

4ª exercício avaliativo (EA4)
Programação II (INF16153) - UFES
20 de Junho de 2023

Nesta atividade, vamos simular um sistema de defesa aérea de mísseis. A ideia é simples, dado o território de uma cidade, dois tipos de defesa aérea podem ser posicionados para defender possíveis mísseis disparados contra o território. Se o míssil de ataque atingir a zona de proteção da defesa aérea, ele poderá ser neutralizado com um míssil defesa. Se não, ele causará um dano na cidade. Para facilitar as coisas, ao invés de longitude e latitude, tudo é simulado em um plano cartesiano com coordenadas X e Y (vide exemplo no final da descrição)

Cidade: a cidade é representada por um plano cartesiano 2D que terá um tamanho máximo de X e Y que será fornecido como entrada

Defesa aérea: a defesa área pode ser de dois tipos, quadrada (Q) ou circunferência (C). Esse tipo define a área de proteção. Serão fornecidas as seguintes informações de uma defesa aérea:

- Nome, sempre começando pela letra D (exemplo: D-256)
- Coordenada X e Y em que a defesa é posicionada na cidade, respeitando o tamanho da cidade
- Tipo da defesa aérea (Q ou C)
- Tamanho de projeção. Esse valor determina a área de proteção da defesa aérea. Se a defesa for do tipo C, o tamanho representa o raio de uma circunferência, se for do tipo Q, representa a metade da diagonal de um quadrado
- Poder de neutralização do míssil de defesa
- Quantidade de mísseis de defesa

Míssil de ataque: é um projétil que tem como objetivo atingir um ponto (X, Y) de uma cidade. Serão fornecidas as seguintes informações:

- Nome, sempre começando com M (exemplo: M-698)
- Coordenada X e Y de ataque
- Poder de ataque do míssil

Regras de funcionamento

- Seu programa vai receber como entrada informações da cidade, defesa(s) aérea(s) e míssil(eis). Baseado nesses dados, ele deve fornecer um relatório como resultado do ataque
- Um míssil de ataque é neutralizado pela defesa aérea se:
 - Ele atinge um ponto da cidade em que está protegido pela mesma
 - Se o poder de neutralização do míssil de defesa é maior do que o de ataque do míssil de ataque
 - Se a quantidade de míssil de defesa é maior do que zero
- Se um míssil de ataque é neutralizado, nenhum dano é causado na cidade. Se ele atinge um ponto que não possui defesa, o dano causado é igual ao poder de ataque do míssil
- Um míssil de ataque pode ser parcialmente neutralizado. Isso acontece quando ele atinge uma área defendida, mas o poder de defesa é menor do que o de ataque. Neste caso, o dano causado é igual ao módulo da diferença entre os dois valores
- Cada defesa aérea lança apenas um míssil de defesa por míssil de ataque que invade a zona defendida. Sempre que uma defesa é utilizada, a quantidade total de mísseis de defesa é diminuída. Ela só é funcional se essa quantidade é maior do que zero
- Se um míssil de ataque atinge um ponto de interseção entre duas defesas, as duas lançam um míssil de defesa
- Se um míssil de ataque é lançado para fora da zona da cidade, nenhum dano é causado

O relatório final deve conter as seguintes informações:

- Quantidade final de mísseis de defesa de cada defesa aérea
- Dano causado por cada um dos mísseis lançados
- Porcentagem de efetividade da defesa, que nada mais é do que o dano total mitigado no ataque dividido pelo dano total que o ataque poderia causar. Na qual, total de ataque é o somatório de poder de ataque de todos os mísseis lançados

Fórmulas úteis

- Teorema de Pitágoras: $h^2 = a^2 + b^2$
- Diagonal de um quadrado: $d = l\sqrt{2}$
- Distância Euclidiana: $d(p, q) = \sqrt{\sum_{i=1}^n (p_i - q_i)^2}$

Padrão de entrada

Seu programa deve ler da entrada padrão informações da cidade, defesas aéreas em mísseis, respeitando essa ordem. A cidade é representada pela primeira linha respeitando o padrão:

C x y

Na sequência poderá existir uma ou mais defesas aéreas respeitando o seguinte padrão:

D-nome x y tipo tamanho poder quantidade

Por fim, poderá existir um ou mais mísseis de ataque no seguinte padrão:

M-nome x y poder

A entrada acaba com um caractere F. Um exemplo válido:

```
C 130.0 100.0
D-1 30.0 30.0 Q 28.30 50.3 4
D-2 93.1 69.0 C 28.30 42.6 8
M-1 90.2 60.2 36.1
M-2 50.7 70.2 32.6
M-3 42 22 60.9
M-4 81 30 35.5
F
```

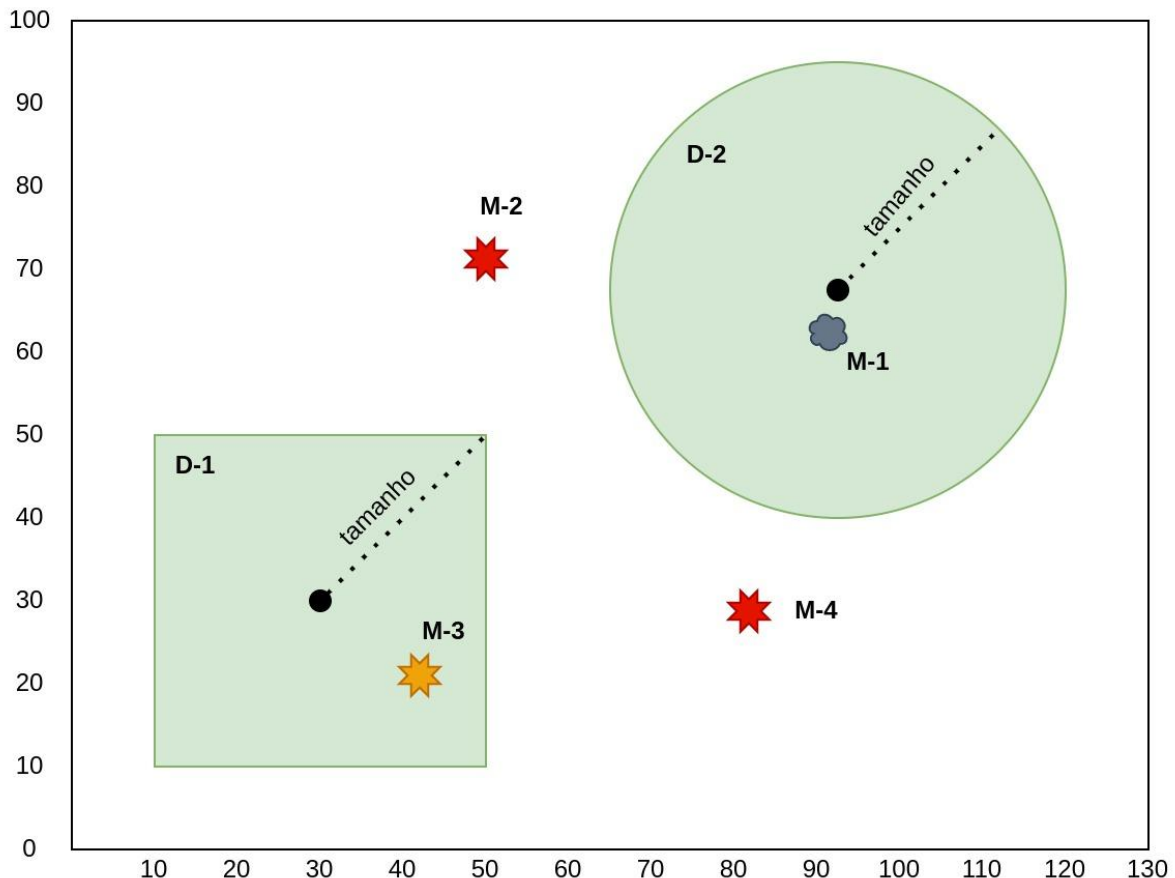
Padrão de saída

A saída deverá ser impressa na tela (utilizando `printf`) de acordo com o padrão a seguir:

```
D-1: 3
D-2: 7
M-1: 0.00
M-2: 32.60
M-3: 10.60
M-4: 35.50
Efetividade: 52.33%
```

Exemplo ilustrativo

Exemplo de entrada e saída anterior, pode ser ilustrado pela figura a seguir:



A figura ilustra a área da cidade, as defesas aéreas D-1 e D-2 posicionadas, bem como suas zonas de proteção. O míssil M-1 é totalmente neutralizado. M-2 e M-4 causam danos totais na cidade, e o M-3 é parcialmente neutralizado.

Regras gerais

- A atividade é **individual**. Todas as questões serão testadas e plágio não será tolerado
- Seu programa deve implementar um TAD com **encapsulamento total (opaco)** para defesa aérea e para o míssil de ataque
- Toda lógica de negócio deve ser implementada fora da sua `main()`. Em outras palavras, você só deve fazer chamadas de funções do seu TAD (como realizado em exercícios na sala de aula)
- Você deve utilizar **pelo menos uma função de callback** no seu programa
 - Sugestão: uma boa utilização é uma função que verifique se um míssil está pode ser neutralizado ou não. Isso depende da forma da defesa e pode ser passado como parâmetro para uma outra função que calcule o dano que míssil causou
- Seu programa será testado com o Valgrind para detectar vazamento de memória e erros de alocação. É sua responsabilidade liberar toda memória alocada.
- **Números de ponto flutuante devem ter precisão simples e apenas duas casas decimais devem ser impressas**
- Você deve fornecer um **Makefile** que gere um arquivo executável chamado **EA4**
- **O seu programa será executado da seguinte forma:**
 - `./EA4 <entrada> saída`
- Haverá **correção automática**, portanto, siga os padrões de saída corretamente
 - O corretor ignora espaços e quebras de linha

- Porém, se você escrever informações na tela, ele retornará um erro na saída (por exemplo: “digite um time”). Portanto, escreva somente o que foi solicitado na tela
- Organização, modularização e boas práticas de programação são critérios fundamentais de avaliação.
- A submissão da atividade será realizada via Github de acordo com as instruções já conhecidas
 - Criar uma pasta EA4 e submeter a atividade dentro dela