



(Estrutura de Dados)

Aula 3

Prof. Dr. Eliseu LS



(Linguagem C ++)
Vetores & Sub rotinas

Prof. Dr. Eliseu LS

Conteúdo da Aula

Revisão de vetores e programação

- Prática de Vetores explícitos e não explícitos
- Criação de sub rotinas
- Leitura de strings com sub rotinas
- Estrutura de Menu de Controle

Sub rotina void maiorMenor ()

Você pode também declarar vetores em forma de lista, neste caso os vetores serão alimentados no ato da declaração do vetor. Por exemplo: `int a [] = {1,2,3,4}`, `string b [] = {"a","b","c","d"}`, `char sex [] = {'F','M','F'}`, `bool x [] = {true, false, true, false}`.

Considere o vetor a seguir:

```
int xy [ ] = {1, 5, 10, -1};
```

Vamos fazer um código para retornar o maior valor e outro para retornar o menor valor desta lista.

Atenção:

- 1) Indente/Organize o Código;
- 2) Coloque um menu;
- 3) Compile e execute no IDE.

PROGRAMA 1

```
#include "iostream"
#include "cstdlib"
using namespace std;

int xy [ ] = { 1, 5, 10, -1 };

void maiorMenor ( ) {
    int maior, menor;
    int tot = sizeof(xy)/sizeof(int);

    for (int i = 0; i < tot; i++) {
        if ( xy[i] > maior || i == 0 ) maior = xy[i];
        if ( xy[i] < menor || i == 0 ) menor = xy[i]; }

    cout << "\nMaior:" << maior;
    cout << "\nMenor" << menor << endl;
    system("pause");
}

int main ( ) { maiorMenor ( ); return 0; }
```

Função MediaVetor ()

Você pode também realizar operações lógicas ou aritméticas com vetores, Considere o vetor a seguir:

```
int vet [ ] = { 1, 3, 4, -1 };
```

Vamos fazer um código para retornar a média dos valores desta lista.

Atenção:

- 1) Indente/Organize o Código;
- 2) Coloque um menu;
- 3) Compile e execute no IDE.

PROGRAMA 2

```
#include "iostream"
```

```
#include "cstdlib"
```

```
using namespace std;
```

```
int vet [ ] = {1, 3, 4, -1 };
```

```
float MediaVetor ( ) { float soma = 0;
```

```
    int n = sizeof(vet)/sizeof(int);
```

```
    for (int i = 0; i<n; i++)  
        { soma += vet[i]; }
```

```
    media = soma / n ;  
    return media ; }
```

```
int main () { float vlmedia = MediaVetor();  
cout << vlmedia << endl;  
system("pause"); }
```

Uma tabela de vetores

PROGRAMA 3: Nesta simulação vamos usar o void guardar() para inserir dados lidos por funções através do teclado, uma linha por vez. Monte o código fonte, compile e execute em um IDE de sua preferência.

1) No início do código, logo após as diretivas do pré-processador, definiremos a variável global **linha** que será incrementada a cada leitura de um novo registro bem como os três vetores que utilizaremos para ler uma lista de nomes, idades e salários.

```
int linha = -1;  
string nome [ 4 ]; int idade [4]; double salario [4];
```

EXEMPLO DE TABELA A SER GERADA PELOS VETORES

linha	Nome	Ida	Salário
0	Francilino	25	2500
1	Benedicto	40	3000
2	Ruanito	19	2500
3	Mariano	30	5000

Funções de leitura

2) Logo após a definição dos vetores não explícitos, iremos definir primeiramente as funções de leitura, para fazer a leitura dos dados via teclado, a saber, o nome, a idade e finalmente o salário dos usuários.

```
string lerNome() { string nome;  
    cout << "\nNome:";  
    cin.ignore( ); // necessário para ignorar o ENTER do cin anterior  
    getline(cin, nome);  
    return nome; }
```

```
int lerIdade() { int idade;  
    cout << "\nIdade:";  
    cin >> idade;  
    return idade; }
```

```
double lerSalario() { double sal;  
    cout << "\nSalário:";  
    cin >> sal;  
    return sal; }
```

Criando uma tabela com vários vetores

3) O void **guardarDados** (**nom**, **ida**, **sal**) será o código responsável que irá guardar cada nome, idade e salário digitado pelo usuário e o **void listar** () irá imprimir as linhas que já foram armazenadas nos vetores de armazenamento.

```
void guardarDados ( string nom, int ida, double sal ) {  
  
    linha ++;  
    nome[linha] = nom ;  
    idade[linha] = ida;  
    salario[linha]=sal;    }  
  
void listar ( )    {  
  
    for ( int i = 0; i<=linha; i++) {  
        cout << nome[i] << " - " << idade[i]  << " - " <<  salario [i] << endl ;  
    }  
  
    system("pause");    }
```


Menu que irá controlar o programa

```
int main()
{
    string nome_; int idade_; double salario_;

    int tecla;

    MENU:
        system("clear");
        cout << "* Menu *\n1 ler\n2 exibir\n3
sair\nitem:";
        cin >> tecla;
```

```
switch ( tecla )
{
    case 1: nome_ = lerNome();
            idade_ = lerIdade();
            salario_ = lerSalario();
            guardarDados ( nome_, idade_,
salario_);
            break;

    case 2: listar ();
            break;

    case 3: exit(0); break;
}

goto MENU;
return 0; }
```

Tarefa / Avaliação Contínua

I- Faça o algoritmo para calcular o valor de uma prestação em atraso com base na multa de 2% e também do juros mensal de 1% sobre o valor principal. Declare vetores não explícitos para entrada do valor da dívida, Divida [5] e da quantidade de dias em atraso: Dias [5]. Crie os vetores de saída, ValorMulta[5], ValorJuros[5] e ValorPagar [5]. Ao final imprima todos os vetores de saída. Faça um menu de controle para controlar o programa.

II - Crie um algoritmo que tenha um vetor de idades e um vetor de nomes, cujos dados deverão ser lidos pelo teclado. Ao final Exiba o nome da pessoa de maior idade, menor idade, a média das idades e o saldo das idades.

NOTA: Entrega obrigatória individual juntamente com o enunciado e o print da tela de execução do programa em um único arquivo no formato DOCX ou PDF pelo MS-Teams.