

# 15ª Maratona de Programação

11 de abril de 2015

## Caderno de Aquecimento

Regras: ACM *International Collegiate Programming Contest*

Brasil – **Desafio ALGAR TELECOM**

(Este caderno contém 2 problemas; as páginas estão numeradas de 1 a 4)

## 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.

# Problema A

## Ordenação

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

Caio está estudando a ordenação por Inserção. Este é um método simples de ordenação, baseado em inserir, passo a passo, em um subvetor já ordenado, um novo elemento. Para isso, os elementos do subvetor que sejam maiores do que está sendo inserido são deslocados para a direita. O algoritmo estudado por Caio usa como sentinela, na posição 0 do vetor, o próprio elemento que está sendo inserido. Desta forma, ao se inserir o elemento na posição  $i$  do vetor, até  $i$  comparações são necessárias para se ajustar o vetor e encontrar a posição de inserção.

Neste problema, você deve ajudar o Caio a compreender a ordenação por Inserção, contando o total de comparações necessárias para a ordenação de uma dada string de caracteres (deve incluir comparações com a sentinela).

### Entrada

Vários casos de teste são propostos. A primeira linha da entrada é um inteiro  $t$  ( $1 \leq t \leq 1000$ ) que indica quantos serão os casos de teste. A seguir são descritos  $t$  testes. Cada teste consiste de uma linha contendo uma string de até 1000 caracteres.

### Saída

Para cada caso de teste imprima o total de comparações para ordenar a string dada, usando o método descrito.

### Exemplo de entrada

```
3
TESTEDEINSERCAO
GFEDCBA
g
```

### Exemplo de saída

```
75
27
0
```

# Problema B

## Números Triangulares

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

O agente secreto Little Barby recebe mensagens codificadas com as tarefas de suas próximas missões. Tudo isto para evitar que o inimigo possa identificar qual é a sua tarefa secreta. Little recebe um documento contendo uma senha, seguidos de outros números em sequência contendo as descrições das missões. Sua tarefa é ajudar Little a descobrir quais números da sequência ele deverá ler para determinar sua missão. A senha é um número triangular, ou seja, ele é produto de três números naturais consecutivos. Mas nem todos os números são válidos, essa é uma estratégia para enganar seus inimigos. Veja um exemplo do documento:

### 120 (senha)

- 1 - Missão XX
- 2 - Missão YYY
- 3 - Missão zzzz
- 4 - Missão Maratona
- 5 - Missão Maratona Aquecimento
- 6 - Missão Secreta 2
- 7 - Missão UUUU

### Entrada

A entrada contém vários casos de teste. Cada linha representa um caso de teste e o número informado representa a senha  $s$  ( $s \geq 0$ ).

### Saída

A saída terá os valores dos números naturais consecutivos ou uma mensagem informando que a mensagem é falsa. ("mensagem falsa")

### Exemplos de Entrada

```
6
7
120
```

### Exemplos de saída para a Entrada

```
1 2 3
mensagem falsa
4 5 6
```