Prova Final de Algoritmos e Estruturas de Dados I
04/07/2008

P: O que será corrigido?

- 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 a clareza.
- A utilização de estruturas de dados adequadas para o problema;
- Utilização adequada de funções, procedimentos e estruturas de dados, bem como passagem de parâmetros e uso de variáveis locais.
- A eficiência do código (tempo de execução). Note que é melhor evitar percorrer a matriz a todo momento. Por isto, você DEVE pensar em usar uma estrutura auxiliar (exemplo: vetor de coordenadas) para armazenar as gotas de chuva que ainda estão na janela;

Questão 1 (Valendo 100 pontos):

A idéia é implementar um programa em Free Pascal que simule gotas de chuva caindo sobre uma janela, durante determinado intervalo de tempo.

Considere uma matriz M de tamanho $L \times C$ utilizada para representar gotas de água (caractere G) em uma janela. A cada unidade de tempo T, NG novas gotas são inseridas na matriz. Além disso, as gotas descem na matriz até que atinjam a base da janela e desapareçam. Inicialmente não há gotas na janela, pois a chuva começa quando $T = 1$.

Exemplo:

```
    Passo T=1           Passo T=2              Passo T=1000
 ----------------    ----------------        ----------------
 |  G        G  |    |        G     |        |              |
 |      G       |    |  G        G  |        | G            |
 |              |    |      G       |        |              |
 |    G   G     |    |              |  ...   |       G      |
 |              |    |    G   G     |        |  G        G  |
 |              |    |              |        |      G       |
 |              |    | G            |        |              |
 ++++++++++++++++    ++++++++++++++++        ++++++++++++++++
```

Faça um programa em Pascal que:

1. Leia o número de linhas (L) e o número de colunas (C) da matriz M, a quantidade de novas gotas a serem criadas a cada iteração (NG), e o número de iterações ($TMAX$) do programa.

   Exemplo de entrada para a matriz acima:

   ```
   7 15 5 1000
   ```

2. A cada unidade de tempo T, insira NG novas gotas na matriz. A posição de uma nova gota é dada por um procedimento cujo protótipo é:

   ```
   Procedure coordenada_nova_gota(L,C:integer; VAR x,y:integer);
   ```

   Este procedimento (que não deve ser implementado, apenas utilizado), recebe quatro parâmetros: os dois primeiros indicam o número de linhas e colunas da matriz M (L, C). Os dois últimos retornam as coordenadas (x, y) da nova gota na matriz.

3. A cada unidade de tempo T, imprima o conteúdo da matriz M, atualizando a posição das gotas G seguindo os seguintes critérios:

   (a) Quando uma gota cai sobre outra, forme-se uma gota "dupla", ou seja, ela desce duas posições a cada instante T. Caso uma nova gota caia sobre uma gota "dupla", surge uma gota "tripla", que desce três posições a cada instante T, e assim por diante.

   (b) As gotas mais rápidas podem encontrar gotas mais lentas pelo caminho, neste caso a velocidade delas é somada.