Aula 14 – Estruturas de Dados - Vetores

Algoritmos e Programação de Computadores

Profs: Ronaldo Castro de Oliveira – ronaldo.co@ufu.br

Anilton Joaquim da Silva – anilton@ufu.br

Introdução

- Imagine o seguinte problema:

  Ler um conjunto de N notas do tipo float, calcular sua média e imprimir a quantidade de notas acima da média.
- Com o que sabemos até aqui isto não é possível porque quando lemos as notas as mesmas não ficam guardadas.
- Se fossem somente 5 notas uma saída seria definir cinco variáveis diferentes (N1, N2...N5).
- De qualquer forma este programa seria limitado a somente 5 notas, é um programa difícil de manter, será necessário um código grande para um problema simples.....
- A solução para esta situação é o uso de vetores (ou variáveis indexadas, ou arranjo, ou array)

Vetores

- Um vetor é uma variável capaz de armazenar vários dados de mesmo tipo, com o mesmo identificador (mesmo nome) e alocadas seqüencialmente na memória. Uma vez que as variáveis têm o mesmo nome, o que as distingue é um índice, que referencia sua localização dentro da estrutura.
- Declarção de um Vetor

  Tipo_de_dado nome_vetor [tamanho_vetor];
- Ex: int vetor1 [4];

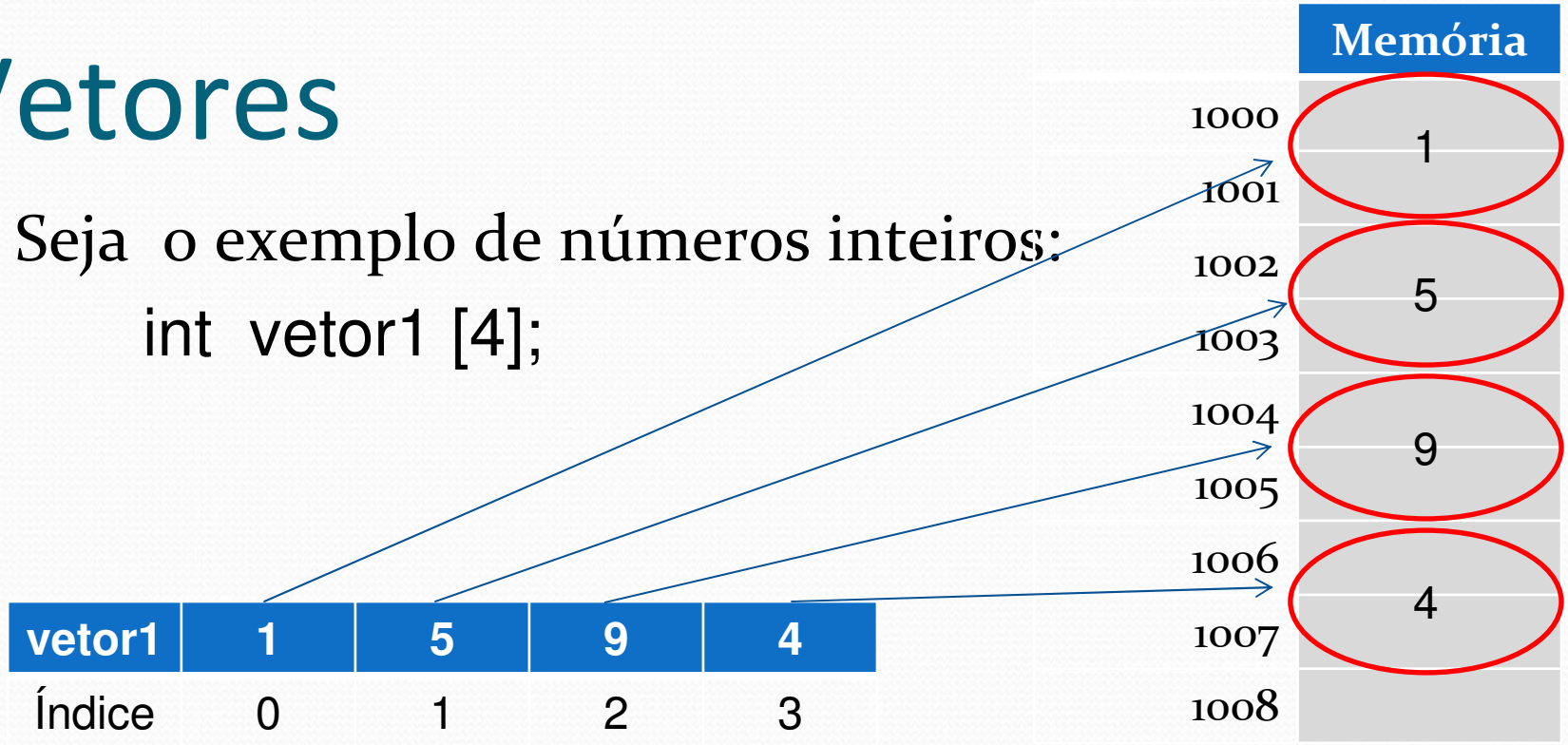
  float notas[100];

  char nome [30];

Vetores

- Seja o exemplo de números inteiros:

  int vetor1 [4];



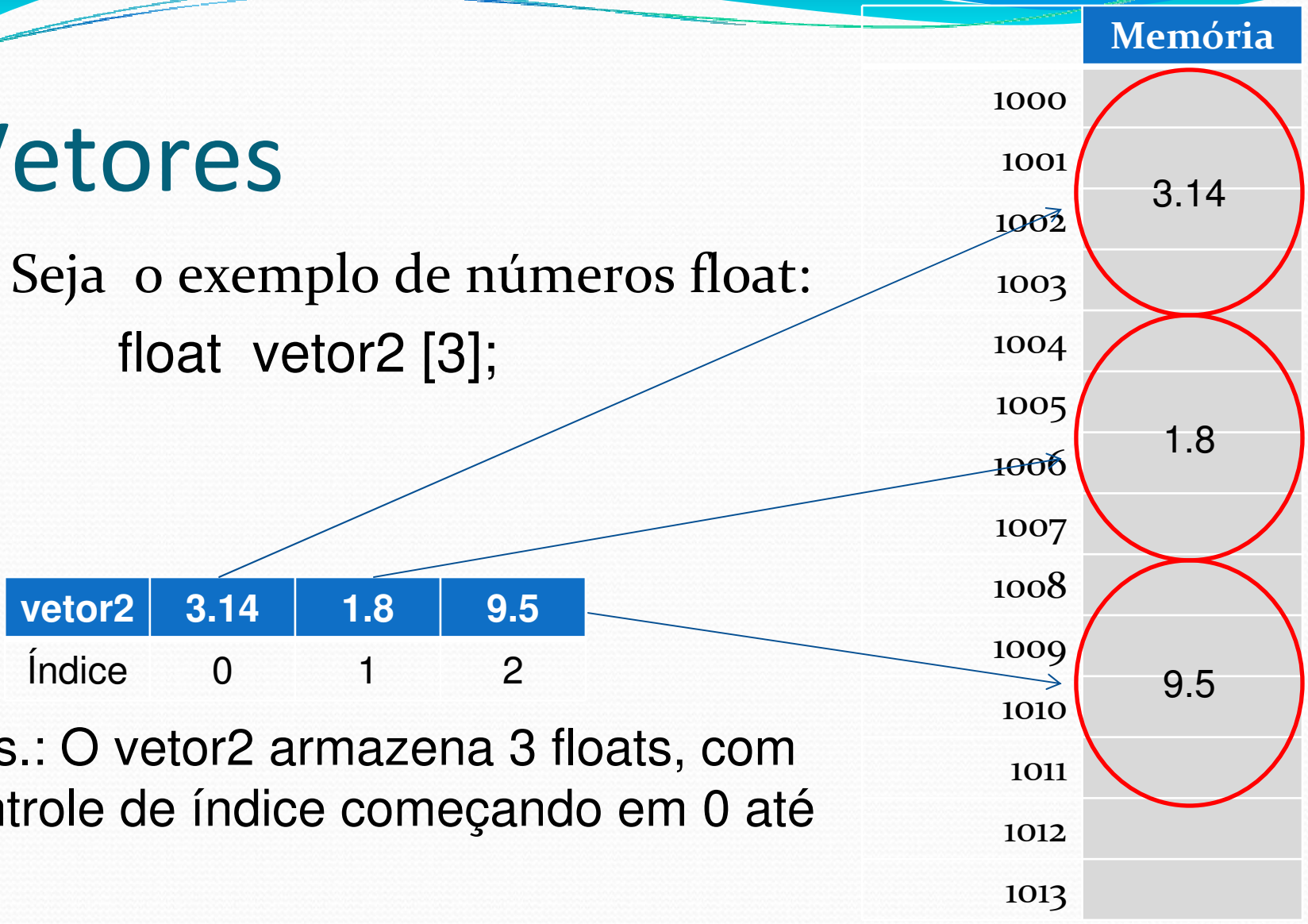
Obs.: o índice do vetor, em C++, sempre começa em 0. Logo o vetor1 pode armazenar 4 inteiros, com controle de índice começando em 0 até 3.

Vetores

- Seja o exemplo de números float:

  float vetor2 [3];

vetor2	3.14	1.8	9.5
Índice	0	1	2



Obs.: O vetor2 armazena 3 floats, com controle de índice começando em 0 até 2.

Vetores

- Seja o exemplo de números inteiros:

  int vetor1 [4];

vetor	1	5	9	4	?
Índice	0	1	2	3	

- Posição:
  - vetor1[2]→ 9
  - vetor1[1]→ 5
  - vetor1[4]→ ? ---> ERRO

Atribuindo valores a um vetor

- Para atribuir diretamente um valor a uma determinada posição do vetor, pode-se usar a seguinte expressão:

nome_vetor[posição] = valor;

Ex.: int vetor1 [6];

vetor1[5]=2;

vetor1[1]=1;

Lendo um vetor

- Para ler um vetor e ter seus valores alocados em posições específicas, pode-se utilizar o comando **for**.

```
void  le_vetor (int vetor [], int N)
{
   for (int i = 0; i < N; i++)
   {
            cout << "Digite o elemento " << i << "do vetor:";
            cin >> vetor [i];
   }
}
```

OBS1: o procedimento le_vetor recebe na variável vetor do tipo inteiro uma passagem por referência ao vetor originalmente declarado. Valores atribuídos a esta variável serão atribuídos a definição original que efetuou a chamada do procedimento.

OBS2: a variável N recebe passagem de valor, sendo definida como o tamanho do vetor.

Imprimindo um vetor

- Para mostrar um vetor já carregado e ter seus valores mostrados na tela, pode-se utilizar o comando **for**.

```
void  mostra_vetor (int vetor [], int N)
{
   cout << "Vetor digitado: ";
   for (int i = 0; i < N; i++)
   {
            cout <<  vetor [i]  << "  ";
   }
   cout << endl;
}
```

Exemplo completo

Programa que lê um vetor de 100 posições e imprime este valores na tela:

```
void  le_vetor (int vetor [], int N)
{
    for (int i = 0; i < N; i++)
    {
        cout << “Digite o elemento “ << i
             << “do vetor:”;
        cin >> vetor [i];
    }
}
void  mostra_vetor (int vetor [], int N)
{
    cout << “Vetor digitado: “;
    for (int i = 0; i < N; i++)
    {
            cout <<  vetor [i]  << “  ”;
    }
    cout << endl;
}
```

```
Int main()
{
    int numero;
    int vetor [100];

    cout << “Digite tamanho do vetor “;
    cin >> numero;

    le_vetor (vetor , numero);

    mostra_vetor (vetor, numero);
}
```

Exemplo completo

Programa que lê um vetor de N notas, calcula a média destas notas e a quantidade de notas acima da média

```
// le_vetor e mostra_vetor são iguais

float calcula_media  (int vetor [], int N)
{
    float soma = 0;
    for (int i = 0; i < N; i++)
    {
            soma = soma + vetor [i];
    }
    retunr (soma / N);
}
int acima_media (int vetor [], int N; float M)
{
    int qtd = 0;
    for (int i = 0; i < N; i++)
    {
            if (vetor [i] > M)
                qtd++
    }
    return qtd;

}
```

```
Int main()
{
    int numero, acima;
    float media;
    int vetor [100];

    cout << "Digite  numero de notas: ";
    cin >> numero;

    le_vetor (vetor , numero);

    media = calcula_media( vetor, numero)
    cout << "Media = " << media << endl;

    acima = acima_media (vetor, N, media);
    mostra_vetor (vetor, numero);
    cout << "Notas acima da media: ";
    cout << acima;

    system ("pause");

}
```

Exercícios

- Faça um programa que lê um vetor de 10 elementos digitado pelo usuário e retorne a soma dos elementos armazenados no vetor.
- Faça um programa que lê um vetor e uma posição, digitados pelo usuário, e retorne o elemento da posição indicada pelo usuário.
- Faça um programa que lê dois vetores **VetorA** e **VetorB** e retorne um terceiro vetor, **VetorC** tal que:
- **VetorC[i] = VetorA[i] + VetorB[i]**
- Faça um programa que lê um vetor **VetA** e retorne um vetor **VetB** tal que: **VetB[i] = VetA[i]^2**