Universidade de Brasília

Vetores

Computação para Engenharia — Tópico 6 — Parte 1

Daniel Guerreiro e Silva

Departamento de Engenharia Elétrica (ENE), Faculdade de Tecnologia (FT)

Roteiro

Introdução

Como armazenar 3 notas?

```
float nota1, nota2, nota3;

cout << "Nota do aluno 1: ";
cin >> nota1;
cout << "Nota do aluno 2: ";
cin >> nota2;
cout << "Nota do aluno 3: ";
cin >> nota3;
```

Como armazenar 100 notas?

```
float nota1, nota2, nota3, /* .... */ nota100;

cout << "Nota do aluno 1: ";
cin >> nota1;
cout << "Nota do aluno 2: ";
cin >> nota2;

/* ... */

cout << "Nota do aluno 100: ";
cin >> nota100;
```

Definição

Vetores

Definição
Vetores (Arrays): Coleção de variáveis do mesmo tipo referenciada por um nome comum.

(Herbert Schildt - C: The Complete Reference)

- acesso por meio de índice
- posições contíguas na memória
- tamanho pré-definido
- índices fora dos limites podem causar comportamento anômalo do código

Como declarar um vetor

<tipo> identificador [<número de posições>];

- A primeira posição de um vetor sempre tem índice 0 (zero).
- A última posição de um vetor tem índice *<número de posições>* - 1.

Exemplo

```
float notas[100];
```

Repare que `notas` equivale a termos 100 variáveis do tipo `float`.

Usando um vetor

Atribuição
variavel = vetor[<posição>];

- Pode-se substituir uma variável de um determinado tipo por um único elemento de um determinado vetor.
- Este elemento se comporta como uma variável: retorna o seu valor como uma expressão e pode ter valores atribuídos.

Exemplo

```
a = nota[10];

nota[5] = 9.5;
```

Vetores - Organização na memória

```
int d;
int vetor[5];
int f;
```

Nome	d	vetor					f
Índice	-	0	1	2	3	4	-
Valor	?	?	?	?	?	?	?

Vetores - Organização na memória

Ao executar `vetor[3]=10;`

Nome	d	vetor					f
Índice	-	0	1	2	3	4	-
Valor	?	?	?	?	10	?	?

Vetores - Organização na memória

O que ocorre ao se executar os comandos:

```
vetor[5]=5;
vetor[-1]=1;
```

?

Vetores - Organização na memória

Ao executar

```
vetor[3]=10;
vetor[5]=5;
vetor[-1]=1;
```

Nome	d	vetor					f
Índice	-	0	1	2	3	4	-
Valor	1	?	?	?	10	?	5

Questões importantes sobre vetores

O tamanho do vetor é pré-definido (ou seja, após a compilação o tamanho não pode ser mudado).

- Índices fora dos limites podem causar comportamento anômalo do código.

Exemplo
Veja o código limites.cpp

Exemplos

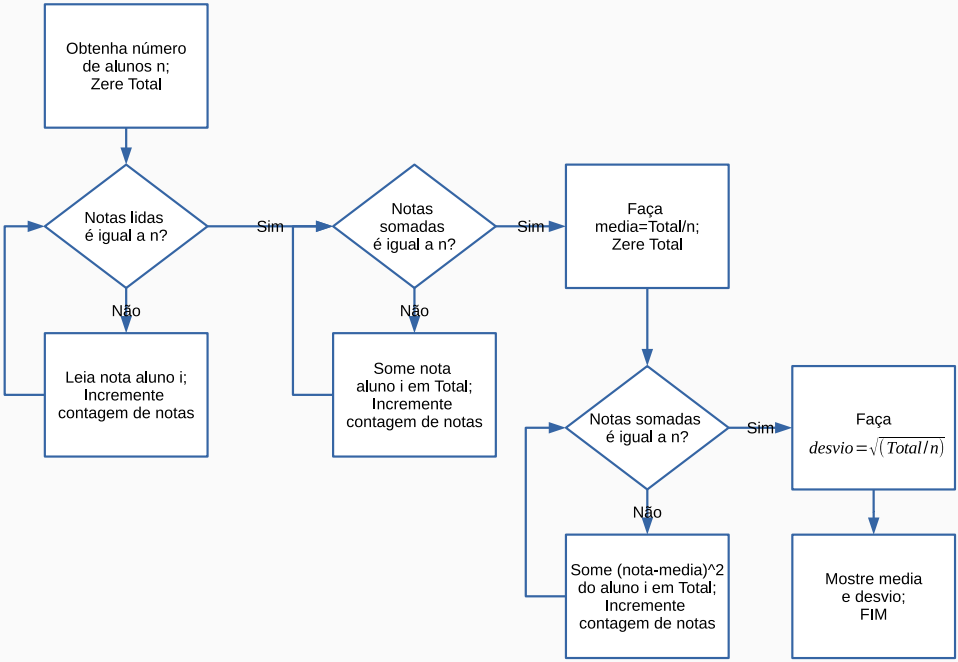
Como armazenar n ($<=$ 100) notas?

Crie um algoritmo que leia uma lista de notas (máximo 100), calcule e exiba a média e o seu desvio-padrão.

$$m = \frac{1}{n} \sum_{i=1}^{n} x_i$$

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (x_i - m)^2}$$

Como armazenar n (<= 100) notas?



Como armazenar n ($<=$ 100) notas?

```cpp
#include <iostream>
#include <cmath>
#include <iomanip>
using namespace std;

int main() {
  float nota[100], media=0,desvpad=0;
  int n, i;

  cout << "Numero de alunos: ";
  cin >> n;
  for (i = 0; i < n; i++) { //leitura das notas
    cout << "Nota do aluno " << i+1 << ": ";
    cin >> nota[i];
  }

  for (i = 0; i < n; i++) //calculo da media
    media = media + nota[i];
  media = media / n;

  for (i = 0; i < n; i++)//calculo do desvio padrao
    desvpad = desvpad + (nota[i]-media)*(nota[i]-media);
  desvpad = sqrt(desvpad/n);

  cout  << "Nota media = " << setprecision(1) << fixed
        << media << "\nDesvio padrao = " << desvpad     << endl;
  return 0;
}
```

Polinômios

Leia os coeficientes $a_0, a_1, ..., a_k$, onde $k \leq 25$, e exiba a expressão completa do referido polinômio de ordem k

$$a_k x^k + a_{k-1} x^{k-1} + \cdots + a_1 x + a_0$$

Polinômios

```cpp
#include <iostream>
#include <iomanip>
using namespace std;

int main() {
        float coef[26];
        int grau, i;

        cout << "Grau do polinomio (grau maximo = 25): ";
        cin >> grau;
        while (grau < 0 || grau > 25) {//digitou fora do limite, repete a leitura
                cout << "Grau invalido, digite outra vez: ";
                cin >> grau;
        }
        for (i = grau; i >= 0; i--) { //leitura dos coeficientes
                cout << "coeficiente de x^" << i << ": ";
                cin >> coef[i];
        }
        cout << setprecision(1) << fixed << coef[grau] << "x^" << grau;
        for (i = grau - 1; i >= 0; i--) { //impressao do polinomio
                if (coef[i] != 0)
                        if (coef[i] >= 0)
                                cout << " + " << coef[i] << "x^" << i;
                        else
                                cout << " - " << -coef[i] << "x^" << i;
        }
        cout << endl;
        return 0;
}
```

Busca em vetores

Um problema comum quando se manipula vetores é encontrar um elemento com um determinado valor.

- A forma direta de se resolver é percorrer da posição inicial até a final todos os elementos do vetor, até achar o valor desejado $\longrightarrow$ Busca Linear ou Sequencial.

Busca sequencial por um valor

```cpp
//Procura um elemento em uma lista de valores - busca linear.
#include <iostream>

using namespace std;

int main() {
  int lista[10], chave;
  int i;

  cout << "Digite uma lista de 10 numeros inteiros: ";
  for(i=0; i<10; i++)
    cin >> lista[i];

  cout << "Digite um numero a procurar na lista: ";
  cin >> chave;

  //laco de busca - veja que e somente 1 linha de comando
  for (i = 0; lista[i] != chave && i<10; i++);

  if (i<10) //se laco for parou antes do 10, significa que encontrou chave
    cout << "O elemento " << chave << " esta presente na lista\n";
  else //caso contrario, nao encontrou a chave
    cout << "O elemento " << chave << " nao esta presente na lista\n";

  return 0;
}
```

Busca sequencial por um valor - outra implementação

```cpp
//Procura um elemento em uma lista de valores - busca linear.
#include <iostream>

using namespace std;
int main() {
  int lista[10], chave;
  int i;
  bool achou;

  cout << "Digite uma lista de 10 numeros inteiros: ";
  for(i=0; i<10; i++)
    cin >> lista[i];

  cout << "Digite um numero a procurar na lista: ";
  cin >> chave;

  achou = false;
  //laco de busca
  for (i = 0; !achou && i<10; i++){
        if(lista[i]==chave)
                achou = true;
  }

  if (achou)
        cout << "O elemento " << chave << " esta presente na lista\n";
  else
        cout << "O elemento " << chave << " nao esta presente na lista\n";

  return 0;
}
```

Matrizes

Utilizando vetores, criamos um programa que lê as notas de uma prova para um conjunto de alunos e então calcula a média da turma.

Agora queremos ler as notas de 4 provas para cada aluno e então calcular a média do aluno e a média da classe. O tamanho máximo da turma é de 50 alunos.

Solução
Criar 4 vetores cada um com 50 posições. E então ler as respectivas informações.

```
float nota0[50],nota1[50],nota2[50],nota3[50];
```

Matrizes

Agora, suponha que estamos trabalhando com no máximo 100 provas e 100 alunos. Seria muito cansativo criar 100 vetores e atribuir 100 nomes diferentes (parece que esse problema não tem fim !!!).

- Para resolver, podemos utilizar matrizes. Uma matriz é um vetor (ou seja, um conjunto de variáveis de mesmo tipo) que possui duas ou mais dimensões.

Declarando uma matriz

`<tipo> nome_da_matriz [<linhas>]; [<colunas>]`

- Uma matriz possui *linhas* $\times$ *colunas* variáveis do tipo `<tipo>`
- As linhas são numeradas de 0 a *linhas* $-$ 1
- As colunas são numeradas de 0 a *colunas* $-$ 1

Exemplo de declaração de matriz

```
int matriz[4][4];
```

	0	1	2	3
0				
1				
2				
3				

Declarando uma matriz de múltiplas dimensões

$<\texttt{tipo}>\ \texttt{nome\_da\_matriz}\ [<dim_1>]\,[<dim_2>]\ldots[<dim_N>]$

- Essa matriz possui $dim_1 \times dim_2 \times \cdots \times \dim_N$ variáveis do tipo $<\texttt{tipo}>$
- Cada dimensão é numerada de 0 a $dim_i - 1$

Acessando uma matriz

Em qualquer lugar onde você escreveria uma variável no seu programa, você pode usar um elemento de sua matriz, da seguinte forma:

`nome_da_matriz [<linha>] [<coluna>]`

Ex.: `matriz [1][10]` — Refere-se a variável na 2^a linha e na 11^a coluna da matriz.

- Lembre-se sempre: o compilador não verifica se você utilizou valores válidos para a linha e para a coluna.

Exemplo: lendo e escrevendo uma matriz

```cpp
#include <iostream>
using namespace std;

int main () {
  int matriz[4][4];
  int i, j;

  /*Leitura*/
  for (i = 0; i < 4; i++) //para todas as linhas
    for (j = 0; j < 4; j++) { //para todas as colunas de cada linha
      cout << "Matriz[" << i << "][" << j << "]: ";
      cin >> matriz[i][j];
    }

  /*Escrita*/
  for (i = 0; i < 4; i++) { //cada linha
    for (j = 0; j < 4; j++) //cada coluna de cada linha
      cout << matriz[i][j] << "\t";
    cout << endl;
  }
  return 0;
}
```

Exemplo: matriz e sua transposta

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main(){
6    int mat[3][3];
7    int i = 0,j = 0,k = 0;
8
9    for (i = 0; i < 3; i++)  //leitura dos elementos da matriz
10     for (j = 0; j < 3; j++){
11       cout << "Digite o valor da posicao (" << i << ", " << j << "): ";
12       cin >> mat[i][j];
13     }
14
15   cout << "Matriz\n";
16   for (i = 0; i < 3; i++){ //exibicao da matriz original
17     for (j = 0; j < 3; j++)
18       cout << mat[i][j] << "\t";
19     cout << "\n";
20   }
21
22   cout << "Transposta\n";
23   for (i = 0; i < 3; i++){ //exibicao da matriz transposta
24     for (j = 0; j < 3; j++)
25       cout << mat[j][i] << "\t";
26     cout << "\n";
27   }
28   return 0;
29 }
```

Até o próximo tópico...