Vetores – Parte 1

Profa. Elloá B. Guedes
www.elloaguedes.com

UEA
UNIVERSIDADE
DO ESTADO DO
AMAZONAS

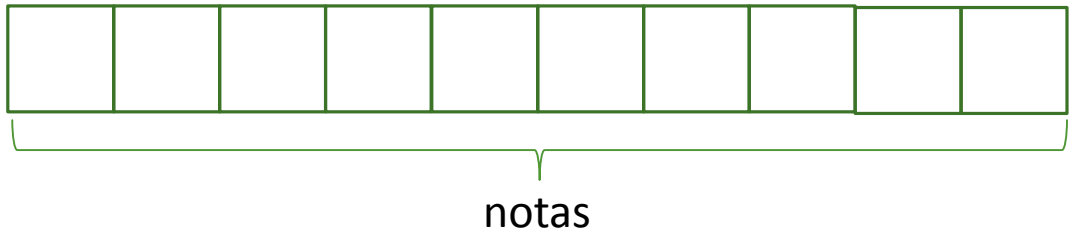
Vetores

- **Vetores** ou **arrays unidimensionais** representam um conjunto de valores do mesmo tipo

- Declaração

  tipo nome[tamanho]

- Exemplo:
- float notas[10]



Vetores

- A contagem das posições do vetor começa em ZERO
  - notas[0] é o primeiro elemento do vetor notas do tipo float

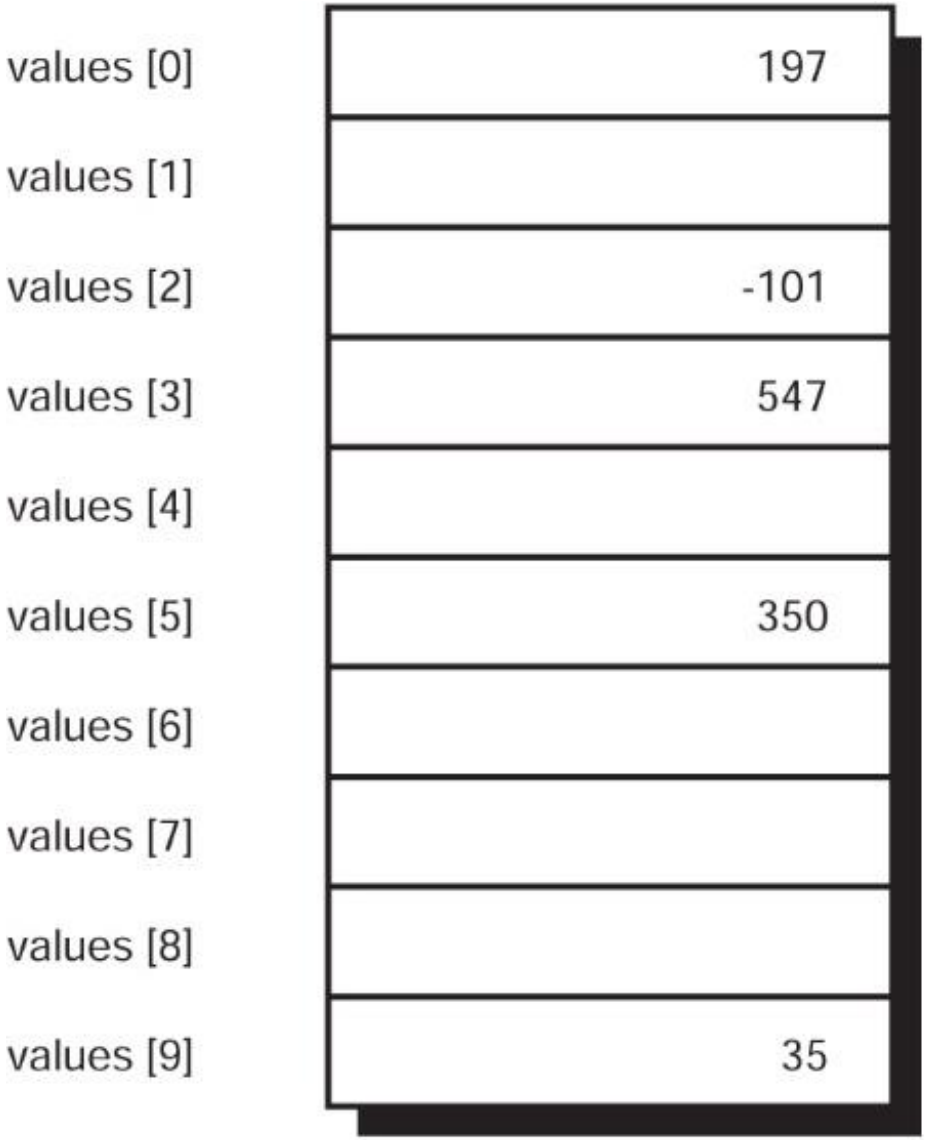
- Elementos de arrays do tipo int, float ou char podem ser manipulados da mesma maneira que as variáveis correspondentes
  - Atribuição
  - Exibição de valores
  - Adicionar, subtrair, entre outros

- Valor armazenado em uma posição no vetor é acessado por meio do índice

Vetores

```
#include <stdio.h>

int main (void)
{
     int  values[10];
     int  index;

     values[0] = 197;
     values[2] = -101;
     values[5] = 350;
     values[3] = values[0] + values[5];
     values[9] =
     values[5] / 10;
     --values[2];

     for ( index = 0;  index < 10;  ++index )
          printf ("values[%i] = %i\n", index, values[index]);

     return 0;
}
```

Vetores


values [0]
197
values [1]
values [2]
-101
values [3]
547
values [4]
values [5]
350
values [6]
values [7]
values [8]
values [9]
35

Vetores

```
values[0] = 197
values[1] = 0
values[2] = -101
values[3] = 547
values[4] = 0
values[5] = 350
values[6] = 0
values[7] = 0
values[8] = 0
values[9] = 35
```

Exercício

- Escreva um algoritmo que leia um vetor com 10 posições de números inteiros e verifique se um determinado valor, também digitado pelo usuário, está no vetor. Caso o elemento não esteja no vetor, apresente uma mensagem informando tal situação.

Exercício

- Escrever um algoritmo que leia 2 vetores X(10) e Y(10) e os escreva. Crie, a seguir, um vetor para cada operação abaixo:
- b) A diferença entre X e Y.
- c) A soma entre X e Y.
- d) O produto entre X e Y.
- e) O resto da divisão de X por Y

- Ao final, escrever o conteúdo de cada operação.

Exercício

- Leia um vetor de 12 posições e, em seguida, leia também dois valores X e Y quaisquer correspondentes a duas posições no vetor. Ao final o algoritmo deverá escrever a soma dos valores encontrados entre as respectivas posições X e Y.

Exercício

- Leia um vetor de 16 posições e troque os 8 primeiros valores pelos 8 últimos. Escreva ao final o vetor obtido.

Exercício

- Faça um programa que leia valores inteiros correspondentes a 90 amostras utilizadas em uma pesquisa (valores entre -400 e 400), armazene-os em um vetor e o escreva. Substitua a seguir todos os valores negativos desse vetor por 999 e escreva o vetor modificado, bem como o número de valores que sofreram substituição.

Vetores como Contadores

- Vetores podem ser utilizados como contadores para um determinado valor

- O valor de cada posição do vetor pode ser incrementado de acordo com uma dada condição

Exercício

- Leia um vetor de 10 posições e verifique se existem valores iguais (repetidos) e os escreva.

Exercícios

- Elabore um algoritmo para criar um vetor de 30 posições com valores reais. As 15 primeiras posições serão informadas pelo usuário e as posições seguintes serão os números digitados pelo usuário, mas na ordem inversa. Ao final, o algoritmo deverá imprimir o vetor.

Inicialização de Vetores

- É possível inicializar o valor de um vetor no momento em que este é declarado

- int contadores[5] = {0, 0, 0, 0, 0};

- int inteiros[5] = {0, 1, 2, 3, 4};

- char vogais[5] = {‘a’,’e’,’i’,’o’,’u’};

Inicialização de Vetores

- Não é necessário definir todos os valores do vetor durante a inicialização

- float exemplo[500] = {100.0, 300.0, 500.5};

- É possível definir valores específicos na inicialização

- float exemplo[500] = { [2]= 500.5, [1] = 300.0, [7] = 100.0};

Inicialização de Vetores

- É possível utilizar valores pré-definidos na inicialização dos vetores

- int x = 1233;
- int a[10] = {[9] = x+1, [2] = 3, [1] = 2, [0] = 1};

Exercício

```
#include <stdio.h>

int main (void)
{
   int  array_values[10] = { 0, 1, 4, 9, 16 };
   int  i;

   for ( i = 5;  i < 10;  ++i )
       array_values[i] = i * i;

   for ( i = 0;  i < 10;  ++i )
       printf ("array_values[%i] = %i\n", i, array_values[i]);

   return 0;
}
```