processado.

Saída

Para cada caso de teste imprima uma linha, contendo um inteiro, representando o índice do assassino, de acordo com o método citado.

Exemplo de Entrada

```
3
3 5 2
5
1 15 3 5 2
0
```

Exemplo de Saída

```
1
4
```

Problema E

Libertadores

Arquivo fonte: libertadores.c, libertadores.cpp ou libertadores.java

A Copa Libertadores da América é a principal competição de futebol entre clubes profissionais da América do Sul, organizada pela Confederação Sul-Americana de Futebol (CONMEBOL). Ela é conhecida por ter um regulamento muito complicado, principalmente nas fases das oitavas, quartas e semi-final.

Nessas fases são jogadas partidas de ida e volta no sistema mata-mata. Ganha quem fizer a maior pontuação no acumulado das duas partidas, sendo 3 pontos para vitória e 1 ponto em caso de empate, ambos por partida. Em caso de igualdade na pontuação, são critérios de desempate:

- 1) saldo de gols (número de gols a favor menos o número de gols contra).
- 2) mais gols marcados na casa do adversário.
- 3) disputa por pênaltis.

Todos os critérios devem ser aplicados considerando o acumulado das duas partidas.

Será que você consegue elaborar um algoritmo que, dados os resultados das partidas de ida e de volta, ele identifica o time vencedor?

Entrada

A primeira linha de entrada indica o número de casos de teste N ($1 \leq N \leq 100$). Cada caso de teste é composto por dois placares: o resultado da partida 1 e o resultado da partida 2. O placar é representado pelo formato $M \times V$, onde M ($1 \leq M \leq 100$) é o número de gols do time mandante da partida e V ($1 \leq V \leq 100$) é o número de gols do time visitante. Como em cada caso de teste existem 2 partidas, considere que o Time 1 é sempre o mandante da primeira e o visitante da segunda e vice-versa para o Time 2.

Saída

Para cada caso de teste, imprima uma linha contendo "Time 1" (sem aspas) caso o Time 1 seja o vencedor do mata-mata, "Time 2" (sem aspas) caso o Time 2 seja o vencedor do mata-mata e "Penaltis" (sem aspas) caso não seja possível identificar o vencedor no tempo convencional.

Exemplo de entrada

```
4
1 x 1
2 x 1
2 x 0
2 x 1
1 x 1
2 x 2
3 x 1
3 x 1
```

Exemplo de saída

```
Time 2
Time 1
Time 1
Penaltis
```

Problema F

Fila de Banco

Arquivo fonte: fila.c, fila.cpp ou fila.java

André, Bruno e Carlos são amigos a um bom tempo, e se tem uma coisa que eles sabem um sobre o outro é o quanto eles são pontuais. André é conhecido por ser sempre o último a chegar em um compromisso entre o três, e Carlos é sempre o primeiro. Bruno sempre chega antes de André, mas nunca antes de Carlos.

Chegou o fim do mês e os três precisam ir ao banco para pagar algumas contas. Contando com eles, há N pessoas na fila para usar o caixa. Sabendo o quanto eles são pontuais entre si, de quantas maneiras possíveis a fila do banco pode estar ordenada?

Lembre-se que as regras acima só se aplicam entre eles, por exemplo, Carlos sempre chega antes que Bruno e André, mas pode chegar depois de outras pessoas na fila. Duas ordenações de fila são consideradas diferentes se ao menos uma pessoa está em um lugar diferente nas duas ordenações.

Entrada

Cada caso de teste inicia com um inteiro N ($3 \leq N \leq 10$), indicando o número de pessoas na fila, incluindo André, Bruno e Carlos.

O último caso de teste é indicado quando $N = 0$.

Saída

Para cada caso de teste imprima uma linha contendo um inteiro, representando o número de maneiras que a fila do banco pode estar ordenada. Como o resultado pode ser um valor muito alto, imprima o resultado com resto de divisão em 1000000009.

Exemplo de entrada

```
3
4
5
10
0
```

Exemplo de saída

```
1
4
20
604800
```

Problema G

Sequência Alienígena

Arquivo fonte: alien.c, alien.cpp ou alien.java

Recentemente foi descoberto um planeta intitulado como Gliese 581g. Ele tem apenas três vezes a massa da Terra, e está à 20 anos-luz de distancia, orbitando uma estrela da constelação de Libra conhecida como Gliese 581, uma anã vermelha. Astrônomos da Universidade da Califórnia e da Carnegie Institution de Washington afirmam que o planeta é o primeiro a apresentar potencial real para conter vida. Eles disseram que essa descoberta só foi possível através da análise de antigas escrituras egípcias, remanescentes da biblioteca de Alexandria. Foram encontradas anotações com uma sequência estranha, que supostamente indicaria as posições dos planetas com vida no universo. Cada elemento é traduzido em coordenadas tridimensionais utilizando um algoritmo extremamente complexo descrito nas anotações, no entanto o algoritmo para cálculo da sequência se perdeu por causa das páginas queimadas no incêndio da biblioteca.

O livro contém apenas os oito primeiros valores dessa sequência, detectada nas páginas legíveis das escrituras, e para conseguir encontrar as outras coordenadas a Universidade da Califórnia lançou um desafio mundial para estudantes de matemática, engenharia e computação. Dados os primeiros elementos da sequência, deve-se escrever um algoritmo capaz de calcular qualquer elemento da série. Os elementos conhecidos são {B, BA, CB, BAA, BCB, CBA, DAB, BAAA}.

Acredita-se que seja possível encontrar qualquer elemento da série, cujos elementos parecem estar escritos em um base numérica alienígena.

Entrada

Cada linha de entrada contém um inteiro N ($1 \leq N \leq 100000$), que descreve a posição na sequência alienígena do elemento a ser calculado. A entrada é finalizada quando $N = 0$.

Saída

Para cada valor lido, deve ser impresso na tela o elemento na n -ésima posição da sequência alienígena, sempre com uma quebra de linha.

Exemplo de entrada

```
1
2
3
4
5
6
7
8
0
```

Exemplo de saída

```
B
BA
CB
BAA
BCB
CBA
DAB
BAAA
```


Problema H

Empresa de Telecom

arquivo: telecom.c , telecom.cpp, telecom.java

Cesário é um analista da Algar Telecom, e está trabalhando em um projeto de análise da rede de telefonia móvel. Ele terá que desenvolver um sistema que analise o alcance de cada uma das antenas dessa rede, e que defina os custos operacionais para o envio de dados de de dispositivo para outro, baseando-se na distancia entre as antenas. O objetivo minimizar esses custos, encontrando a melhor rota disponível. Os cálculos também visam descobrir se é possível estabelecer um caminho entre dois dispositivos, de forma a detectar graves problemas na rede.

Mesmo com todos os dados disponíveis para processamento, Cesário tem enfrentado problemas na implementação devido a alta complexidade desse algoritmo, por isso você foi contratado para ajudá-lo. O seu objetivo é analisar todas as antenas da rede da Algar Telecom, observando as suas coordenadas e raios de alcance; verificar quais as antenas possíveis de serem acessadas (dentro do raio de alcance); e calcular o menor caminho entre duas antenas determinadas.

Entrada

A entrada é composta de vários casos de testes. Sendo que, a primeira linha contém um inteiro não negativo, N ($2 \leq N \leq 100$), que indica o número de antenas disponíveis para interconexão na rede. Seguem-se N linhas, cada uma contendo três números inteiros X ($0 \leq X \leq 1000$), Y ($0 \leq Y \leq 1000$) e R ($1 \leq R \leq 1000$), que descrevem a posição da antena, coordenadas X e Y , e o seu raio de alcance R (separados por espaço em branco). A linha seguinte contém outro inteiro não negativo, C ($1 \leq C \leq 100$), que descreve a quantidade de cálculos à serem realizados nessa rede. As C linhas seguintes contém 2 inteiros cada, $A1$ ($1 \leq A1 \leq N$) e $A2$ ($1 \leq A2 \leq N$), que descrevem o índice das antenas a serem utilizadas e também separadas por espaço em branco.

O fim das entradas é sinalizado por um número 0.

Saída

Para cada caso de teste, deve-se imprimir C linhas, sendo que cada uma representa a distância do menor caminho entre as duas antenas. Os valores devem ser **INTEIROS**, ou seja, a parte real deve ser truncada (não arredondada), e sempre com uma quebra de linha. Caso não seja identificada uma rota entre as antenas, deve ser impresso o valor -1.

Exemplo de Entrada

```
2
0 3 3
2 1 1
2
1 2
2 1
3
1 1 2
0 0 1
1 2 2
4
1 3
2 3
3 1
3 3
0
```

Exemplo de Saída

```
2
-1
1
-1
1
0
```

Problema I

Planejando a Energia

arquivo: energia.c , energia.cpp, energia.java

Você está participando de um comitê que irá ajudar a planejar o crescimento da energia elétrica no Brasil, garantindo assim que as usinas consigam fornecer a energia necessária no futuro.

Para isso você tem as seguintes informações:

- a) durante o ano de 2010 o consumo médio do brasileiro foi de 104.326 GWh.
- b) em 2013 o consumo foi de 127.755 GWh.

Você deve determinar a taxa de crescimento anual para diferentes situações e previsões futuras, considerando o fato deste crescimento ser linear. Nesse caso, a taxa foi de 7.809,66 GWh/ano.

Entrada

A primeira linha da entrada contém um número inteiro N ($1 \leq N \leq 1000$) representando o total de casos de testes.

As N linhas seguintes são compostas de 4 números inteiros A , B ($B > 0$), C , D ($D > 0$) separados por espaço. O número A representa o ano, o número B representa o consumo do ano A . O número C representa um outro ano e o número D representa o consumo de C .

Saída

Para cada caso de teste deverá ser impresso a taxa de crescimento anual com apenas duas casas decimais, separadas por vírgula e truncadas, ou seja, sem arredondamentos.

Exemplo de Entrada

```
1
2010 104326 2013 127755
```

Exemplo de Saída

```
7809,66
```

Problema J

Construindo Casas

arquivo: casas.c , casas.cpp, casas.java

Sr Pi é um construtor famoso na cidade de Programolândia. Ele precisa de sua ajuda para encontrar, os melhores terrenos da cidade, para os vários projetos que ele possui, para a construção de casas.

Por exemplo, ele tem um projeto para construir uma casa de 8 metros por 10 metros mas, a legislação do município só permite construir, neste bairro, em no máximo 20% do terreno. Todos os terrenos nesta cidade são perfeitamente quadrados. Sabendo que o valor dos lados da casa são apenas uma referência para a área total a ser construída. (Ex: Uma casa de 1 metro por 10 metros com construção de 100% permitida, o sr PI precisaria de um terço de 10m². O valor do lado do terreno neste caso, teria 3 metros, pois é truncado) Ajude o sr PI a determinar o tamanho mínimo do terreno.

Entrada

A entrada é composta de vários casos de testes. Cada caso de teste é composto de três números inteiros A e B (> 0 e ≤ 1000) e C separados por um espaço. Estes números representam as medidas da casa (A e B) e o percentual máximo liberado para construir nesse bairro (C Inteiro). Um único valor igual a 0 indica o fim das entradas.

Saída

Você deverá informar um número inteiro, o qual representa a medida do lado do terreno. Este valor deverá ser truncado caso necessário.

Exemplo de Entrada

```
8 10 20
1 10 100
10 3 100
0
```

Exemplo de Saída

```
20
3
5
```

Problema K

Lendo Livros

arquivo: livros.c , livros.cpp, livros.java

Você começou a competir com seu amigo para ver quem consegue ler mais livros em menos tempo. Seu amigo lia muito mais que você, até o dia que você percebeu que ele lia somente livros muito finos.

Então você resolveu contar as páginas dos livros, aumentando também a quantidade de páginas lidas por dia. Agora você lê 5 páginas por dia e termina 16 dias antes do que se estivesse lendo 3 páginas por dia. Neste cenário, quantas páginas tem o livro?.

Entrada

A entrada é composta de vários casos de testes. Cada caso de teste é composto de três números Q ($0 < Q < 20$), D ($0 < D < 20$) e P ($0 < P < 20$) separados por um espaço. Sendo que Q é a quantidade de páginas lidas por dia. D é o número de dias que você adiantaria a leitura caso estivesse lendo a quantidade de páginas informada pelo número P . Um único valor zero indica o fim da entrada.

Saída

Para cada caso de teste deverá ser impresso a quantidade de páginas do livro. (utilize o plural corretamente e não use acentos). Este número deverá ser um inteiro, o qual representa a quantidade de página. Este valor deverá ser truncado caso necessário.

Exemplo de Entrada

```
3 16 5
0
```

Exemplo de Saída

```
120 paginas
```

Problema L

Quadro Premiado

arquivo: quadro.c , quadro.cpp, quadro.java

Você está em um programa de televisão, e tem uma ótima chance de ganhar muito dinheiro. Trata-se de um jogo com algumas regras peculiares, e o montante de dinheiro resultante dependerá apenas da sua esperteza, podendo-se até sair perdendo caso se jogue mal.

O jogo funciona da seguinte maneira: há um quadro, com N linhas e M colunas, e em cada posição deste quadro há um inteiro positivo, representando uma quantia em dinheiro. Em cada uma dessas posições você tem a opção de colocar um dos seguintes sinais:

- '+' - Significa que o valor daquela posição deve ser somado à seu prêmio.
- '-' - Significa que o valor daquela posição deve ser subtraído do seu prêmio.
- '.' - Significa que tal posição deve ser ignorada.

A vida seria muito simples se você pudesse colocar '+' em todas as posições, portanto há duas regras adicionais ao jogo: para cada linha do quadro, você deve preencher as posições com um dos padrões de sinais montados pelos organizadores do jogo; e para cada coluna do quadro, não é permitido que duas posições adjacentes verticalmente tenham o mesmo sinal (se aplica aos sinais '+' e '-'). É possível usar o mesmo padrão mais de uma vez, desde que não desrespeitando a segunda regra acima.

Veja um exemplo na imagem abaixo, onde os padrões são: "++", "--", ".+" e "+.".

3 ₊	4 ₊	3 ₊	4 ₊	3 ₊	4 _.
1 ₋	2 ₋	1 _.	2 ₊	1 _.	2 ₊
Total: 4		Inválido		Total: 5	

Considere que há sempre ao menos uma maneira de se completar o quadro.

Como o jogo é novo, eles deixaram que você usasse seu computador para te ajudar na decisão, sem saber que você era um programador. Escreva um algoritmo que lhe diga qual a soma máxima que é possível alcançar no jogo.

Entrada

Haverá diversos casos de teste. Cada caso de teste inicia com dois inteiros, N e M ($1 \leq N, M \leq 100$), indicando o número de linhas e de colunas do quadro, respectivamente.

A seguir haverá N linhas, contendo M inteiros cada, representando os valores do quadro. Seja v o valor de qualquer posição do quadro, $1 \leq v \leq 100$.

A seguir haverá um inteiro K ($1 \leq K \leq 100$), indicando o número de padrões. Em seguida haverá K linhas, cada uma com M caracteres, representando cada um dos padrões, conforme a simbologia descrita no enunciado.

O último caso de teste é indicado quando $N = M = 0$, o qual não deverá ser processado.

Saída

Para cada caso de teste imprima uma linha, contendo um inteiro, representando a soma máxima que é possível alcançar se os padrões forem escolhidos de forma ótima.

Exemplo de Entrada

```
2 2
3 4
1 2
4
++
--
```

+ .
+ .
3 3
1 3 2
4 2 3
3 5 1
2
+ . +
- + -
0 0

Exemplo de Saída

5
8