

13ª Maratona de Programação

29 de março de 2014

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 5)

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.algartelecom.com.br>

Descrição: Site oficial das maratonas Algar Telecom.

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

Descrição: Site oficial das maratonas nacionais.

Problema A

Limite de Velocidade

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

Cesarino e Juvenil estavam viajando felizes da vida pelos Estados Unidos até que o odômetro de seu possante quebrou. Como eles tinham combinado de dividir as despesas, eles ficaram sem saber quantas milhas haviam sido percorridas. Para tentar resolver o problema, Juvenil resolveu registrar, usando um cronômetro de um relógio, a velocidade e o tempo gasto naquela velocidade. Como a forma de registrar ficou meio estranha, você precisa fazer um programa para ajudá-los. Por exemplo, se os registros fossem:

Velocidade em milhas por hora	Tempo Total gasto em horas
20	2
30	6
10	7

Significa que eles dirigiram 2 horas a 20 milhas por hora. Então $6-2=4$ horas a 30 milhas por hora, então $7-6=1$ hora a 10 milhas por hora. A distância percorrida é então $(2)(20) + (4)(30) + (1)(10) = 40 + 120 + 10 = 170$ milhas. Note que o tempo total percorrido é considerado sempre à partir do início da viagem, não dá linha anterior mostrada nos registros.

Entrada

A entrada consiste de um ou mais conjuntos de registros. Cada conjunto inicia com uma linha contendo um inteiro n , $1 \leq n \leq 10$, seguido por n pares de valores, um par por linha. O primeiro valor, s , é a velocidade em milhas por hora e o segundo valor, t , é o tempo total gasto. Ambos os valores, s e t são inteiros, $1 \leq s \leq 90$ e $1 \leq t \leq 12$. Os valores de t estão sempre em ordem crescente. Um valor de -1 para n finaliza a entrada.

Saída

Para cada conjunto de entrada, escreva a distância percorrida, seguida por um espaço em branco, seguida pela palavra "milhas" e uma quebra de linha.

Exemplo de Entrada

```
3
20 2
30 6
10 7
2
60 1
30 5
-1
```

Saída para o exemplo de entrada

```
170 milhas
180 milhas
```

Problema B

RGB

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

Você está testando um novo formato de arquivo para armazenar imagens. Para isso, você estudou vários outros formatos, como o bmp e jpeg. Em linhas gerais, para ler um arquivo contendo uma imagem, é necessário ler o arquivo, interpretar o conteúdo, ler os bytes e plotar a imagem na tela. Para facilitar os estudos, você considerou apenas imagens de 16 ou 256

cores (representadas por 4 ou 8 bits) e o modelo RGB (sistema de cores aditivas formado por Vermelho (Red), Verde (Green) e Azul (Blue)).

Você gravou um arquivo contendo uma sequência de bytes representando as cores de cada pixel. Para que você consiga plotar o pixel, você precisa saber os componentes RGB.

Sua missão é descobrir cada componente do RGB dos bytes informados.

Entrada

A entrada é composta de vários casos de testes. A primeira linha é composta de um inteiro N ($1 \leq N \leq 1000$) representando o número de casos de testes. Cada instância dos testes é composta de 3 linhas. A primeira linha representa o número de bits usado na composição da cor do pixel (4 ou 8). A segunda linha informa o número de bytes da sequência (sempre será múltiplo de 3 e maior que 1). A terceira linha representa a sequência de bytes separados por um espaço.

Saída

A saída para cada caso de teste deverá ser os componentes do RGB. Cada linha da saída terá os componentes R seguido de um espaço em branco, G seguido de um espaço em branco e B seguido de um espaço em branco. Uma quebra de linha é adicionada a cada instância de teste.

Exemplo de entrada

```
2
4
6
15 255 15 170 192 192
8
6
230 200 130 144 130 120
```

Exemplo de saída

```
0 15 15 15 0 15 10 10 12 0 12 0
230 200 130 144 130 120
```