

## ALGORITMOS I

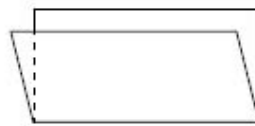
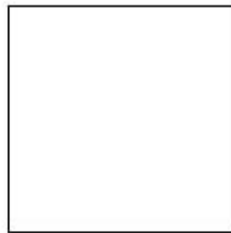
### 3ª LISTA DE EXERCÍCIOS

#### 1 (OBI2002) DOBRADURA

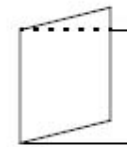
Zeinho tem aulas de Iniciação Artística em sua escola, e recentemente aprendeu a fazer dobraduras em papel. Ele ficou fascinado com as inúmeras possibilidades de se dobrar uma simples folha de papel. Como Zeinho gosta muito de matemática, resolveu inventar um quebra-cabeça envolvendo dobraduras.

##### Tarefa

Zeinho definiu uma operação de dobradura D que consiste em dobrar duas vezes uma folha de papel quadrada de forma a conseguir um quadrado com  $1/4$  do tamanho original, conforme ilustrado na figura.

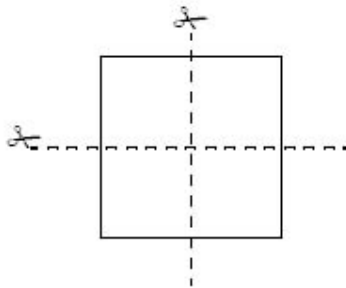


Primeira dobra



Segunda dobra

Depois de repetir N vezes esta operação de dobradura D sobre o papel, Zeinho cortou o quadrado resultante com um corte vertical e um corte horizontal, conforme a figura abaixo.



Zeinho lançou então um desafio aos seus colegas: quem adivinha quantos pedaços de papel foram produzidos?

##### Entrada

A entrada é composta de vários conjuntos de teste. Cada conjunto de teste é composto de uma única linha, contendo um número inteiro N que indica o número de vezes que a operação de dobradura D foi aplicada. O final da entrada é indicado por N = -1.

### Exemplo de Entrada

1  
0  
-1

### Saída

Para cada conjunto de teste da entrada seu programa deve produzir três linhas na saída. A primeira linha deve conter um identificador do conjunto de teste, no formato "Teste n", onde n é numerado a partir de 1. A segunda linha deve conter o número de pedaços de papel obtidos depois de cortar a dobradura, calculado pelo seu programa. A terceira linha deve ser deixada em branco.

A grafia mostrada no Exemplo de Saída, abaixo, deve ser seguida rigorosamente.

### Exemplo de Saída

Teste 1  
9

Teste 2  
4

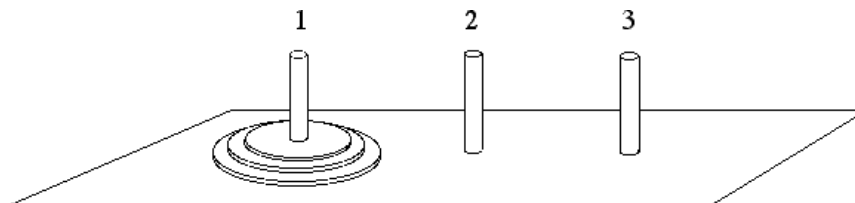
(esta saída corresponde ao exemplo de entrada acima)

### Restrições

$-1 \leq N \leq 15$  (N = -1 apenas para indicar o fim da entrada)

## 2 (OBI2003) TORRES DE HANÓI

As Torres de Hanói são um quebra-cabeça muito antigo e conhecido. Ele é constituído de um conjunto de N discos de tamanhos diferentes e três pinos verticais, nos quais os discos podem ser encaixados.



Cada pino pode conter uma pilha com qualquer número de discos, desde que cada disco não seja colocado acima de outro disco de menor tamanho. A configuração inicial consiste de todos os discos no pino 1. O objetivo é mover todos os discos para um dos outros pinos, sempre obedecendo à restrição de não colocar um disco sobre outro menor.

Um algoritmo para resolver este problema é o seguinte.

*procedimento Hanoi(N, Orig, Dest, Temp)*  
se  $N = 1$  então

```
    mover o menor disco do pino Orig para o pino Dest;  
senão  
    Hanoi(N-1, Orig, Temp, Dest);  
    mover o N-ésimo menor disco do pino Orig para o pino Dest;  
    Hanoi(N-1, Temp, Dest, Orig);  
fim-se  
fim
```

### Tarefa

Sua tarefa é escrever um programa que determine quantos movimentos de trocar um disco de um pino para outro serão executados pelo algoritmo acima para resolver o quebra-cabeça.

### Entrada

A entrada possui vários conjuntos de teste. Cada conjunto de teste é composto por uma única linha, que contém um único número inteiro  $N$  ( $0 \leq N \leq 30$ ), indicando o número de discos. O final da entrada é indicado por  $N = 0$ .

### Exemplo de Entrada

```
1  
2  
0
```

### Saída

Para cada conjunto de teste, o seu programa deve escrever três linhas na saída. A primeira linha deve conter um identificador do conjunto de teste, no formato "Teste n", onde n é numerado seqüencialmente a partir de 1. A segunda linha deve conter o número de movimentos que são executados pelo algoritmo dado para resolver o problema das Torres de Hanói com  $N$  discos. A terceira linha deve ser deixada em branco. A grafia mostrada no Exemplo de Saída, abaixo, deve ser seguida rigorosamente.

### Exemplo de Saída

```
Teste 1  
1  
  
Teste 2  
3  
(esta saída corresponde ao exemplo de entrada acima)
```

### Restrições

$0 \leq N \leq 30$  ( $N = 0$  apenas para indicar o final da entrada)

## 3 (OBI2009) NOTAS DAS PROVAS

Rosy é uma talentosa professora do Ensino Médio que já ganhou muitos prêmios pela qualidade de sua aula. Seu reconhecimento foi tamanho que foi convidada a dar aulas em uma escola da Inglaterra. Mesmo falando bem inglês, Rosy ficou um pouco apreensiva com a responsabilidade, mas resolveu aceitar

a proposta e encará-la como um bom desafio.

Tudo ocorreu bem para Rosy até o dia da prova. Acostumada a dar notas de 0 (zero) a 100 (cem), ela fez o mesmo na primeira prova dos alunos da Inglaterra. No entanto, os alunos acharam estranho, pois na Inglaterra o sistema de notas é diferente: as notas devem ser dadas como conceitos de A a E. O conceito A é o mais alto, enquanto o conceito E é o mais baixo.

Conversando com outros professores, ela recebeu a sugestão de utilizar a seguinte tabela, relacionando as notas numéricas com as notas de conceitos:

Nota	Conceito
0	E
1 a 35	D
36 a 60	C
61 a 85	B
86 a 100	A

O problema é que Rosy já deu as notas no sistema numérico, e terá que converter as notas para o sistema de letras. Porém, Rosy precisa preparar as próximas aulas (para manter a qualidade que a tornou reconhecida), e não tem tempo suficiente para fazer a conversão das notas manualmente.

### **Tarefa**

Você deve escrever um programa que recebe uma nota no sistema numérico e determina o conceito correspondente.

### **Entrada**

A entrada contém um único conjunto de testes, que deve ser lido do dispositivo de entrada padrão (normalmente o teclado). A entrada contém uma única linha com um número inteiro  $N$  ( $0 \leq N \leq 100$ ), representando uma nota de prova no sistema numérico.

### **Saída**

Seu programa deve imprimir, na saída padrão, uma letra (A, B, C, D, ou E em maiúsculas) representando o conceito correspondente à nota dada na entrada.

### **Informações sobre pontuação**

- Em um conjunto de casos de teste que totaliza 35 pontos,  $N \leq 10$ .
- Em um conjunto de casos de teste que totaliza 70 pontos,  $N \leq 50$ .

### **Exemplos**

#### **Entrada**

12

#### **Saída**

D

#### **Entrada**

87

**Saída**

A

**Entrada**

0

**Saída**

E

#### 4 (OBI2010) COMETA

O cometa Halley é um dos cometas de menor período do Sistema Solar, completando uma volta em torno do Sol a cada 76 anos; na última ocasião em que ele tornou-se visível do planeta Terra, em 1986, várias agências espaciais enviaram sondas para coletar amostras de sua cauda e assim confirmar teorias sobre suas composições químicas.

**Tarefa**

Escreva um programa que, dado o ano atual, determina qual o próximo ano em que o cometa Halley será visível novamente do planeta Terra. Se o ano atual é um ano de passagem do cometa, considere que o cometa já passou nesse ano (ou seja, considere sempre o próximo ano de passagem, não considerando o ano atual).

**Entrada**

A única linha da entrada contém um único inteiro  $A$  ( $2010 \leq A \leq 10^4$ ), indicando o ano atual.

**Saída**

Seu programa deve imprimir uma única linha, contendo um número inteiro, indicando o próximo ano em que o cometa Halley será visível novamente do planeta Terra.

**Exemplos****Entrada**

2010

**Saída**

2062

**Entrada**

10000

**Saída**

10042

**Entrada**

2062

**Saída**

2138

## 5 (OBI2013) LENÇOL

João dispõe de dois pedaços retangulares de tecido, e quer usá-los para fazer um lençol, também retangular, de dimensões  $A \times B$ . Se necessário, os dois pedaços retangulares podem ser unidos por uma costura, mas João quer que a costura seja paralela aos lados dos retângulos. Os cortes, se necessários, também devem ser paralelos aos lados dos retângulos.

Dadas as dimensões dos pedaços de tecido e do lençol, escreva um programa que determina se é possível João fazer o lençol com as dimensões desejadas.

### Entrada

A entrada contém uma única linha, com seis inteiros  $A_1$ ,  $B_1$ ,  $A_2$ ,  $B_2$ ,  $A$  e  $B$ , representando, respectivamente, as dimensões dos dois retângulos disponíveis, e as dimensões do retângulo desejado.

### Saída

Seu programa deve imprimir uma única linha contendo um caractere  $S$  se é possível fazer o lençol, e  $N$  caso contrário.

### Restrições

- $1 \leq A_1, B_1, A_2, B_2, A, B \leq 10^6$

### Exemplos

#### Entrada

4 2 3 5 4 4

#### Saída

S

#### Entrada

4 2 2 5 4 5

#### Saída

N

#### Entrada

1 2 3 5 5 2

#### Saída

S

#### Entrada

3 4 10 9 9 10

#### Saída

S

## 6 (OBI2013) DISTÂNCIA DE MANHATTAN

Maria é uma moradora de Nlogópolis, uma cidade na Nlogônia que tem uma ca-

racterística muito interessante: todas as ruas da cidade ou são orientadas no sentido norte-sul ou são orientadas no sentido leste-oeste. Isso significa que, dadas duas ruas, ou elas são paralelas ou elas são perpendiculares entre si.

Todas as ruas da cidade são de mão dupla e é possível seguir em qualquer direção em um cruzamento.

Agora Maria está atrasada para uma reunião e precisa de sua ajuda. Dadas as coordenadas iniciais de Maria e da reunião, determine o número mínimo de cruzamentos que Maria deve atravessar para chegar ao seu destino. Esse número inclui o cruzamento onde ocorrerá a reunião mas não inclui a posição inicial de Maria.

### **Entrada**

A única linha da entrada contém quatro inteiros,  $X_m$ ,  $Y_m$ ,  $X_r$ ,  $Y_r$ , indicando as coordenadas de Maria ( $X_m$ ,  $Y_m$ ) e da reunião ( $X_r$ ,  $Y_r$ ). O ponto de partida de Maria nunca será igual ao local da reunião, ou seja, pelo menos uma das coordenadas será diferente.

### **Saída**

Seu programa deve imprimir uma única linha contendo um único inteiro: o número mínimo de cruzamentos que Maria precisa atravessar para chegar até o local da reunião.

### **Restrições**

- $0 \leq X_m, Y_m \leq 1000$
- $0 \leq X_r, Y_r \leq 1000$

### **Exemplos**

#### **Entrada**

0 0 5 6

#### **Saída**

11

#### **Entrada**

52 75 120 75

#### **Saída**

68