



**UnB**

Departamento de  
Ciência da Computação

4 de novembro de 2014  
2012/2

*Prof Guilherme N. Ramos*

# Micro Maratona de Programação SemEx 2014

Instruções:

1. Este caderno contém 5 problemas.
2. Em todos os problemas, a entrada de seu programa deve ser lida da *entrada padrão*. A saída deve ser escrita na *saída padrão*.
3. A avaliação é individual e tentativas de plágio não serão admitidas.
4. A duração do simulado é 90 minutos.

# PROBLEMA A

## Mim quer somar...

*Arquivo:* soma.[c|cpp|java]

Mim gosta somar dinheiros. Portanto, leia 2 valores inteiros e os armazene nas variáveis **salario** e **bonus**. Efetue a soma de **salario** e **bonus** atribuindo o seu resultado a variável **renda**. Então imprima o valor de **renda** conforme exemplo abaixo. Não apresente mensagem alguma além do especificado, e não esqueça de imprimir o fim de linha (o famoso ‘\n’) após o resultado, ou receberá o aviso de “*Presentation Error*”. Lembre-se também de retornar 0!

### Entrada

A entrada contém 2 valores inteiros.

### Saída

Imprima a variável **renda** conforme exemplo abaixo, com um espaço em branco antes e depois da igualdade.

### Exemplo

Entrada	Saída
2 5	renda = 7

# PROBLEMA B

$$3n + 1$$

*Arquivo:* 3nPlus1.[c|cpp|java]

Considere o seguinte algoritmo para gerar uma sequência de números. Comece com um valor inteiro  $n$ . Se  $n$  for par, divida-o por 2. Se  $n$  for ímpar, multiplique-o por 3 e some 1. Repita este processo para o novo valor de  $n$ , terminando quando  $n = 1$ . Por exemplo, a seguinte sequência é gerada para  $n = 22$ :

22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1

Conjectura-se que este algoritmo terminará ( $n = 1$ ) para qualquer valor inteiro  $n$ . Isto ainda não foi provado matematicamente, mas sabe-se que é verdade para  $n \in (0, 10^6)$ . Para um valor  $n$  de entrada, o *comprimento do ciclo* de  $n$  é a quantidade de números gerados até 1 (inclusive). Para o exemplo, o comprimento do ciclo de 22 é 16.

Dados dois números  $i$  e  $j$  quaisquer, você deve construir um algoritmo que determine o máximo ciclo para qualquer número neste intervalo. Por exemplo, sabe-se que o maior ciclo no intervalo  $[1, 10]$  é 20.

## Entrada

A entrada é formada por um par de inteiros  $i$  e  $j$  por linha, tais que  $0 < i, j < 10^6$ . Ela termina com EOF.

## Saída

Para o par de inteiros  $i$  e  $j$  fornecido, você deve imprimir  $i$  e  $j$ , e o máximo ciclo computado para todos os inteiros  $k$  ( $i \leq k \leq j$ ). Os inteiros  $i$  e  $j$  devem aparecer na mesma ordem em que foram lidos. Estes três números devem ser separados por espaços, todos na mesma linha, com uma linha de saída para cada linha de entrada.

## Exemplo

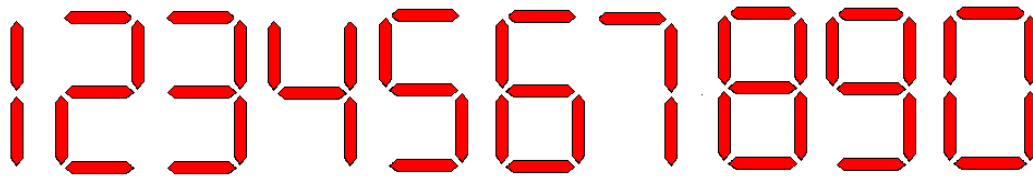
Entrada	Saída
1 10	1 10 20
100 200	100 200 125
201 210	201 210 89
900 1000	900 1000 174

# PROBLEMA C

## LED Fácil

*Arquivo:* `easyled.[c|cpp|java]`

Um *diodo emissor de luz* (LED) pode ser usado como uma lâmpada extremamente eficiente em um letreiro, e D’Barros quer montar um painel mostrando a quantidade de clientes que ele atendeu em sua lojinha antes de se aposentar. Ele não possui muitos LEDs, e não tem certeza se conseguirá montar o número desejado. Considerando a configuração de LEDs dos números abaixo (cada traço é um LED), faça um algoritmo que ajude D’Barros a descobrir o número de LEDs necessário para exibir tal valor.



### Entrada

A entrada contém um inteiro  $N$ , ( $1 \leq N \leq 2 \cdot 10^6$ ), correspondendo a quantidade de clientes atendidos por D’Barros.

### Saída

Mostre uma linha contendo o número de LEDs que D’Barros precisará para exibir a quantidade desejada, seguido pela palavra “LEDs” (e quebra de linha!).

### Exemplo

Entrada	Saída
12	7 LEDs

# PROBLEMA D

## Quantos Quadrados?

*Arquivo:* `quadradofacil.[c|cpp|java]`

Quem quer quantificar quantos quadrados? Quasimodo quer! Seu passatempo favorito é quantificar quantos quadrados existem em uma grade quadrada. Por exemplo, se a grade tem 2 quadrados de lado, há um total de 5 quadrados:



### Entrada

Cada linha da entrada irá conter um inteiro  $0 \leq Q \leq 256$  representando a quantidade de quadrados por lado da grade. A entrada termina quando  $Q = 0$ .

### Saída

Para cada linha da entrada, imprima em uma linha o valor da quantidade de quadrados diferentes para grade definida (seguido de quebra de linha, sem espaços).

### Exemplo

Entrada	Saída
2	5
5	55
0	

# PROBLEMA E

## Minas Terrestres

*Arquivo:* mina.[c|cpp|java]

O robô MT-321 consegue indicar exatamente o local onde está enterrada uma mina terrestre, mas seu companheiro de trabalho que desarmava estas terríveis armas, o robô MT-123, coçou a engrenagem na hora errada e explodiu no cumprimento do dever.

Sobrou para o programador do MT-123 a tarefa ingrata de desarmar as minas. Felizmente, isto é fácil com a nova tecnologia, basta fazer um círculo com o cabo ACME-NT1RA ao redor da mina. Infelizmente, a liberação do pulso eletromagnético que desarma a mina tem de ser feita manualmente no local. Assim, se conseguir preparar o círculo corretamente (envolvendo completamente a mina), ele volta para casa como herói, se não, bem, não volta...

Dadas as características de dois círculos, um definindo a posição do centro de ACME-NT1RA e o outro representando a área da mina terrestre, faça um programa que determine se o programador consegue desarmar o artefato.

### Entrada

A entrada é composta por diversas instâncias e termina com o caractere final de arquivo (EOF). Cada instância consiste em uma linha com seis inteiros,  $R_1$  ( $1 \leq R_1$ ),  $X_1$  ( $|X_1|$ ),  $Y_1$  ( $|Y_1|$ ),  $R_2$  ( $R_2 \leq 1000$ ),  $X_2$  ( $|X_2|$ ),  $Y_2$  ( $|Y_2|$ ). O círculo definido pelo programador possui raio  $R_1$  e centro  $(X_1; Y_1)$ . O círculo representando a área da mina terrestre possui raio  $R_2$  e centro  $(X_2; Y_2)$ .

### Saída

Para cada instância imprima uma única linha contendo *MARTIR*, se o programador morre, ou *HEROI* se ele consegue desarmar a mina..

### Exemplo

Entrada	Saída
6 -8 2 3 0 0	MARTIR
7 3 4 2 4 5	HEROI
3 0 0 4 0 0	MARTIR
5 4 7 1 8 7	HEROI