

# Aula 2: Upsolving do Warmup 1

*Disciplina:* Maratona de Programação 1

**Profs. Edmilson Marmo e Luiz Olmes**

*edmarmo@unifei.edu.br, olmes@unifei.edu.br*

## Problema A: Só para entender

---

- ▶ **Total de Submissões:** 150
- ▶ **Submissões Aceitas:** 47 (31%)
- ▶ Como o próprio enunciado diz, este problema é muito simples de ser resolvido.
- ▶ É necessário ler os três valores de entrada, ordená-los, e imprimir os 3 valores em ordem crescente.
- ▶ A forma mais direta de resolver este problema é usar a função `sort()`, da linguagem C++.
  - ▶ Mas eu não conheço a função `sort()`...
  - ▶ Bom, agora você será apresentado a ela. Use-a sempre que for preciso!

## Problema A: Só para entender

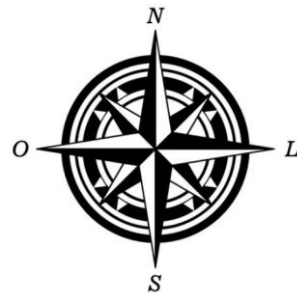
---

- ▶ Inicialmente, pode-se declarar um vetor `v` de números inteiros, com 3 posições:
  - ▶ `int v[3];`
- ▶ Para realizar a leitura, como são apenas 3 valores, não há necessidade de usar um laço. Pode-se proceder diretamente com as posições dos vetores na função `scanf()` ou `cin`:
  - ▶ `scanf("%d %d %d", &v[0], &v[1], &v[2]);`
  - ▶ `cin >> v[0] >> v[1] >> v[2];`
- ▶ Após a leitura, invoca-se a função `sort()`:
  - ▶ Ela pertence à `#include <algorithm>` (ou insira apenas `#include <bits/stdc++.h>`)
  - ▶ A função ordena os elementos que estão no intervalo `[first, last)`.
  - ▶ No caso, para ordenar o vetor `v`, usa-se: `sort(v, v + 3);`

## Problema B: Esquerda, volver!

---

- ▶ **Total de Submissões:** 53
- ▶ **Submissões Aceitas:** 29 (55%)
- ▶ Pode-se criar duas variáveis inteiras para somar quantas vezes as letras D e E aparecem na string de entrada.
- ▶ Perceba que a cada 4 comandos “direita, volver”, o recruta dá uma volta completa e estará novamente de frente para o Norte.
  - ▶ O mesmo para cada 4 comandos “esquerda, volver”.



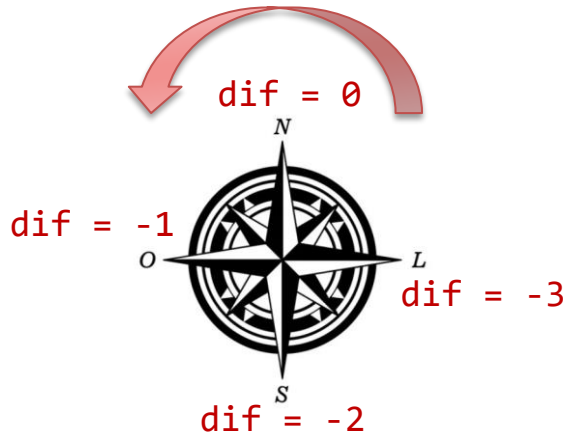
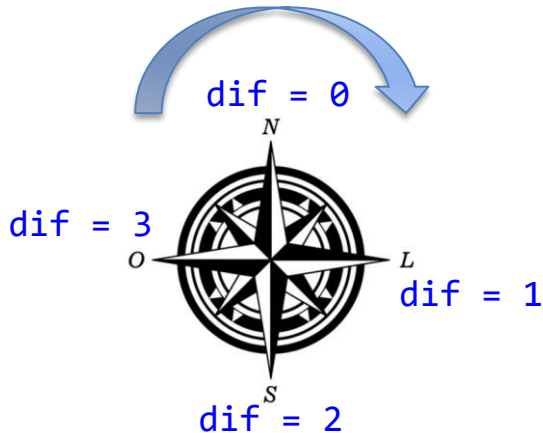
## Problema B: Esquerda, volver!

- Então, basta pegar o resto da divisão por 4 para cada variável (D e E), para calcular a direção que o recruta estará posicionado.

▶  $D = D \% 4;$

▶  $E = E \% 4;$

- No final, pode-se fazer a diferença de uma variável por outra e resolver com condicionais `if/else`. Para  $\text{dif} = D - E$ , tem-se que valores positivos indicam que ele vira para a direita, e negativos para a esquerda:



1. `if(dif == 0) printf("N\n");`
2. `else if(dif == 1 || dif == -3)`
3. `printf("L\n");`
4. `else if(dif == 2 || dif == -2)`
5. `printf("S\n");`
6. `...`

## Problema C: Aprendendo a Contar

---


- ▶ **Total de Submissões:** 166
- ▶ **Submissões Aceitas:** 46 (28%)
- ▶ Após ler a entrada, este problema pode ser resolvido com um único laço, de modo que os múltiplos da variável **x** são pulados.

```
1.  while(i <= f)
2.  {
3.      if(i % x) printf("%d ", i);
4.
5.      i++;
6.  }
```

- ▶ Note que **i % x** será **zero** para múltiplos de **x**, e **if(0)** é falso.

## Problema D: Guarda Costeira

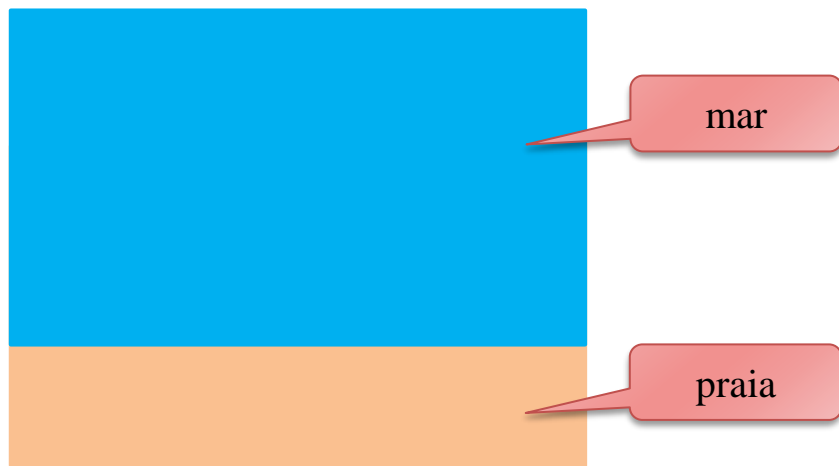
---

- ▶ **Total de Submissões:** 187
- ▶ **Submissões Aceitas:** 22 (12%)
- ▶ Vários alunos não conseguiram resolver esta questão porque estão com um sério **problema de interpretação de texto**.
  - ▶ Essa habilidade você deve trazer do seu Ensino Fundamental!!! 
  - ▶ Vários problemas da maratona possuem enunciados longos, para **confundir o leitor** e/ou para **esconder detalhes de sua solução**.
  - ▶ Interpretar corretamente o enunciado é crucial para se tornar um maratonista!
- ▶ Vejamos como esquematizar a solução, de acordo com o enunciado.

## Problema D: Guarda Costeira

---

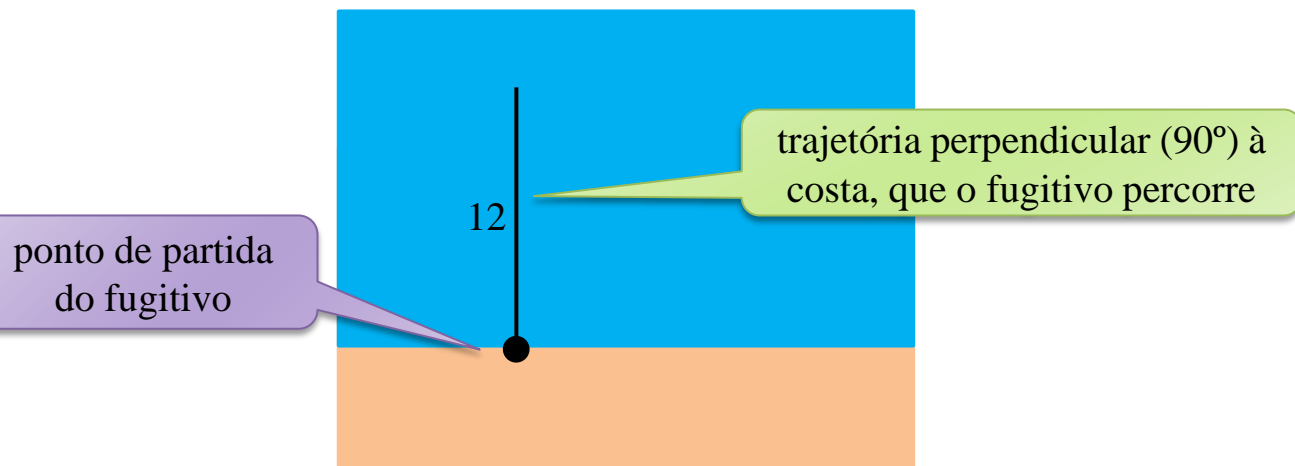
- ▶ *O fugitivo pretende seguir perpendicularmente à costa (da praia) em direção ao limite de águas internacionais, que fica a 12 milhas náuticas.*





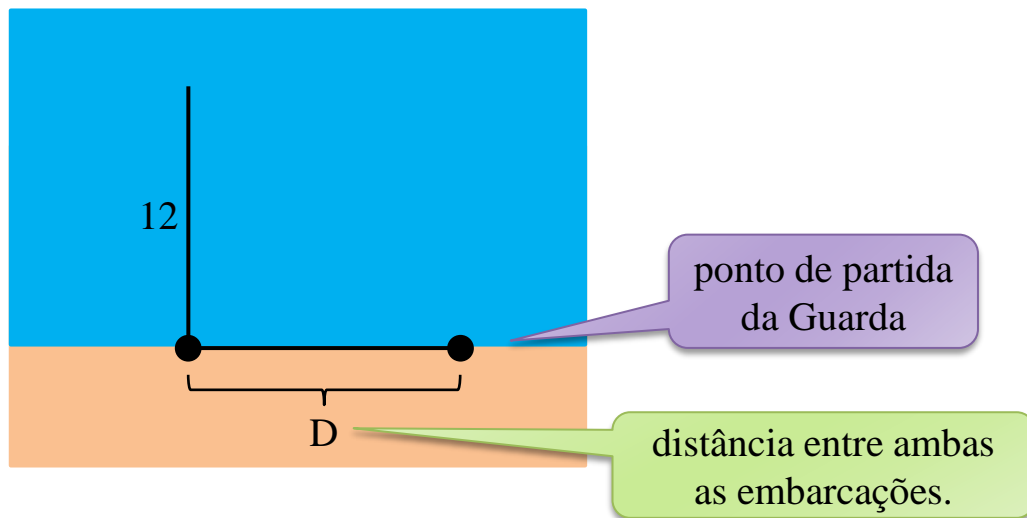
## Problema D: Guarda Costeira

- ▶ O fugitivo pretende seguir *perpendicularmente* à costa (da praia) em direção ao limite de águas internacionais, que fica a *12* milhas náuticas.



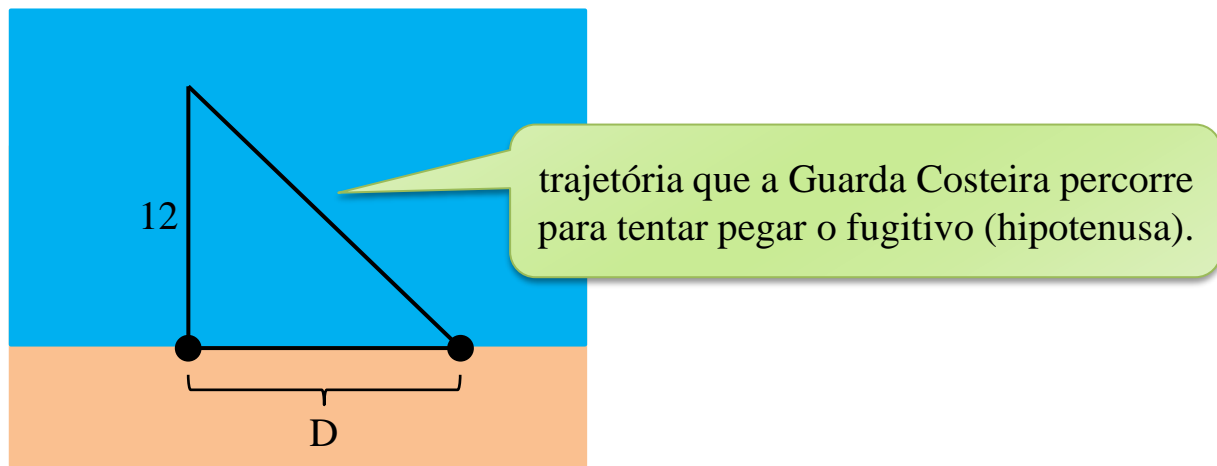
## Problema D: Guarda Costeira

- Supondo que ambas as embarcações (fugitivo e Guarda Costeira) *partam da costa* exatamente no mesmo instante, com uma *distância de  $D$  milhas náuticas entre elas...*



## Problema D: Guarda Costeira

- *Supondo que ambas as embarcações (fugitivo e Guarda Costeira) partam da costa exatamente no mesmo instante, com uma distância de  $D$  milhas náuticas entre elas...*



## Problema D: Guarda Costeira

---

- ▶ Foram dadas as medidas dos catetos (12 e D). Basta computar a hipotenusa do triângulo retângulo que se formou.
- ▶ A seguir, de posse das velocidades de ambas as embarcações, calculamos o tempo que o fugitivo gasta para percorrer 12 milhas e o tempo que a Guarda gasta para percorrer o valor de hipotenusa:
  - ▶ Movimento uniforme:  $\text{velocidade média} = \text{distância} / \text{tempo}$
  - ▶ Note que **divisão de int por int resulta em int**. Use valores reais (float / double).
- ▶ Se o tempo do fugitivo for menor que o tempo da Guarda, ele escapa. Caso contrário, a Guarda o captura.

## Problema E: Triângulo

---

- ▶ **Total de Submissões:** 62
- ▶ **Submissões Aceitas:** 30 (48%)
- ▶ Este problema já foi proposto na Aula 01, no dia 21/08, e apareceu novamente no Warmup 1.
- ▶ Considerando que apenas 30 submissões foram aceitas, e temos mais de 60 alunos matriculados, **mais da metade dos alunos não fez o dever de casa**, que é **resolver todas as questões que não conseguiu resolver em sala de aula**.
  - ▶ Ou, ao menos, tentar resolver...



## Problema E: Triângulo

---

- ▶ Dadas 4 varetas, com medidas variadas, a intenção é formar um triângulo.
- ▶ Não adianta selecionar as duas menores varetas e tentar formar triângulo com a maior delas, pois pode-se violar a condição de existência do triângulo.
- ▶ Perceba que, se as duas menores (A e B) formarem um triângulo com a maior (D), elas também formariam um triângulo com a 3ª menor vareta (C).
  - ▶ Isto é, as 3 menores formam um triângulo.
  - ▶ O mesmo raciocínio se aplica à menor com as duas maiores. Se elas formam um triângulo, as duas maiores também formariam triângulo com a segunda menor.
- ▶ Nesse caso, basta selecionar as 3 menores ou as 3 maiores varetas, e verificar se elas formam um triângulo.

## Problema E: Triângulo

---

- ▶ De forma similar ao Problema A, basta ler as medidas em um vetor, ordená-lo com a função `sort()` e verificar se:
  - ▶ As 3 menores formam um triângulo:  $v[0] + v[1] > v[2]$
  - ▶ OU
  - ▶ As 3 maiores formam um triângulo:  $v[1] + v[2] > v[3]$

```
1.  if( v[0] + v[1] > v[2] || v[1] + v[2] > v[3] )
2.      printf("S\n");
3.  else
4.      printf("N\n");
```

# Aula 2:

# Upsolving do Warmup 1

*Disciplina:* Maratona de Programação 1

**Profs. Edmilson Marmo e Luiz Olmes**

*edmarmo@unifei.edu.br, olmes@unifei.edu.br*

