UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Vetores

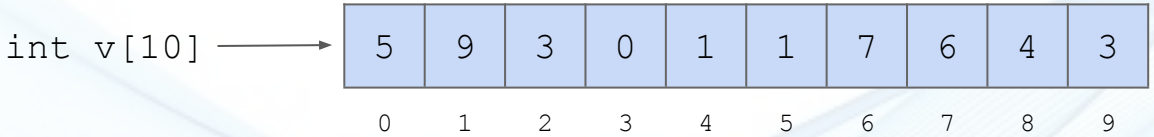
Aula 09

Marcos Silvano Almeida
marcossilvano@professores.utfpr.edu.br
Departamento de Computação
UTFPR Campo Mourão

BCC31A

Vetores / Arrays

- Um vetor ou array, é uma estrutura de dados composta que permite armazenar uma sequência de tamanho fixo de valores do mesmo tipo
  - É o equivalente a termos uma sequência de variáveis do mesmo tipo



- O tamanho (quantidade de elementos) do vetor é definido em sua declaração.
  - `tipo nome_variável[tamanho];`

- Em um vetor **v** de tamanho **10**:
  - Primeiro elemento **v**[0]
  - Último elemento **v**[9]
  - ❌ Inválido v[-4] ⇐ Índices negativos
  - ❌ Inválido v[12] ⇐ Acessar elementos foram do intervalo

Vetores: declarando e acessando

```
int v1[6];                          << tamanho (fixo para toda execução)
                                       (posições contêm lixo de memória)

int v2[] = {2,4,6,8,10};             << inicialização
v2[3] = 55;                          << {2,4,6,55,10}
int a = v2[1];                       << {2,4,6,55,10}
printf("3a posicao:%d", v2[2]);      << {2,4,6,55,10}

int v3[5]= {5,6,7,8,9};           << tamanho + inicialização

int n = 10;
int v4[n];                        << tamanho varia pela execução do programa
                                     (posições contêm lixo de memória)


❌ v2 = {10,20,30,40,50};         << não é possível atribuição de vetor,
❌ v1 = v2;                          somente de seus elementos
```

Trabalhando com vetores

- Para trabalhar com vetores, os comandos de laço são essenciais

```c
#include <stdio.h>
int main() {
   printf("\nIMPRIME OS ELEMENTOS DE UM VETOR\n\n");

   int v[] = {1,2,3,4,5,6,7,8,9,10};     // vetor de tamanho 10

   for (int i = 0; i < 10; i++) {        // i vai de 0 à 9
       printf(" %d", v[i]);
   }
   printf("\n");

   return 0;
}
```

Trabalhando com vetores

```c
#include <stdio.h>
#include <stdlib.h>
int main() {
   int n;
   printf("\nVETOR DE VALORES ALEATORIOS ");
   printf("Tamanho > ");
   scanf("%d", &n);
   int v[n];                          // atribui valores aleatórios às posições do
vetor
   for (int i = 0; i < n; i++) { // i vai de 0 à n-1
       v[i] = rand() % 10 + 1;    // rand() sorteia valores de 1 à 10
   }
   for (int i = 0; i < n; i++) { // i vai de 0 à n-1
       printf(" %d", v[i]);
   }
   printf("\n\n");
   return 0;
```

Passando vetores para funções

- Quando passados para funções, vetores sempre são “referências”
  - O parâmetro aponta para o vetor passado à função
  - Qualquer alteração no vetor, dentro da função, refletirá no vetor passado

```
#include <stdio.h>

void vectorSet(int n, int v[]) { // necessário passar o tamanho do vetor
    for (int i = 0; i < n; i++) { // i vai de 0 à n-1
        v[i] = i+1;               // atribui valores de 1 à n
    }
}
int main() {
    int vec[10];
    vectorSet(10, vec);           // {1,2,3,4,5,6,7,8,9,10}
    return 0;
}
```

Passando vetores para funções

```c
void vectorSet(int n, int v[]) {
    for (int i = 0; i < n; i++) { // i vai de 0 à n-1
        v[i] = i+1;               // atribui valores de 1 à n
    }
}
void vectorPrint(int n, int v[]) {
    for (int i = 0; i < n; i++) { // i vai de 0 à n-1
        printf(" %d", v[i]);
    }
    printf("\n\n");
}
int main() {
    int vec[10];
    vectorSet(10, vec);
    vectorPrint(10, vec);
    return 0;
}
```

Referências

- Algoritmos e Programação
  - Marcela Gonçalves dos Santos
  - Disponível pelo Moodle

- Estruturas de Dados, Waldemar Celes e José Lucas Rangel
  - PUC-RIO - Curso de Engenharia
  - Disponível pelo Moodle

- Linguagem C, Silvio do Lago Pereira
  - USP - Instituto de Matemática e Estatística
  - Disponível pelo Moodle

- Curso Interativo da Linguagem C
  - https://www.tutorialspoint.com/cprogramming