



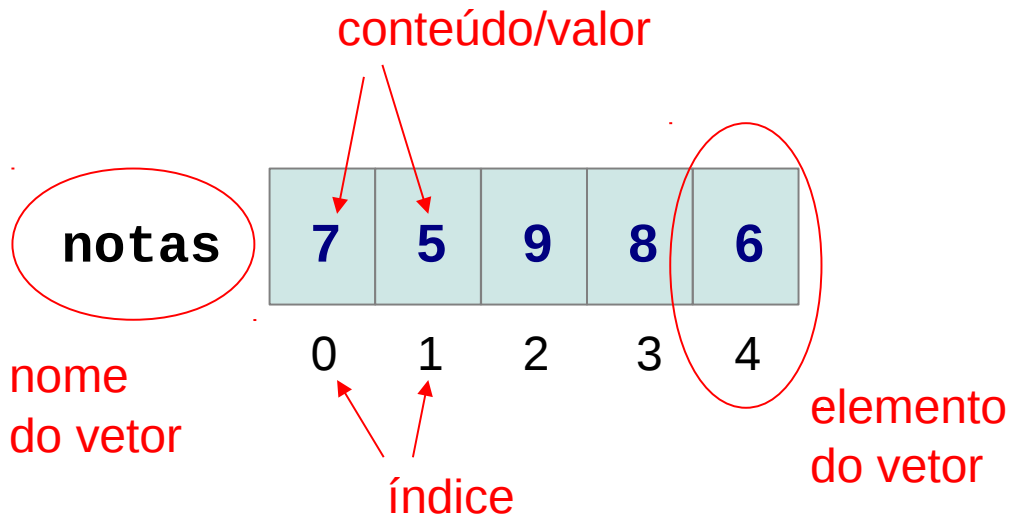
LÓGICA DE PROGRAMAÇÃO

PROF^a. M.Sc. JULIANA H Q BENACCHIO

- Também chamadas de **Vetor**, **Arranjo**, **Array** ou **Matriz unidimensional**
- São coleções de item de dados relacionados
- São estruturas de dados consistindo em itens de dados do mesmo tipo relacionados
- Portanto, um **vetor** é um grupo de variáveis que contém valores que são todos do mesmo tipo

Vetor

- Um **vetor** são posições contíguas na memória que possuem o mesmo nome e o mesmo tipo
- Por exemplo: `nota1`, `nota2`, `nota3`, `nota4`, `nota5`



<code>notas[0]</code>	7
<code>notas[1]</code>	5
<code>notas[2]</code>	9
<code>notas[3]</code>	8
<code>notas[4]</code>	6

- O vetor **notas** contém 5 elementos

- Para declarar um vetor utiliza-se a seguinte forma geral:

```
tipo nome_variavel[tamanho];
```

- Para declarar um vetor com 5 elementos:

```
int notas[5];
```

- Na linguagem C a numeração começa sempre em zero. Isto significa que os dados serão indexados de **0** a **n-1**, onde **n** é o tamanho do vetor.
- Qualquer um desses elementos pode ser referenciado fornecendo-se o nome do vetor seguido pelo número da posição do elemento (índice) entre colchetes []

- Para o vetor de exemplo: **int notas[5];**
 - **notas[0]** é o primeiro elemento do vetor **notas**
 - **notas[1]** é o segundo elemento do vetor **notas**
 - **notas[2]** é o terceiro elemento do vetor **notas**
 - **notas[3]** é o quarto elemento do vetor **notas**
 - **notas[4]** é o quinto e último elemento do vetor **notas**
- Portanto, o *i-ésimo* elemento do vetor **notas** é referido sempre como **notas[i-1]**

- Um índice deve ser um inteiro não-negativo
- O compilador da linguagem C não verifica se o índice está dentro dos limites válidos do vetor.
- Este é um cuidado que o programador deve tomar para que não ocorram bugs no programa ao tentar acessar uma posição inexistente do vetor.

- É possível utilizar uma expressão como um índice
- Por exemplo, assumindo que a variável **a** é 2 e a variável **b** é 1, a instrução

notas[a+b] = 9;

- irá atribuir o valor 9 ao elemento **notas[3]**

- As operações devem ser feitas acessando cada um dos elementos do vetor
- Por exemplo, para calcular a soma dos valores contidos nos primeiros três elementos do vetor **notas** e armazenar o resultado na variável **soma**:

```
soma = notas[0] + notas[1] + notas[2]
```

- É possível declarar vetores de qualquer tipo
- Cada elemento de um vetor irá conter um valor do tipo declarado

```
int notas[5];
```

```
float valores[20];
```

```
double salario[15];
```

```
char letras[10];
```

- Para fazer a leitura das 5 notas e armazenar no vetor:

```
#include <stdio.h>
```

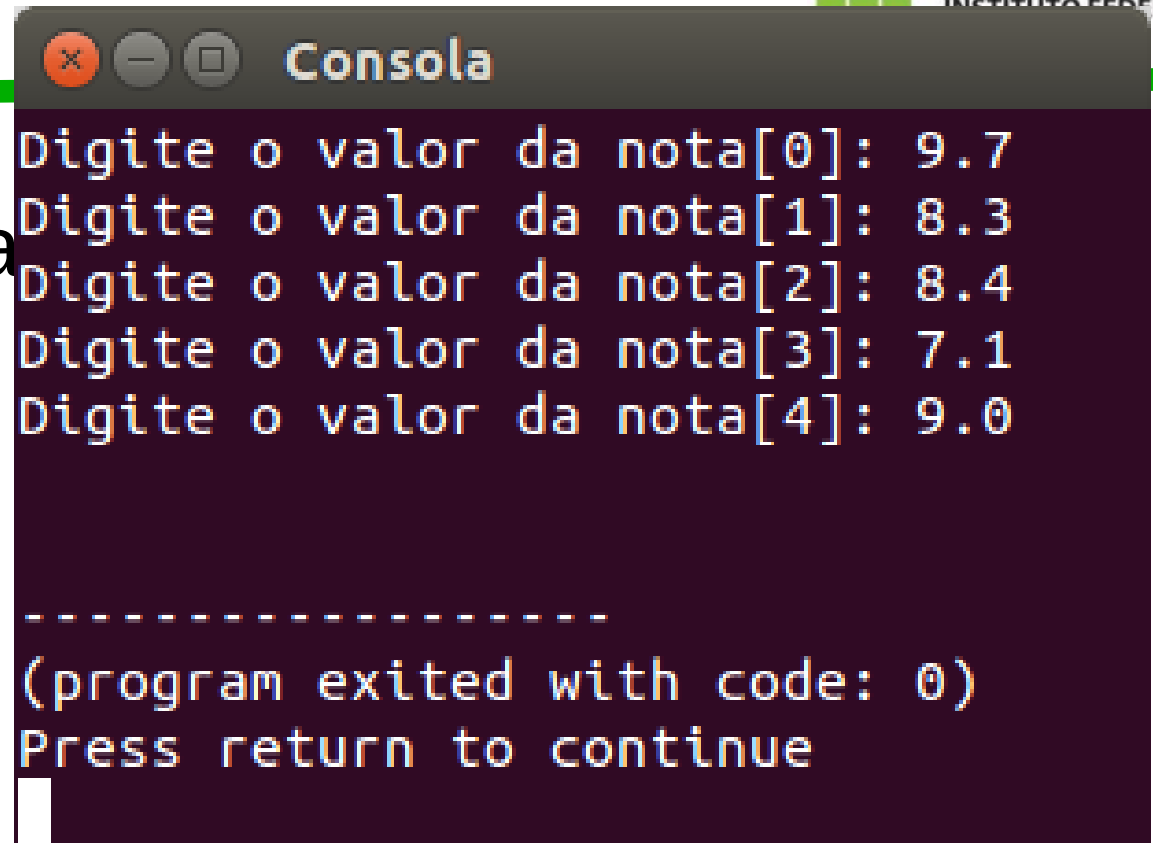
```
int main(){  
    float notas[5];  
    int i;  
    for(i=0; i<5; i++)  
    {  
        printf("Digite o valor da nota[%d]: ", i);  
        scanf("%f", &notas[i]);  
    }  
    return 0;  
}
```

Vetor

- Para fazer a leitura de um vetor:

```
#include <stdio.h>
```

```
int main(){  
    float notas[5];  
    int i;  
    for(i=0; i<5; i++)  
    {  
        printf("Digite o valor da nota[%d]: ", i);  
        scanf("%f", &notas[i]);  
    }  
    return 0;  
}
```



```
Consola  
Digite o valor da nota[0]: 9.7  
Digite o valor da nota[1]: 8.3  
Digite o valor da nota[2]: 8.4  
Digite o valor da nota[3]: 7.1  
Digite o valor da nota[4]: 9.0  
  
-----  
(program exited with code: 0)  
Press return to continue
```

- Podemos criar um vetor e inicializar seus elementos com um inicializador de vetor, que é uma lista de expressões separadas por vírgulas colocadas entre chaves ({ e })

```
int v[5] = {10, 20, 30, 40, 50}
```

- Por exemplo, a declaração com a lista inicializadora cria o vetor com 5 elementos com os valores de índice 0, 1, 2, 3 e 4.
- O elemento **v[0]** é inicializado com 10, o **v[1]** é inicializado com 20 e assim por diante
- Ou seja, o vetor é criado com a quantidade de elementos na lista de inicializadores

- Também é possível inicializar um vetor com apenas alguns elementos.

```
int numeros[5] = {2, 4, 6}
```

- É o equivalente a

```
int numeros[5] = {2, 4, 6, 0, 0}
```

- Neste caso, quando o número de itens inicializados é menor que o número total de itens do vetor, os itens não inicializados são automaticamente zerados.

- Podemos inicializar um vetor sem especificar a quantidade de elementos.
- Nesse caso, o tamanho do vetor é determinado pelo número de elementos na lista inicializadora

```
int valores[] = {9, 2, 7}
```



```
#include <stdio.h>

int main(){
    float notas[5], soma = 0, media;
    int i;
    for(i=0; i<5; i++)
    {
        printf("Digite o valor da nota[%d]: ", i);
        scanf("%f", &notas[i]);
        soma = soma + notas[i];
    }
    printf("\nCalculo da media\n");
    media = soma/5;
    printf("\nAs notas digitadas foram: \n");
    for(i=0; i<5; i++)
    {
        printf("nota[%d] = %.1f\n", i, notas[i]);
    }
    printf("\nMedia = %.1f", media);
    return 0;
}
```



```
Consola
Digite o valor da nota[0]: 7.4
Digite o valor da nota[1]: 8.9
Digite o valor da nota[2]: 9.1
Digite o valor da nota[3]: 8.3
Digite o valor da nota[4]: 7.5

Calculo da media

As notas digitadas foram:
nota[0] = 7.4
nota[1] = 8.9
nota[2] = 9.1
nota[3] = 8.3
nota[4] = 7.5

Media = 8.2

-----
(program exited with code: 0)
Press return to continue
█
```

```
#include <stdio.h>

int main(){
    float notas[5], soma = 0, media;
    int i;
    for(i=0; i<5; i++)
    {
        printf("Digite o valor da %dª nota : ", i+1);
        scanf("%f", &notas[i]);
        soma = soma + notas[i];
    }
    printf("\nCalculo da media\n");
    media = soma/5;
    printf("\nAs notas digitadas foram: \n");
    for(i=0; i<5; i++)
    {
        printf("%dª nota = %.1f\n", i+1, notas[i]);
    }
    printf("\nMedia = %.1f", media);
    return 0;
}
```



```
Consola
Digite o valor da 1ª nota : 7.4
Digite o valor da 2ª nota : 8.9
Digite o valor da 3ª nota : 9.1
Digite o valor da 4ª nota : 8.3
Digite o valor da 5ª nota : 7.5

Calculo da media

As notas digitadas foram:
1ª nota = 7.4
2ª nota = 8.9
3ª nota = 9.1
4ª nota = 8.3
5ª nota = 7.5

Media = 8.2

-----
(program exited with code: 0)
Press return to continue
█
```