

Trabalho Prático 12

Divisória

Algoritmos e Estruturas de Dados I

Prazo: **15** de **novembro** de 2017

1 Introdução

A cidade onde Joãozinho e Zezinho moram cresceu demais. Após muita discussão e votações, o prefeito resolveu dividir a cidade em quatro. Para isso, ficou definido que seria definido um ponto na cidade, chamado ponto divisor. Duas linhas, ambas contendo o ponto divisor, uma na direção norte-sul e uma na direção leste-oeste, seriam traçadas no mapa, dividindo a cidade em quatro novas cidades, sendo chamadas de cidade do Noroeste (NO), do Nordeste (NE), do Sudeste (SE) e do Sudoeste (SO), de acordo com sua posição em relação ao ponto divisor (Figura 1).

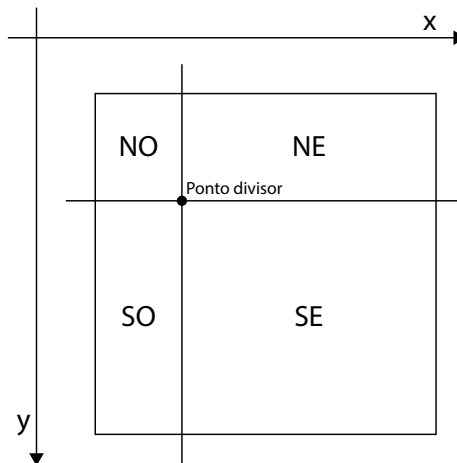


Figura 1: Divisão da cidade

Além disso, ficou definido que, um morador nas divisas norte-sul pertence à cidade à esquerda (leste) da divisa. Já um morador nas divisas leste-oeste pertence à cidade acima (norte) da divisa. Ou seja, um morador na divisa entre NO e NE, pertence à NO, assim como um morador na divisa entre NO e SO.

2 Tarefa

Pedro quer ajudar os moradores de sua cidade, então sugeriu ao prefeito que te contratasse para criar um programa que, dado o ponto divisor e diversas coordenadas de moradores, definir em qual cidade cada coordenada está localizada. Você foi prontamente requisitado para criar o programa.

3 Entrada

A entrada é formada de várias consultas. A primeira linha é composta de 1 inteiro e 2 números reais separados por espaço, no formato $N \ P_x \ P_y$ onde N indica o número de consultas que serão feitas e P_x , P_y a coordenada do ponto divisor. As próximas N linhas são formadas de 2 números reais separados por espaço no formato $C_x \ C_y$ onde C_x , C_y são as coordenadas de cada morador.

4 Saída

Para cada uma das N consultas seu programa deverá indicar a qual das cidades o morador pertence (NO, NE, SE ou SO). Na dúvida, siga o exemplo abaixo.

5 Exemplo

Entrada

```
7 0.3000000000 0.7000000000
0.3287359958 0.3917005322
0.8153209273 0.8605168107
0.1502028764 0.9333855361
0.2511852013 0.6225099598
0.2145791337 0.7980938792
0.4798102828 0.8220082538
0.8961866311 0.03161341632
```

Saída

```
NE
SE
SO
NO
SO
SE
NE
```

Outros exemplos estão disponíveis na página deste trabalho no Moodle.

6 Avisos

Avisos mandatórios para o envio do trabalho:

- O código deve ser escrito em linguagem **C**
- As entradas que serão utilizadas para teste não conterão erros, então não será necessário testar a validade das mesmas
- Não utilize chamadas da função **system** (por exemplo, **system("pause")**) pois essas podem variar de acordo com o sistema operacional e os programas instalados da máquina onde o programa está rodando
- Deixe seu código bem comentado para facilitar a correção
- Não utilize espaços ou caracteres especiais nos nomes dos arquivos. Utilize apenas os caracteres de **A** a **Z** (tanto maiúsculas como minúsculas) sem acento, os números de **0** a **9** e os caracteres **-** (hífen), **_** (*underscore*) e **.** (ponto final)
- Utilize a extensão **.c** para arquivos de código e **.h** para arquivos de cabeçalho, quando aplicar
- Se for submeter os arquivos via upload, não envie um arquivo comprimido (por exemplo, **.zip**, **.rar**, etc.). Utilize os diversos campos da aba “Submissão”, um para cada arquivo
- Envie apenas os arquivos **.c** e **.h**
- Não copie o trabalho de algum colega ou baixe da internet. Lembre-se que o prejudicado será você pois o aprendizado obtido nessa disciplina será utilizado durante diversas outras etapas do seu curso

7 Dicas

Dicas importantes para o desenvolvimento deste trabalho:

- Utilize a função **scanf** para ler as entradas do usuário
- Utilize a função **printf** para imprimir os resultados das operações
- Lembre-se que apenas a primeira linha da entrada não produz saída
- Qualquer dúvida que tiver, utilize o fórum de dúvidas no Moodle. Inicie o assunto do tópico com a tag [TP12]
- Caso prefira, participe das monitorias toda quarta das 17h às 18h na sala 2012

8 Checklist

Checklist não exaustiva de passos até a entrega do trabalho:

1. Pesquise o funcionamento das funções citadas acima para facilitar o uso
2. Implemente e compile o programa
3. Teste para o exemplo dado acima. Compare as saídas para garantir o funcionamento correto
4. Faça o mesmo do item anterior para os exemplos disponibilizados na página deste trabalho no Moodle
5. Envie o trabalho pelo Moodle, onde ele será testado automaticamente para todos os casos disponíveis
6. Caso algum teste dê errado, volte ao passo 2

Bom trabalho e divirta-se!