



Universidade de São Paulo

Instituto de Ciências Matemáticas e de Computação

Departamento de Ciências da Computação

SCC0218 — Algoritmos Avançados e Aplicações

Exercício 0.5: Desvende o mundo!

Professora:

Leo Sampaio Ferraz Ribeiro

Estagiária PAE:

Raissa Rosa dos Santos Januário

Pessoas Monitoras:

Clara Ernesto de Carvalho e Lucas Henrique Sant'Anna

Desenvolva o trabalho sem olhar o de colegas.

Se precisar de ajuda pergunte, a equipe de apoio está aqui por você.

1 Introdução

Alguns alunos do ICMC se juntaram para desenvolver o **SpotGuessr**, um jogo on-line que desafia suas habilidades de localização. Eles até criaram um lema para seu jogo "*Desvende o mundo!*". No **SpotGuessr**, os participantes recebem uma imagem panorâmica e interativa de um lugar real e devem indicar no mapa, através de coordenadas de latitude e longitude — eles ainda estão trabalhando numa forma melhor de interação com o usuário, mas vão chegar lá — onde acreditam que o local se encontra.

Após todos enviarem seus palpites, o sistema calcula a distância entre o ponto correto e o palpite de cada jogador. O vencedor da rodada é aquele cujo palpite estiver mais próximo do ponto real, e os demais jogadores são classificados em ordem crescente de distância.

Você foi convidado para implementar o protótipo do módulo do jogo responsável por calcular essas distâncias e gerar o ranking final da rodada.

2 Descrição do Problema

Seu programa deve calcular a distância de cada palpite até o ponto correto utilizando a **Fórmula de Haversine**. Para organizar os resultados de forma eficiente e em tempo real, o uso de uma *priority queue* foi um dos requisitos estabelecidos pelo grupo, permitindo identificar o melhor palpite a cada nova entrada e exibi-lo como aviso imediato. Empates na distância devem respeitar a ordem em que os palpites foram recebidos.

Ao final, o sistema deve gerar o ranking completo dos jogadores, classificando-os do palpite mais próximo ao mais distante do ponto correto.

3 Entrada

A entrada é composta por segmentos. Na primeira linha, um inteiro positivo M ($1 \leq M \leq 100\,000$) que indica o **número de jogadores**. Na segunda linha, são fornecidos dois números reais $LatC$ e $LonC$ ($-90 \leq LatC \leq 90$, $-180 \leq LonC \leq 180$), representando a **latitude e longitude do ponto correto**, respectivamente, expressos em graus decimais com até quatro dígitos após a vírgula (ponto decimal).

Em seguida, nas próximas M linhas, cada linha contém uma string que representa o **nickname** do participante (sem espaços, com até 20 caracteres), seguida de dois números reais Lat e Lon ($-90 \leq Lat \leq 90$, $-180 \leq Lon \leq 180$), correspondentes à **latitude e longitude do palpite do jogador**, também expressos em graus decimais com até quatro dígitos após a vírgula (ponto decimal).

4 Saída

O programa deve imprimir M linhas, cada uma contendo a **posição** do jogador no ranking, alinhada à **direita** em um campo de dois caracteres e seguida de um ponto, depois o **nickname**, alinhado à **esquerda** em um campo de vinte caracteres (preenchendo com espaços se for menor). Em seguida, deve aparecer um espaço, dois pontos e outro espaço (" : "), seguido da **distância** em quilômetros (km) entre o palpite e o ponto real, alinhada à **direita** em um campo de seis caracteres, com **três casas decimais** e acompanhada do sufixo " km".

As linhas devem ser ordenadas do jogador mais próximo ao mais distante. Se a distância do melhor palpite até o momento for menor que 0.050 km (50 metros), a linha deve incluir a mensagem especial [FANTASTICO] após a distância.

A cada novo palpite, o sistema exibe um aviso no formato > [AVISO] MELHOR PALPITE: <distância>km, mostrando a distância do palpite mais próximo do ponto correto até o momento.

5 Exemplo

5.1 Entrada

```
12
-23.5505 -46.6333
toninho -24.2000 -47.2000
rafael123 -23.5450 -46.6300
lana_ana -24.0000 -47.0000
alice2 -23.5510 -46.6340
gloria_gloria -23.3000 -46.2000
paschoal0 -23.5510 -46.6340
manoeleo -23.7000 -46.8000
orlando -23.55028 -46.63293
t_carolina -23.4000 -46.4000
```

```
marcelo -23.5550 -46.6350
beatriz -23.5600 -46.6400
lk12 -23.5480 -46.6310
```

5.2 Saída

```
> [AVISO] MELHOR PALPITE: 92.391km
> [AVISO] MELHOR PALPITE: 0.698km
> [AVISO] MELHOR PALPITE: 0.698km
> [AVISO] MELHOR PALPITE: 0.090km
> [AVISO] MELHOR PALPITE: 0.090km
> [AVISO] MELHOR PALPITE: 0.090km
> [AVISO] MELHOR PALPITE: 0.090km
> [AVISO] MELHOR PALPITE: 0.090km
> [AVISO] MELHOR PALPITE: 0.045km
> [AVISO] MELHOR PALPITE: 0.045km
> [AVISO] MELHOR PALPITE: 0.045km
> [AVISO] MELHOR PALPITE: 0.045km
> [AVISO] MELHOR PALPITE: 0.045km
```

RANKING

```
-----
1. orlando           : 0.045 km [FANTASTICO]
2. alice2            : 0.090 km
3. paschoal0         : 0.090 km
4. lk12              : 0.364 km
5. marcelo           : 0.530 km
6. rafael123         : 0.698 km
7. beatriz           : 1.258 km
8. manoeleo          : 23.765 km
9. t_carolina        : 29.090 km
10. gloria_gloria    : 52.253 km
11. lana_ana         : 62.375 km
12. toninho          : 92.391 km
```

6 Observações

A distância entre dois pontos na superfície da Terra pode ser calculada pela *fórmula de Haversine*:

$$d = 2R \times \arcsin \left(\sqrt{\sin^2 \left(\frac{\Delta\varphi}{2} \right) + \cos(\varphi_1) \cos(\varphi_2) \sin^2 \left(\frac{\Delta\lambda}{2} \right)} \right)$$

onde

$$R = 6371 \text{ km}$$

é o raio médio da Terra, e as variáveis são definidas como:

$$\varphi_1 = \text{LatC em radianos}, \quad \varphi_2 = \text{Lat em radianos}$$

$$\lambda_1 = \text{LonC em radianos}, \quad \lambda_2 = \text{Lon em radianos}$$

$$\Delta\varphi = \varphi_2 - \varphi_1, \quad \Delta\lambda = \lambda_2 - \lambda_1$$

Para aplicar a fórmula, as coordenadas em graus decimais devem ser convertidas para radianos pela relação:

$$\text{radianos} = \text{graus} \times \frac{\pi}{180}$$

7 Submissão

1. **Envie** seu código fonte para o run.codes.
2. **Tire Dúvidas com a Equipe de Apoio.** Se não conseguiu chegar em uma solução, dê um tempo para descansar a cabeça e converse com a equipe de apoio sobre a dificuldade encontrada.