Estruturas de Dados Homogêneas: Vetores
Algoritmos e Estrutura de Dados I

Instituto de Engenharia – UFMT

Agenda

Objetivos

- Aprender a utilizar vetores.
- Fazer diversas operações sobre vetores.

Introdução

Como armazenar 3 notas?

```
#include<stdio.h>
int main () {
  float nota1, nota2, nota3;
  printf("Digite a nota 1:");
  scanf("%f",&nota1);
  printf("Digite a nota 2:");
  scanf("%f",&nota2);
  printf("Digite a nota 3:");
  scanf("%f",&nota3);
  return 0;
}
```

Introdução

Como armazenar 100 notas?

```
#include<stdio.h>
int main () {
  float nota1, nota2, nota3, /*... ,*/ nota100;
  printf("Digite a nota 1:");
  scanf("%f",&nota1);
  printf("Digite a nota 2:");
  scanf("%f",&nota2);
  printf("Digite a nota 3:");
  scanf("%f",&nota3);
  /* ... */
  printf("Digite a nota 100:");
  scanf("%f",&nota100);
  return 0;
}
```

Declaramos 100 variáveis? Totalmente inviável.

Introdução

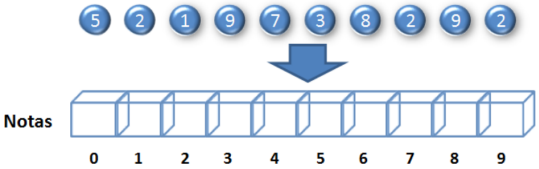
Deseja-se saber quantas notas de um total 10 são maiores que a média.

Introdução

Sem salvar cada uma das notas, não é possível determinar quantas delas são maiores que a média.

Introdução

A solução é utilizar um vetor para armazenar os valores.



Introdução

Vetores

Um vetor é:

- uma variável composta homogênea unidimensional;
- formada por uma sequência de variáveis, todas do mesmo tipo de dados;
- com o mesmo identificador (mesmo nome);
- alocadas sequencialmente na memória.

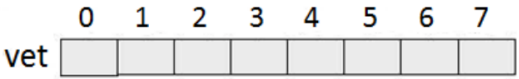
Vetores

Um vetor é:

- uma variável composta homogênea unidimensional,
- formada por uma sequência de variáveis, todas do mesmo tipo de dados,
- com o mesmo identificador (mesmo nome),
- alocadas sequencialmente na memória.

Vetores

Uma vez que as variáveis que compõem o vetor têm o mesmo nome, o que distingue cada uma delas é um índice, que referencia sua localização dentro da estrutura.



Declaração de Vetores

Para declarar um vetor, utiliza-se a forma geral:

```
<tipo> identificador [ <numero de posicoes> ];
```

- Primeiro o tipo dos dados: int, float, char;
- Segundo, o nome da variável, usando as mesmas convenções de uma variável comum: vetor, notas, valores, pesos...
- E por fim, o tamanho do vetor necessário para armazenamento entre colchetes: [5], [10], [1000].
- Exemplos:

```
float notas[100];
int pesos[10];
char nome_completo[30];
```

Vetores

- Observe que a primeira posição do vetor tem índice 0 (zero);
- E a última posição tem índice <numero de posicoes> -1;
- Na linguagem C não se verifica se ocorre um acesso fora dos limites do vetor.

Vetores

- Todo vetor é um espaço linear dividido de acordo com o tamanho que declaramos;
- Considere a declaração de um vetor de 4 posições, conforme o código abaixo:

```
int vetor[4];
```

- Na memória é representado da seguinte forma:

Vetores

- Para acessar ou atribuir um valor é necessário indicar entre colchetes o índice no qual será inserido o valor.
- **Exemplo:**

```
int vetor[4];
vetor[1] = 540;
vetor[3] = 8456;
printf("%d", vetor[1]);
printf("%d", vetor[3]);
```

- Após a atribuição, a memória é representada da seguinte forma:

Exemplos vetores

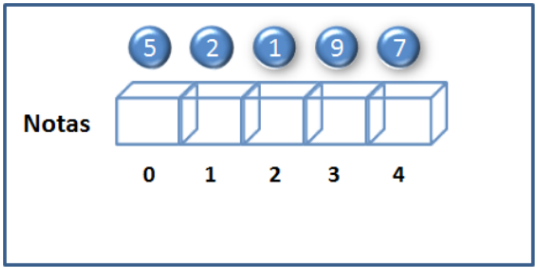
Exemplo: Inserir 5 notas em um vetor, depois disso imprimir os valores inseridos.

```
#include<stdio.h>
int main() {
  int i, vetor[5];
  printf("Digite 5 notas: ");
  for (i=0;i<5;i++)
    scanf("%d", &vetor[i]);
  printf("As notas digitadas foram:");
  for (i=0;i<5;i++)
    printf("%d ", vetor[i]);
  return 0;
}
```

Exemplos vetores

Exemplo de execução

```
Digite 5 notas:
5 2 1 9 7
As notas digitadas foram:
5 2 1 9 7
```



Exemplos vetores

Exemplo: Inserir 5 notas em um vetor, depois disso imprimir os valores em ordem contrária a que foram inseridos.

```
#include<stdio.h>
int main() {
  int i, vetor[5];
  printf("Digite 5 notas: ");
  for (i=0;i<5;i++)
    scanf("%d", &vetor[i]);
  printf("As notas digitadas em ordem contraria foram:");
  for (i=4;i>=0;i--)
    printf("%d ", vetor[i]);
  return 0;
}
```

Exemplos vetores

Exemplo: Modifique o programa anterior de forma que depois de inseridas as 5 notas, calcule a média.

```
#include<stdio.h>
int main() {
  int i, vetor[5], soma=0;
  printf("Digite 5 notas: ");
  for (i=0;i<5;i++)
    scanf("%d", &vetor[i]);
  /* Calculo da media */
  for (i=0;i<5;i++)
    soma += vetor[i];
  printf("A media eh: %d", soma/5);
  return 0;
}
```

Exemplos vetores

Exemplo: Faça uma programa que receba 10 números inteiros e armazene-os em um vetor. O programa deve calcular e mostrar o maior elemento do vetor.

Exemplos vetores

```
#include<stdio.h>
int main() {
  int vetor[10], i, maior, pMaior;
  printf("Digite 10 numeros:");
  for (i=0;i<10;i++)
    scanf("%d",&vetor[i]);
  /* Busca pelo maior */
  maior = vetor[0];
  for (i=1;i<10;i++) {
      if (vetor[i] > maior) {
        maior = vetor[i];
        pMaior = i;
      }
  }
  printf("O maior elemento %d estah na posicao %d.", maior, pMaior);
  return 0;
}
```

Exemplos vetores

Exemplo: Faça um programa que receba 10 números inteiros e armazene-os em um vetor. O programa deve calcular e mostrar dois vetores resultantes, sendo o primeiro com os números pares e o segundo com os números ímpares, pertencentes ao vetor lido.

Exemplos vetores

```
#include<stdio.h>
int main() {
  int  vetor[10], pares[10], impares[10];
  int i, nPares = 0, nImpares = 0;
  printf("Digite 10 numeros: ");
  for (i=0;i<10;i++)
    scanf("%d", &vetor[i]);
  /* Separando pares dos impares */
  for (i=0;i<10;i++) {
    if (vetor[i]%2 == 0) {
      pares[nPares]=vetor[i];
      nPares++;
    } else {
      impares[nImpares]=vetor[i];
      nImpares++;
    }
  }
  /* Continua... */
```

Exemplos vetores

```
#include<stdio.h>
int main() {
 /* Codigo anterior */

  printf("Pares");
  for (i=0;i<nPares;i++) {
    printf("%d ", pares[i]);
  }
  printf("Impares");
  for (i=0;i<nImpares;i++) {
    printf("%d ", impares[i]);
  }
  return 0;
}
```

Inicialização de vetores

- Quando declaramos um vetor, os seus elementos não são inicializados;
- Ao declarar um vetor, é possível atribuir valores iniciais;
- Os valores iniciais são colocados entre chaves. **Exemplo:**

```
int vetor[5] = {16, 36, 3, 8, 26};
```

Inicialização de vetores

- A quantidade de valores entre chaves não deve ser maior que o número de elementos;
- A fim de facilitar a inicialização, a linguagem C permite deixar o número de elementos em branco [ ].
- Neste caso, o compilador vai supor que o tamanho do vetor é igual ao número de valores especificados entre chaves .

```
int vetor[] = {16, 36, 3, 8, 26};
```

Inicialização de vetores

- A fim de facilitar a programação, pode-se definir o tamanho do vetor como uma constante.
- Para definir uma constante, no início do código inclua o comando na forma geral:

```
#define nome_constante valor
```

- Exemplos de declarações de constantes:

```
#define TAM_MAX 100
#define Pi 3.14159
```

Inicialização de vetores

Considere o exemplo a seguir em que se inicializa um vetor com zeros:

```
#include<stdio.h>
#define TAM_MAX 100
int main() {
  float vetor[TAM_MAX];
  int i;
  /* Inicializa o vetor com zero */
  for (i=0;i<TAM_MAX;i++) {
    vetor[i] = 0;
  }
  return 0;
}
```

Busca Sequencial

Dada uma coleção com n valores, pretende-se saber se um determinado valor está presente nessa coleção. Considere que a coleção é implementada utilizando um vetor de n elementos inteiros: vetor[0].. vetor[n-1].

Busca Sequencial

Uma solução possível é percorrer o vetor desde a primeira posição até a última. Para cada posição i, comparamos o elemento na posição com o valor.

- Se foram iguais, significa que o valor existe no vetor.
- Se chegou ao final do vetor e não encontrou nenhum valor igual, significa que o valor não existe no vetor.

Busca Sequencial

Passos: 1. Inicialização

```
#define TAM_MAX 100
/* ... */
i = 0;
encontrado = 0; // Falso
```

Busca Sequencial

2. Pesquisa

```
while (i<TAM_MAX && !encontrado) {
  if (vetor[i] == valor)
    encontrado = 1; // Verdadeiro
  else
    i++;
}
```

Busca Sequencial

3. Tratamento do resultado

```
if (encontrado)
  printf("O valor %d se encontra na posicao %d.", valor, posicao);
else
  printf("Valor nao encontrado");
```

Busca Sequencial

```
#include<stdio.h>
#define TAM_MAX 100
int main() {
  float vetor[TAM_MAX], valor;
  int i = 0, posicao, encontrado = 0; // Falso
  for (i=0;i< TAM_MAX;i++)
    scanf("%f",&vetor[i]); // Preenchendo vetor
  printf("Digite o valor a ser buscado no vetor:");
  scanf("%f",&valor);
  while (i<TAM_MAX && !encontrado) {
    if (vetor[i] == valor) {
      encontrado = 1; // Verdadeiro
      posicao = i;
    }
    else
      i++;
  }
  /* Continua */
```

Busca Sequencial

```c
int main() {
  /* Codigo anterior */
  if (encontrado)
    printf("O valor %f se encontra na posicao %d.", valor, posicao);
  else
    printf("Valor nao encontrado");
  return 0;
}
```

Busca Sequencial

Faça uma função chamada busca_vetor que receba como parâmetros de entrada um vetor preenchido com TAM_MAX elementos e um valor. Caso o valor exista no vetor, a função deve retornar sua posição no vetor, caso contrário, deve retornar -1.

Busca Sequencial

```
int busca_vetor(float vetor[TAM_MAX], float valor) {
  int i = 0, posicao, encontrado = 0; // Falso
  while (i<TAM_MAX && !encontrado) {
    if (vetor[i] == valor) {
      encontrado = 1; // Verdadeiro
      posicao = i;
    }
    else
      i++;
  }
  if (encontrado)
    return posicao;
  else
  return -1;
}
```

Fim