

# ESTRUTURA DE DADOS

---



# AGENDA 07/03/2017

---

- Ponteiros
- Vetores
- Matrizes
- Alocação dinâmica

# PONTEIROS

---

- Variável que armazena um endereço de memória
- Uma região da memória armazena o endereço de outra
- Exemplo de declaração: `int *p;`
- Operador `*` (conteúdo de)
- Operador `&` (endereço de)

# BATALHA NAVAL

---

- N participantes
- Cada participante recebe um número de 0 até  $n-1$  (endereço)
- Divide-se dois grupos
- Cada grupo possui um encouraçado (4 partes), 2 contratorpedeiro (3 partes) e 2 submarinos (1 parte). O grupo pode distribuir as embarcações entre seus participantes, mas com as partes de forma contínua a partir do primeiro número de endereço. (ex: contratorpedeiro em 21,22 e 23)
- São escolhidos k participantes de cada grupo para apontar os números do disparo
- Cada grupo sugere um endereço no espaço de endereçamento do outro grupo
- Ganha quem acertar mais partes

# DEFINIÇÕES

---

- Vetor (Array): `int vetor[10] = {10,39,943,1,94,7,6,8,5,1};`
- Matriz: `int matriz[2][2] = {{1,2},{3,4}};`



# ITERANDO

---

- Vetor -> for(...
- Matrix -> for(... for(...

# E NA BATALHA NAVAL?



# ATIVIDADE

---

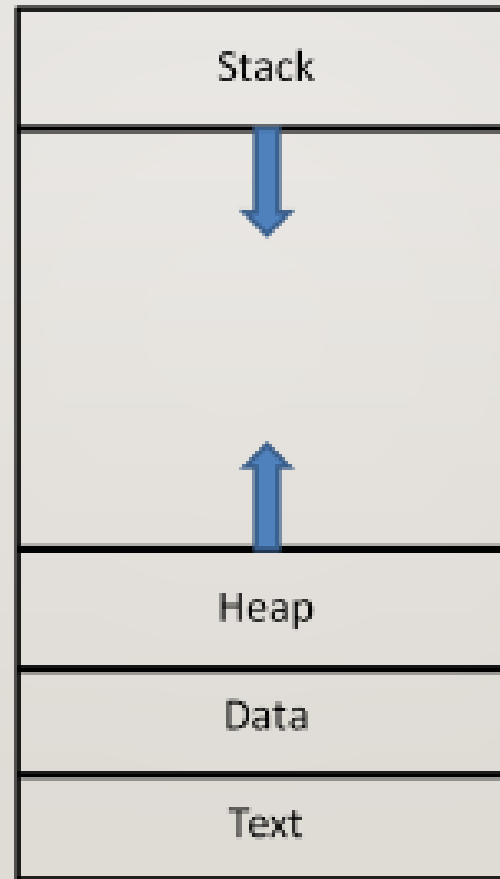
- Escreva um programa para armazenar dados em um vetor de 10 elementos e depois imprima-os
- Repita para uma matriz 3x4 com quebra de linha entre as linhas da matriz





# ALOCAÇÃO DINÂMICA

---



# ALOCAÇÃO DINÂMICA

---

- Função sizeof e malloc

***tipo \*ponteiro = (tipo\*) malloc(n\*sizeof(tipo));***

***tipo \*\*ponteiro = (tipo\*\*) malloc(n\*sizeof(tipo));***

***free(ponteiro);***

# ALOCAÇÃO DINÂMICA

---

- Implemente um programa que recebe um número de elementos junto com os valores inteiros a serem processados e aloque um vetor para eles, calcule a média e a imprima

# STRUCT

---

```
struct ponto {  
    float x;  
    float y;  
}
```

```
struct ponto coord01;  
coord01.x = 10.0;  
coord01.y = 2.4;
```