

UNIVERSIDADE FEDERAL DE ITAJUBÁ  
ECOX21 - MARATONA DE PROGRAMAÇÃO I  
*Profs. Edmilson Marmo e Luiz Olmes*



## Warmup 1

28/08/2024

### Regras

1. Há 5 problemas que devem ser resolvidos no tempo estipulado.
2. Os dados de entrada devem ser lidos a partir da entrada padrão.
3. Os dados de saída devem ser escritos na saída padrão.
4. Quando uma linha contém vários valores, eles estarão separados por um único espaço. Não há espaços extras na entrada/saída. Não há linhas em branco na entrada.
5. A entrada e a saída usam o alfabeto latino. Não haverá letras com til, acentos, tremas ou outros sinais diacríticos.
6. Todas as linhas da entrada e da saída, incluindo a última contêm o caractere de fim de linha.
7. O código fonte de cada solução deve ser enviado pelo Boca: `boca.unifei.edu.br`

## Ambiente de Testes

A correção das soluções enviadas é realizada no sistema operacional Red Hat Enterprise Linux, versão 8.6 (Ootpa), usando os seguintes compiladores/interpretadores:

C: gcc versão 8.5.0 20210514 (Red Hat 8.5.0-10)  
C++: g++ versão 8.5.0 20210514 (Red Hat 8.5.0-10)  
Java: openjdk versão 1.8.0\_342  
Python: python3 versão 3.6.8

## Limites

Memória (C, C++, Python): 1GB  
Memória (Java): 1GB + 100MB stack  
Tamanho máximo do código fonte: 100KB  
Tamanho máximo do arquivo executável: 1MB

Códigos que extrapolem os limites permitidos receberão *Runtime Error* como resposta.

## Comandos de Compilação

C: gcc -g -O2 -std=gnu11 -static -lm  
C++: g++ -g -O2 -std=gnu++17 -static -lm  
Java: javac

## Códigos C/C++

- O programa deve retornar zero, executando, como último comando, `return 0` ou `exit(0)`.
- Para entradas grandes, objetos `iostream` podem ser lentos, devido a questões de sincronização de buffer com a biblioteca `stdio`. Recomenda-se desabilitar este mecanismo de sincronização em programas que empregam `std::cin` e `std::cout` através dos seguintes comandos:

```
std::ios::sync_with_stdio(0);  
std::cin.tie(0);
```

Note que, neste caso, deve-se evitar usar `printf` e `scanf` no mesmo programa, pois a separação dos buffers pode levar a resultados inesperados.

## Códigos Java

- O programa não deve estar encapsulado em um package.
- Para cada problema, o arquivo `.java` e a `public class` devem ter o mesmo nome `basename` mostrado no Boca.
- Comando de execução: `java -Xms1024m -Xmx1024m -Xss100m`

## Códigos Python

- Apenas Python 3 é suportado. Python 3 não é compatível com Python 2.
- **Atenção:** não é garantido que soluções escritas em Python executarão dentro do tempo limite especificado para cada problema.
- Comando de execução: `python3`

**Problema  $\mathcal{A}$**   
**SÓ PARA ENTENDER**  
*Timelimit: 1s*

Este primeiro exercício é muito simples. O seu objetivo é permitir que o aluno possa fazer um código rápido, sem muitos problemas e submeter a sua solução com grande probabilidade de acerto.

Seu objetivo é ler três números inteiros e apresentá-los em ordem crescente. Simples, não? Então, mãos a obra!

**Entrada**

A entrada contém um único caso de teste. A primeira linha de um caso de teste contém um inteiro  $N$  que indica o número de sequências que serão ordenadas ( $1 \leq N \leq 1000$ ). As  $N$  linhas seguintes possuem, cada uma, 3 números inteiros  $A, B, C$  ( $-15000 \leq A, B, C \leq 15000$ ).

**Saída**

Para cada sequência de números  $A, B, C$  da entrada seu programa deve produzir uma única linha de saída, indicando a ordem correta dos números. Cada número deve estar separado por um único espaço em branco.

**Exemplos**

<i>Entrada</i>	<i>Saída</i>
3	1 2 3
1 2 3	1 2 3
3 2 1	1 2 3
3 1 2	

## Problema B

### ESQUERDA, VOLVER

Timelimit: 1s

Este ano o sargento está tendo mais trabalho do que de costume para treinar os recrutas. Um deles é muito atrapalhado, e de vez em quando faz tudo errado. Por exemplo, ao invés de virar à direita quando comandado, vira à esquerda, causando grande confusão no batalhão. O sargento tem fama de durão e não vai deixar o recruta em paz enquanto este não aprender a executar corretamente os comandos. No sábado à tarde, enquanto todos os outros recrutas estão de folga, ele obrigou o recruta a fazer um treinamento extra. Com o recruta marchando parado no mesmo lugar, o sargento emitiu uma série de comandos “Esquerda, volver!” e “Direita, volver!”. A cada comando, o recruta deve girar sobre o mesmo ponto e dar um quarto de volta na direção correspondente ao comando. Por exemplo, se o recruta está inicialmente com o rosto voltado para a direção norte, após um comando de “Esquerda volver!” ele deve ficar com o rosto voltado para a direção oeste. Se o recruta está inicialmente com o rosto voltado para o leste, após um comando “Direita, volver!” ele deve ter o rosto voltado para o sul. No entanto, durante o treinamento, em que o recruta tinha inicialmente o rosto voltado para o norte, o sargento emitiu uma série tão extensa de comandos, e tão rapidamente, que até ele ficou confuso, e não sabe mais para qual direção o recruta deve ter seu rosto voltado após executar todos os comandos. Você pode ajudar o sargento?

#### Entrada

A entrada contém vários casos de teste. A primeira linha de um caso de teste contém um inteiro  $N$  que indica o número de comandos emitidos pelo sargento ( $1 \leq N \leq 1000$ ). A segunda linha contém  $N$  caracteres, descrevendo a série de comandos emitidos pelo sargento. Cada comando é representado por uma letra: ‘E’ (para “Esquerda, volver!”) e ‘D’ (para “Direita, volver!”). O final da entrada é indicado por  $N = 0$ .

#### Saída

Para cada caso de teste da entrada seu programa deve produzir uma única linha da saída, indicando a direção para a qual o recruta deve ter sua face voltada após executar a série de comandos, considerando que no início o recruta tem a face voltada para o norte. A linha deve conter uma letra entre ‘N’, ‘L’, ‘S’ e ‘O’, representando respectivamente as direções norte, leste, sul e oeste.

#### Exemplos

Entrada	Saída
3 DDE 2 EE 0	L S

**Problema C**  
**APRENDENDO A CONTAR**  
*Timelimit: 1s*

Tia Dorinha é uma excelente professora do ensino fundamental. Ela está ensinando seus alunos a contar. Ela sempre prepara jogos para dinamizar a suas aulas.

Desta vez, ela preparou o seguinte jogo: ela apresenta 3 números ( $I, F, X$ ) para os alunos e eles devem escrever no caderno a sequência que começa com o primeiro número ( $I$ ) e vai até o último número ( $F$ ), mas os alunos não devem escrever os números que são múltiplos do terceiro número ( $X$ ) na sequência.

Desta forma, ela solicitou a sua ajuda para já deixar pronto o gabarito que irá entregar aos alunos no final da aula para que eles confirmem se acertaram ou erraram cada rodada.

### Entrada

A entrada contém vários casos de teste. A única linha de um caso de teste contém os três inteiros  $I, F$  e  $X$  ( $0 \leq I < F \leq 1000$ ;  $2 \leq X \leq 10$ ). O final da entrada é indicado por  $I = F = X = 0$ .

### Saída

Para cada caso de teste da entrada seu programa deve produzir uma única linha de saída contendo a sequência correta.

### Exemplos

<i>Entrada</i>	<i>Saída</i>
3 5 2	3 5
1 11 2	1 3 5 7 9 11
90 100 6	91 92 93 94 95 97 98 99 100
0 0 0	

## Problema $\mathcal{D}$

### GUARDA COSTEIRA

*Timelimit: 1s*

“Pega ladrão! Pega ladrão!”

Roubaram a bolsa de uma inocente senhora que caminhava na praia da Nlogônia e o ladrão fugiu em direção ao mar. Seu plano parece óbvio: ele pretende pegar um barco e escapar!

O fugitivo, que a essa altura já está a bordo de sua embarcação de fuga, pretende seguir perpendicularmente à costa em direção ao limite de águas internacionais, que fica a 12 milhas náuticas de distância, onde estará são e salvo das autoridades locais. Seu barco consegue percorrer essa distância a uma velocidade constante de  $V_F$  nós.

A Guarda Costeira pretende interceptá-lo e sua embarcação tem uma velocidade constante de  $V_G$  nós. Supondo que ambas as embarcações partam da costa exatamente no mesmo instante, com uma distância de  $D$  milhas náuticas entre elas, será possível a Guarda Costeira alcançar o ladrão antes do limite de águas internacionais?

Assuma que a costa da Nlogônia é perfeitamente retilínea e o mar bastante calmo, de forma a permitir uma trajetória tão retilínea quanto a costa.

### Entrada

A entrada contém vários casos de teste. Cada caso de teste consiste de apenas uma linha, contendo três valores,  $D$ ,  $V_F$  e  $V_G$ , indicando respectivamente a distância inicial entre o fugitivo e a Guarda Costeira, a velocidade da embarcação do fugitivo e a velocidade da embarcação da Guarda Costeira ( $1 \leq D, V_F, V_G \leq 100$ ). Valores  $D = V_F = V_G = 0$  indicam o fim da entrada.

### Saída

Seu programa deve produzir uma única linha, contendo ‘S’ se for possível que a Guarda Costeira alcance o fugitivo antes que ele ultrapasse o limite de águas internacionais, ou ‘N’ caso contrário

### Exemplos

<i>Entrada</i>	<i>Saída</i>
5 1 12	S
12 10 7	N
12 9 10	N
0 0 0	

**Problema E**  
**TRIÂNGULO**  
*Timelimit: 1s*

Ana e suas amigas estão fazendo um trabalho de geometria para o colégio, em que precisam formar vários triângulos, numa cartolina, com algumas varetas de comprimentos diferentes. Logo elas perceberam que não dá para formar triângulos com três varetas de comprimentos quaisquer: se uma das varetas for muito grande em relação às outras duas, não dá para formar o triângulo.

Neste problema, você precisa ajudar Ana e suas amigas a determinar se, dados os comprimentos de quatro varetas, é ou não é possível selecionar três varetas, dentre as quatro, e formar um triângulo.

**Entrada**

A entrada contém vários casos de teste. Cada caso de teste contém quatro inteiros  $A, B, C$  e  $D$  dispostos em ordem aleatória, indicando o tamanho das quatro varetas, em centímetros ( $1 \leq A, B, C, D \leq 100$ ). A entrada termina com final de arquivo (EOF).

**Saída**

Para cada caso de teste da entrada, seu programa deverá imprimir o caracter S (maiúsculo), caso seja possível formar o triângulo; ou N (maiúsculo), caso não seja possível.

**Exemplos**

<i>Entrada</i>	<i>Saída</i>
6 9 22 5 14 40 12 60	S N