Módulo 6

Matrizes e Strings

Linguagem da Computação

(Rone Ilídio)

Matrizes

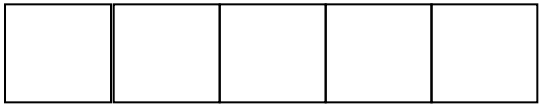
- É um conjunto de variáveis contínuas na memória, do mesmo tipo, referenciado por um único nome,onde cada variável é diferenciada por meio de um número denominado índice
- A primeira posição sempre têm índice zero
- São também chamadas de array (Pascal)

Matrizes

- Declaração

int m[5];

Essa declaração cria cinco variáveis contínuas na memória, todas elas do tipo inteiro, referenciadas pelo identificador “m”

Matriz m → 

Índices → 0 1 2 3 4

Matrizes

- A matriz “m” aloca 5 posições de memória, onde a primeira é referenciada pelo índice 0, a segunda pelo índice 1 e assim sucessivamente
- Conseqüentemente, a última posição da matriz é o tamanho do vetor menos 1 (no caso 5-1=4)
- O índice de uma matriz é sempre um número inteiro ou uma expressão inteira, exemplo:

```
int a=1, b= 2; // Variáveis inteiras
m[a+b] = 15; // Índice representado por uma
                    // expressão inteira.
```

Matrizes

- Os valores contidos em uma matriz devem ser referenciando um a um, exemplo:

```
int v[5]; // Declaração da matriz
v[0] = 14;// Atribuição do valor 14 à primeira posição da matriz
v[1] = 8;// Atribuição do valor 8 à segunda posição da matriz
cout << “A primeira posição tem valor” << v[0];
cout << “A segunda posição tem valor” << v[1];
```

```
//Preenche uma matriz com valores fornecidos pelo usuario e imprime
//esses valores
#include <iostream>
#include <conio.h>
using namespace std;
int main(){
    int mat[5];
    int i,u;
    for(i=0; i<=4; i++){
        cout <<"\nDigite um valor inteiro: ";
        cin >> mat[i];
    }
    for(u=0; u<=4; u++){
        cout << "\nMatriz na posicao"<< u <<  mat[u];
    }
    cout << "\n"
    system("pause");
}
```

Matrizes

- As matrizes pode ser de qualquer tipo de dados, primitivo ou não
- A seguir temos o exemplo anterior modificado para receber caracteres. O próximo é um programa que receber valores do tipo float (notas de alunos) insere em um vetor e retorna a maior e a menor nota

```
//Programa que recebe 10 notas, coloca cada uma em uma posição de uma
//matriz com 10 posições e retorna a maior nota
#include <iostream>
#include <conio.h>
using namespace std;
int main(){
     int i, u;
    float notas[10];
    for(i=0; i<10; i++){
    cout<<“\nInforme uma nota: ";
            cin>> notas[i];
    }
    float maior;
    maior = notas[0];
    for(u=1; u<10; u++){
            if (notas[u]>maior) maior = notas[u];
    }
    cout << "\nA maior nota e:"<< maior << "\n";
    system("pause");
}
```

Matrizes

- Para inicializar um matrizes já com valores, ou seja, criar e atribuir valores junto com a criação, utiliza-se a seguinte sintaxe:

int m = {5, 4, 3, 2, 1}

- Esse código gera uma matriz “m”, de 5 posições, onde a primeira posição (índice 0) terá valor 5, a segunda (índice 1) valor 4 e assim sucessivamente.

```
//Preenche uma matriz com valores fornecidos pelo programador e pede para que o
    usuario tente adivinha um desses valores
#include <iostream>
#include <conio.h>
using namespace std;
int main(){
    int mat[]={12,34,53,52,1};
    int num, aux=0,i;
    cout << "\nTente adivinha um dos numeros da matriz: ";
    cin>> num;
   system("cls");
    for (i=0; i<=4; i++){
            if (mat[i]==num){
                    aux = 1;
                    break;
            }
    }
    if (aux==0) cout << "\nEsse numero nao consta na matriz.";
    else cout << "\nVoce acertou!";
   getch();
}
```

Ordenação de Matrizes

- Utilizaremos o método mais simples para ordenação, denominado seleção.
- Ele varre o vetor com dois for e vai inserindo os elementos de valores menores nas primeiras posições
- Veja exemplo:

```
#include <iostream>
#include <conio.h>
using namespace std;
int main(){
      int aux,i,u, m[]={12,34,53,52,1,32,4,95,32,54};
      cout << "Antes: ";
      for (i=0; i<=9; i++){
            cout << " "<< m[i];
      }
      for (i=0; i<=8; i++){
            for (u=i+1; u<=9; u++){
                  if (m[i]>m[u]){
                        aux = m[i];
                        m[i] = m[u];
                        m[u] = aux;
                  }
            }
      }
     cout << "\nDepois: ";
      for (i=0; i<=9; i++){
            cout << " " << m[i];
      }
   cout << "\n";  system("pause");
}
```

```
#include <iostream>
#include <conio.h>
using namespace std;
int main(){
      int aux,i,u;
      char m[]={'d','n','f','q','z','s','b','s','o','a'};
    cout << "Antes: ";
      for (i=0; i<=9; i++){
              cout << m[i] << "  ";
      }
      for (i=0; i<=8; i++){
              for (u=i+1; u<=9; u++){
                      if (m[i]>m[u]){
                              aux = m[i];
                              m[i] = m[u];
                              m[u] = aux;
                      }
              }
      }
    cout << "\nDepois: ";
      for (i=0; i<=9; i++){
              cout <<  m[i] << " ";
      }
      getch();
}
```

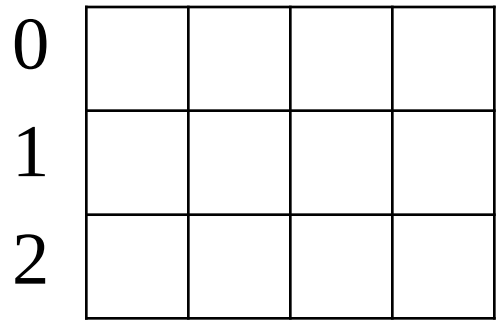
Matrizes Multidimensionais

- Até agora vimos matrizes formadas por elementos dispostos um do lado do outro, formando uma única linha (uma única dimensão). Esse tipo de matriz é também denominado de “vetor”
- Veremos que matrizes podem possuir mais de uma linha de valores, ou seja, possuir mais de uma dimensão

Matrizes Multidimensionais

- Primeiramente veremos matrizes de duas dimensões (linhas e colunas)
- O código a seguir cria uma matriz 3x4 (lê-se três por quatro), com três linhas e quatro colunas:

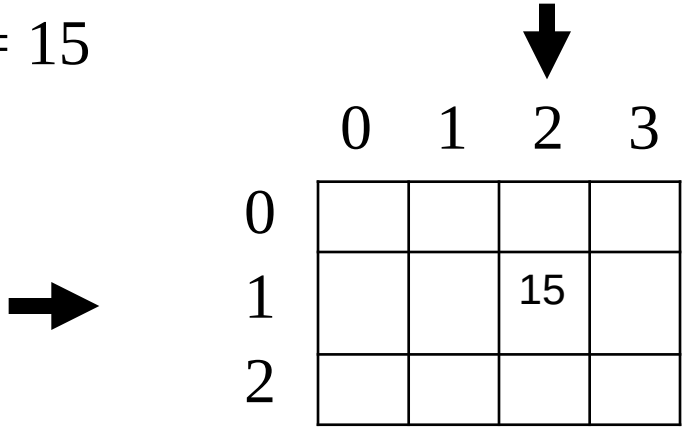
float mult[3][4];



Matrizes Multidimensionais

- Para a manipulação dos valores de matrizes multidimensionais utiliza-se dois índices. Um fala qual será a linha acessada e outro qual será a coluna.
- Ex: para atribuir valor ao elemento que está na linha com índice igual a 1 e coluna com índice igual 2 usa-se:

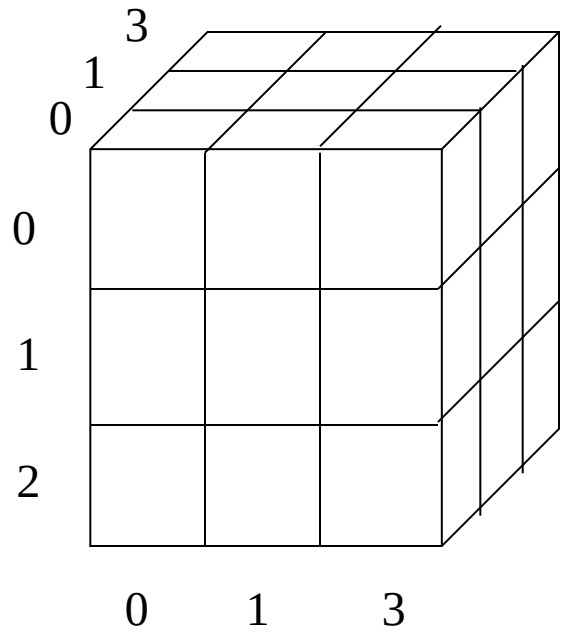
  mult[1][2] = 15



```
//Entrando com valores do usuário em uma matriz 3x4 e depois imprimindo tais valores
#include <iostream>
#include <conio.h>
using namespace std;
int main(){
    int mat[3][4];
    int i,u;
     for(i=0; i<=2; i++){
            for (u=0; u<=3; u++){
                    cout << "\nDigite um numero: ";
                    cin >> mat[i][u];
            }
     }
     for(i=0; i<=2; i++){
            for (u=0; u<=3; u++){
                    cout << mat[i][u] << "  " ;
            }
            cout << "\n";
     }
     getch();
}
```

Matrizes Multidimensionais

- Matrizes podem ser quantas dimensões forem necessários
- Aqui temos a declaração de uma matriz de 3 dimensões:



```
//Entrando com valores do usuario em uma matriz 2x2x2 (3 dimensoes)
#include <iostream>
#include <conio.h>
using namespace std;
int main(){
    int mat[2][2][2];
    int a,b,c;
    for(a=0; a<=1; a++){
        for (b=0; b<=1; b++){
            for (c=0; c<=1; c++){
                cout <<"\nDigite um numero ";
                cin >> mat[a][b][c];
            }
        }
    }
    getch();
}
```

Matrizes Multidimensionais

- Para inicializar matrizes multidimensionais deve-se faze como no exemplo abaixo:
  - int m2[2][2] = {{0,0},{0,0}}
  - int m3[2][2] = {{{0,0},{0,0}},{{0,0},{0,0}}}
  - E assim sucessivamente.

O Tipo string

- Utilizado para receber sequência de caracteres
- Para pegar do usuário uma palavra utiliza-se o cin
- Para pegar do usuário uma frase utiliza-se
  - getline(cin,variável);
- Obs: cin.ignore() apaga o buffer de teclado
- Veja exemplo

```
#include <iostream>
#include <conio.h>
using namespace std;
int main(){
	string p, t;
	cout << "Digite uma palavra: ";
	cin>>p;
	cin.ignore();
	cout << "Digite uma frase: ";
	getline(cin,t);
	cout << "Palavra digitada -- > "<< p;
	cout << "\nTexto digitado --> " << t;
	getch();
}
```

Acessando cada um dos caracteres de uma string

```
#include <iostream>
#include <conio.h>
using namespace std;
int main(){
	string texto;
	cout << "Digite uma frase: ";
	getline(cin,texto);
	for(int i=0; i<texto.size();i++){
	    cout << texto[i] << endl;
	}
	getch();
}
```

Strings com vetores de caracteres

- Declaração:

  char v[30];

- Obs: a uso desse tipo de string requer funções especiais para sua manipulação.
- Exemplo:

```
#include <iostream>
#include <conio.h>
using namespace std;
int main(){
	char v[30];
	cout << "Digite uma palavra: ";
	cin>>v;

	cout << "\nVoce digitou: " << v;
	getch();
}
```

Inicializando strings

- Duas formas: por atribuição e como parâmetro

```
#include <iostream>
#include <conio.h>
using namespace std;
int main(){
    string n="por atribuicao";
    string p("como parametro");
    cout << "Primeira forma de inicializacao -- > "<< n;
    cout << "\nSegunda forma de inicialização --> " << p;
    getch();
}
```

Concatenação de string

- Concatenação é união de string

```
#include <iostream>
#include <conio.h>
using namespace std;
int main(){
    string nome, sobrenome, completo;
    cout << "Digite se nome:";
    cin >> nome;
    cout << "Digite seu sobrenome:";
    cin >> sobrenome;
    completo = nome + " " + sobrenome;
    cout << endl << "Nome completo: " << completo;
    getch();
}
```

Comparação de string

- O tipo string utiliza todos os operadores de comparação: ==, !=, <, >, <= e >=.
- Os operadores <, >, <= e >= são utilizados para fazer ordenação em ordem alfabética.
  - Abacate < Zumbi

```
#include <iostream>
#include <conio.h>
using namespace std;
int main(){
    string n1, n2, ordem;
    cout << "Digite se nome:";
    cin >> n1;
    cout << "Digite outro nome:";
    cin >> n2;
    if(n1 <= n2)
        ordem = n1 + ", " + n2;
    else
        ordem = n2 + ", " + n1;
    cout << endl << "Em ordem alfabetica: " << ordem;
    getch();
}
```

Ordenação de strings

- Utilizaremos o método Seleção
- Veja exemplo

```
#include <iostream>
#include <conio.h>
using namespace std;
int main(){
    int i,u, t=4;
    string nomes[t];
   cout << "Preenchimento\n\n ";
    for (i=0; i<t; i++){
            cout << "\nDigite um nome  ";
            cin >> nomes[i];
    }
    for (i=0; i<=t-2; i++){
            for (u=i+1; u<t; u++){
                    if (nomes[i]>nomes[u]){
                            string aux;
                            aux = nomes[i];
                            nomes[i] = nomes[u];
                            nomes[u] = aux;
                    }
            }
    }
   cout << "\nDepois: ";
    for (i=0; i<t; i++){
            cout <<  nomes[i] << " ";
    }
    getch();
}
```