

# 3ª Prova de Algoritmos e Estruturas de Dados I

## 03/12/2010

### Perguntas comuns e suas respostas:

- P: Tenho uma dúvida na questão tal.  
R: A compreensão do enunciado faz parte da prova.
- P: Se eu consultar algum material próprio ou de algum colega, o que acontecerá comigo?  
R: A prova é individual e sem consulta. Qualquer tentativa de fraude acarretará abertura de processo administrativo na UFPR.
- P: Posso entregar a prova a lápis?  
R: prova é um documento, portanto deve ser entregue à caneta.
- P: O que será corrigido?  
R: A lógica, a criatividade, a sintaxe, o uso correto dos comandos, a correta declaração dos tipos, os nomes das variáveis, a indentação, uso equilibrado de comentários no código e, evidentemente, a clareza. A modularidade, correto uso de funções e procedimentos, incluindo passagem de parâmetros e bom uso de variáveis locais e globais serão especialmente observados.

Escreva um programa que, dado um tabuleiro e uma lista de sub-partes retangulares do tabuleiro, retorna o número de posições que não pertencem a nenhuma sub-parte. Quando uma posição não pertence a nenhuma sub-parte dizemos que ela está *perdida*.

### Entrada

A entrada consiste de uma série de conjuntos de teste.

Um conjunto de teste começa com uma linha com três números  $W$ ,  $H$  e  $N$ , indicando, respectivamente, a largura e a altura do tabuleiro e o número de sub-partes deste. Estes valores satisfazem as seguintes restrições:  $1 \leq W$ ,  $H \leq 500$  e  $0 \leq N \leq 99$ .

Seguem  $N$  linhas, compostas de quatro inteiros  $X_1$ ,  $Y_1$ ,  $X_2$  e  $Y_2$ , tais que  $(X_1, Y_1)$  e  $(X_2, Y_2)$  são as posições de dois cantos opostos de uma sub-parte. Estes valores satisfazem as seguintes restrições:  $1 \leq X_1$ ,  $X_2 \leq W$  e  $1 \leq Y_1$ ,  $Y_2 \leq H$ .

O fim da entrada acontece quando  $W = H = N = 0$ . Esta última entrada não deve ser considerada como um conjunto de teste.

### Saída

O programa deve imprimir um resultado por linha, seguindo o formato descrito no exemplo de saída.

### Exemplo

Entrada:

```
1 1 1
1 1 1 1      {fim do primeiro conjunto de testes}
2 2 2
1 1 1 2
1 1 2 1      {fim do segundo conjunto de testes }
493 182 3
349 148 363 146
241 123 443 147
303 124 293 17      {fim do terceiro conjunto de testes}
0 0 0      {fim do conjunto de testes}
```

Saída

Não há posições perdidas.

Existe uma posição perdida.

Existem 83470 posições perdidas.