

ESTRUTURAS COMPOSTAS

Variáveis Compostas Unidimensionais VETOR

Baseado nos slides de Rosely Sanches e Simone Senger de Souza

ESTRUTURAS COMPOSTAS

- Pode-se organizar os dados dos tipos simples em tipos mais complexos formando-se as ESTRUTURAS COMPOSTAS
- Exemplo:
 - variáveis compostas unidimensionais (VETOR)

VETOR

- O conceito de VETOR será introduzido através de um exemplo.
- Suponhamos o seguinte problema:

Calcular a média aritmética das notas de 3 alunos.

Exibir a média e as notas que estão abaixo da média

1ª Solução

- Ler as três notas uma primeira vez para calcular a média.
- Ler novamente cada nota para comparar com a média e verificar quais notas estão abaixo da média

Algoritmo

Variaveis

I: inteiro

soma, nota, media: real

Inicio

soma = 0

para i de 1 até 3

faça início

leia (nota)

soma = soma + nota

fim

fim-para

media = soma/3

escreva (media)

para i de 1 até 3

faça início

leia (nota)

se (nota < media)

então escreva (nota, “abaixo da média”)

fim-se

fim

fim-para

Fim.

Algoritmo

Variaveis

I: inteiro

soma, nota, media: real

Inicio

soma = 0

para i de 1 até 3

faça início

leia (nota)

soma = soma + nota

fim

fim-para

media = soma/3

escreva (media)

para i de 1 até 3

faça início

leia (nota)

se (nota < media)

então escreva (nota, "abaixo da média")

fim-se

fim

fim-para

Fim.



Ineficiente

2ª Solução

- Ler as três notas e armazenar na memória do computador, dando um nome diferente para cada nota.

Algoritmo Variaveis

nota1, nota2, nota3, media: real

Inicio

leia (nota1,nota2,nota3)

media = (nota1 + nota2 + nota3)/3

escreva (media)

se (nota1 < media)

então escreva (nota1,"abaixo da média")

fim-se

se (nota2 < media)

então escreva (nota2,"abaixo da média")

fim-se

se (nota3 < media)

então escreva (nota3,"abaixo da média")

fim-se

Fim

o programa só
vale para três notas

3ª Solução

- Associar o nome NOTA ao CONJUNTO
ORDENADO de notas

$$\text{NOTA} = \{N1, N2, \dots N1000\}$$

$1^a \quad 2^a \quad 1000^a$

3ª Solução

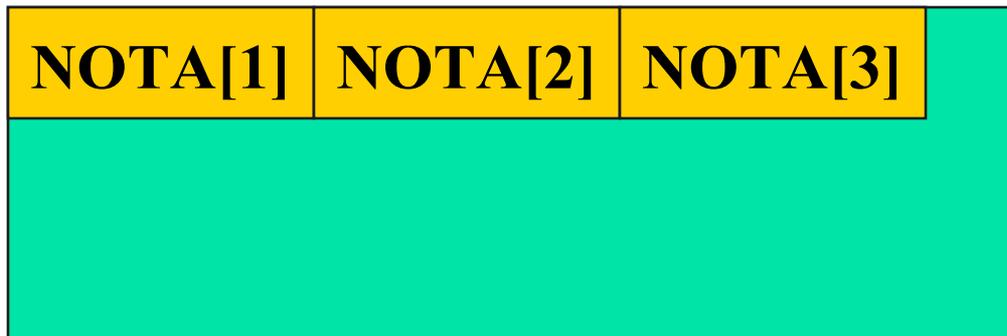
NOTA = {N1, N2, ... N1000}

1ª 2ª 1000ª

- para fazer referência ou selecionar uma nota específica uso um **índice**
- Exemplo:
 - a 3ª nota é indicada por **NOTA[3]**
 - a 1000ª nota é indicada por **NOTA[1000]**
 - uma kª nota é indicada por **NOTA[k]**

VARIÁVEL INDEXADA

- Cada **variável indexada** é associada à uma posição de memória, como acontece com variáveis simples.
- Exemplo:

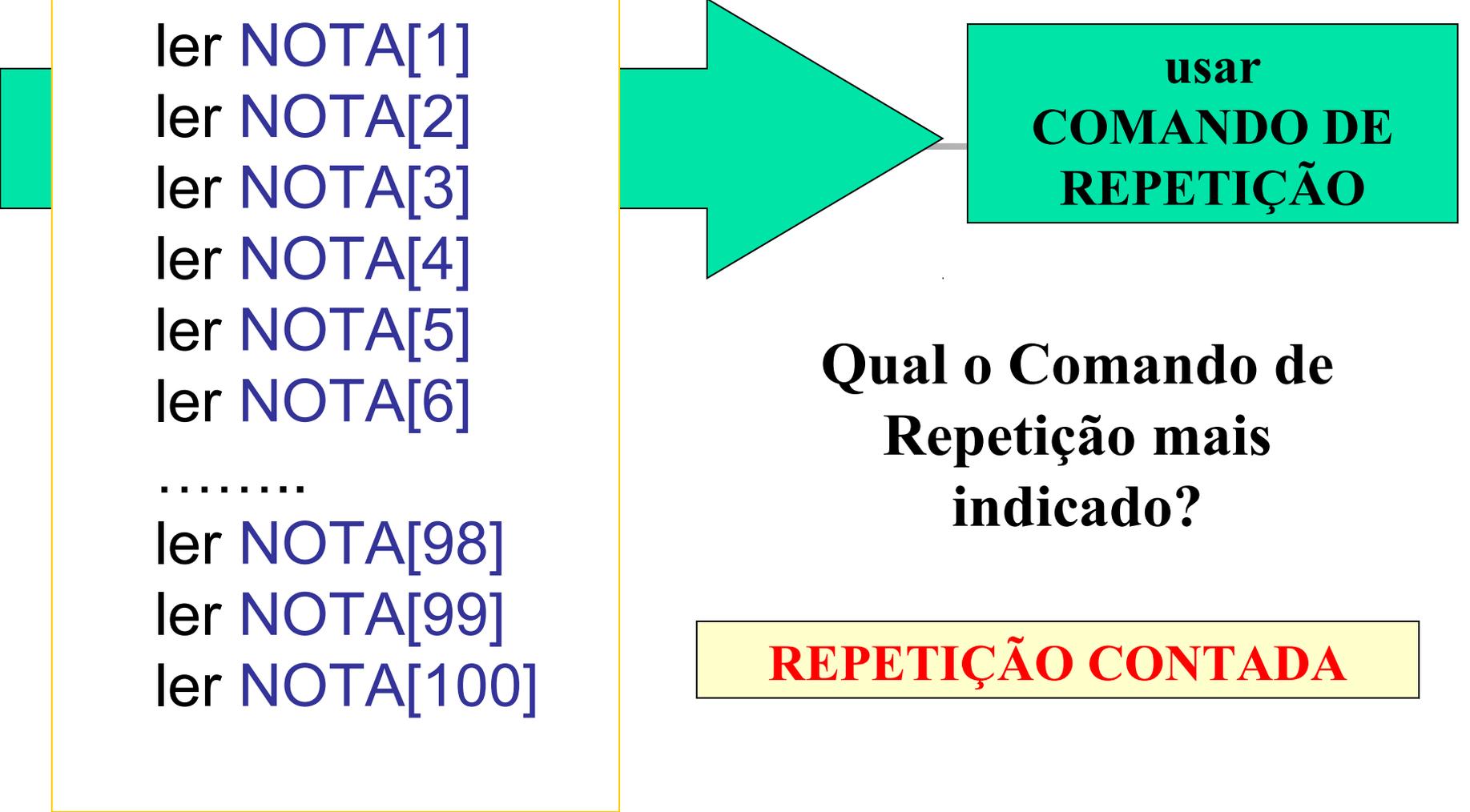


VETOR

- Um **VETOR** é um conjunto ordenado que contém um número fixo de elementos
- Todos os elementos do vetor devem ser do mesmo tipo

VETOR - Exemplo 1

- Ler um conjunto de 100 notas, armazená-las no vetor denominado NOTA e escrever este vetor.

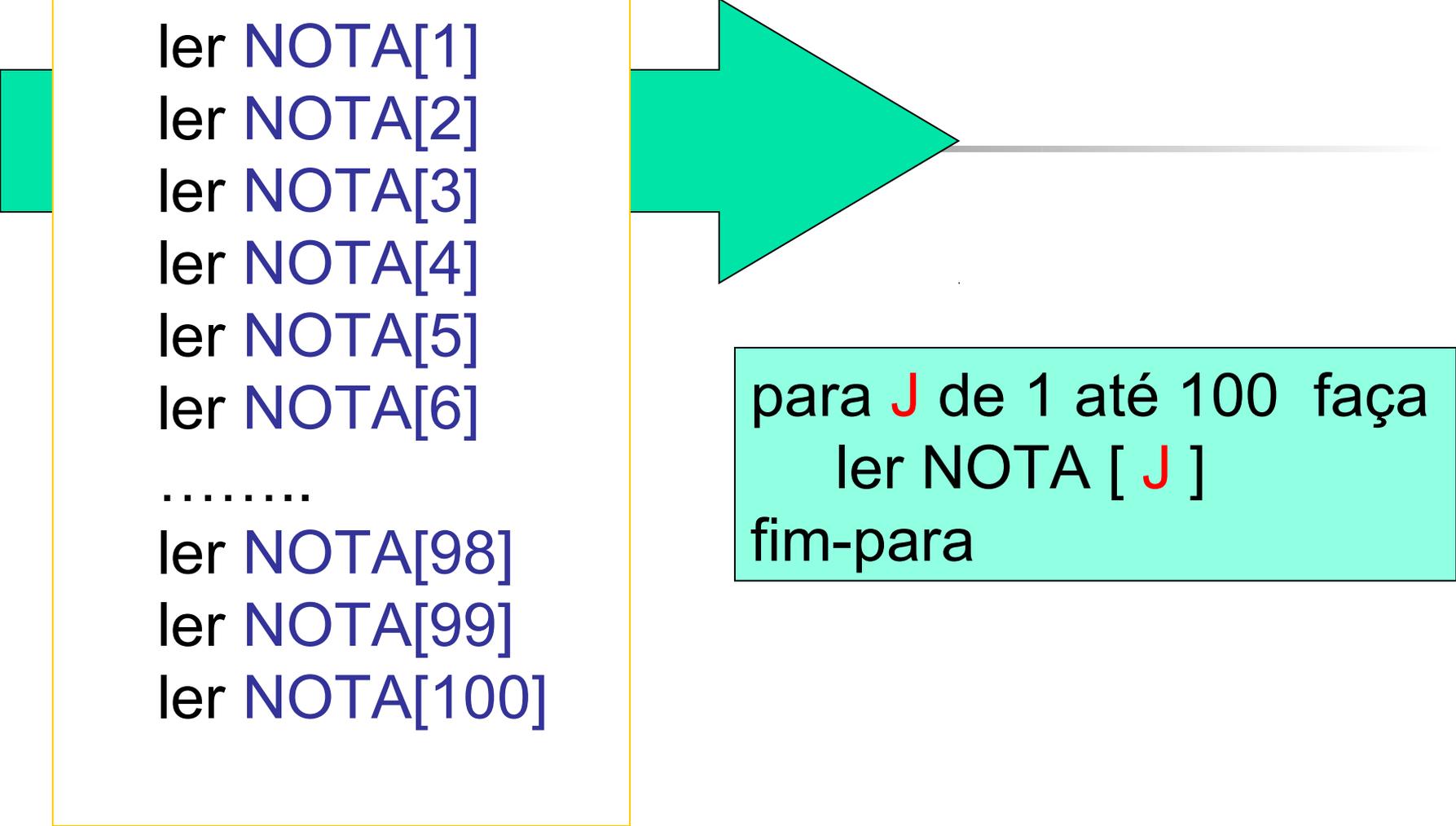


```
ler NOTA[1]
ler NOTA[2]
ler NOTA[3]
ler NOTA[4]
ler NOTA[5]
ler NOTA[6]
.....
ler NOTA[98]
ler NOTA[99]
ler NOTA[100]
```

**usar
COMANDO DE
REPETIÇÃO**

**Qual o Comando de
Repetição mais
indicado?**

REPETIÇÃO CONTADA



```
ler NOTA[1]
ler NOTA[2]
ler NOTA[3]
ler NOTA[4]
ler NOTA[5]
ler NOTA[6]
.....
ler NOTA[98]
ler NOTA[99]
ler NOTA[100]
```

```
para J de 1 até 100 faça
    ler NOTA [ J ]
fim-para
```

ler NOTA[1]
ler NOTA[2]
ler NOTA[3]
ler NOTA[4]
ler NOTA[5]
ler NOTA[6]
.....
ler NOTA[98]
ler NOTA[99]
ler NOTA[100]

para **J** de 1 até 100 faça
 ler NOTA [**J**]
fim-para

mesmo efeito que

para **I** de 1 até 100 faça
 ler NOTA [**I**]
fim-para

mesmo efeito que

para **K** de 1 até 100 faça
 ler NOTA [**K**]
fim-para

I, J e K
são apenas
índices que
assumem um
valor e que
junto como
nome dado ao
conjunto
formam a
variável
indexada
NOTA

```
para J ← 1 até 100 faça  
  ler NOTA [ J ]  
fim-para
```

mesmo efeito que

```
para I ← 1 até 100 faça  
  ler NOTA [ I ]  
fim-para
```

mesmo efeito que

```
para K ← 1 até 100 faça  
  ler NOTA [ K ]  
fim-para
```

Algoritmo

Variaveis

j: inteiro

nota[100]: real

Inicio

para j de 1 até 100

faça leia (nota[j])

fim-para

para j de 1 até 100

faça escreva (nota[j])

fim-para

Fim

VETOR - Exemplo 2

- Ler um conjunto de 100 notas, armazená-las no vetor denominado NOTA, calcular a soma dessas notas, escrever o vetor de notas e a soma das notas.

Algoritmo

Variaveis

i: inteiro

nota[100], soma: real

Inicio

para i de 1 até 100

faça leia (nota[i])

fim-para

soma = 0

para i de 1 até 100

faça soma = soma + nota[i]

fim-para

para i de 1 até 100

faça escreva (nota[i])

fim-para

escreva (soma)

Fim

VETOR - Exemplo 3

- Ler um conjunto de 100 notas armazenando-as no vetor denominado NOTA.
- Verificar se existe nota 10.0. Se existir, dizer quantas existem

Algoritmo

Variaveis

contador, i: inteiro

nota[100]: real

Inicio

para **i** de 1 até 100

faça leia (nota[**i**])

fim-para

contador = 0

para **i** de 1 até 100

faça se (nota[**i**] = 10)

então contador = contador + 1

fim-se

fim-para

escreva (contador)

Fim

VETOR - Exemplo 4

- Ler um conjunto de 100 notas armazenando-as no vetor denominado NOTA.
- Calcular a média, verificar e exibir as notas abaixo da média.



Declaração (C) de Variável Indexada Unidimensional

- Deve ser especificado o número máximo de elementos do conjunto
- Deve ser especificado o tipo dos elementos do conjunto
- Exemplo:

```
float x[100];
```

Declaração (C) de Variável Indexada Unidimensional

Tipo dos elementos do conjunto

Nome da Variável

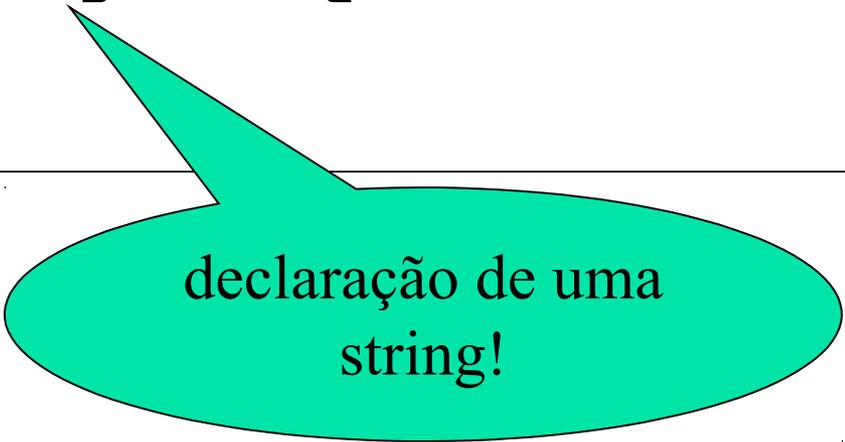
Número máximo de elementos do conjunto

```
float x[100];
```

Declaração (C) de Variável Indexada Unidimensional - Exemplos

- Declaração de um vetor S com no máximo 100 elementos do tipo character

```
char S[100];
```



declaração de uma
string!

Declaração (C) de Variável Indexada Unidimensional

- Pode ser definida uma **constante** e esta ser utilizada no dimensionamento.
- Exemplo:

```
//Início do programa C
const int MAX = 5; OU
#define MAX 5
//declaração:
float X[MAX];
```

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
# define MAX 10
```

```
int main(int argc, char *argv[])
```

```
{
```

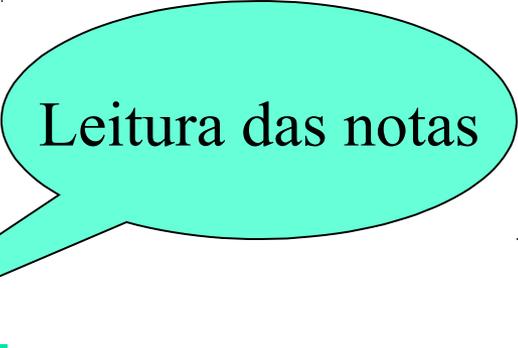
```
    float nota[MAX], media, soma = 0;
```

```
    int i;
```

```
    printf("Entre com as notas dos alunos:");
```

```
    for(i=0; i<MAX; i++) //leitura das notas
```

```
        scanf("%f", &nota[i]);
```



Leitura das notas

```
for(i=0; i<MAX; i++)  
    soma = soma + nota[i];  
media = soma/MAX;
```

Cálculo da média

```
printf("média da turma = %.1f\n", media);
```

Exibe a média

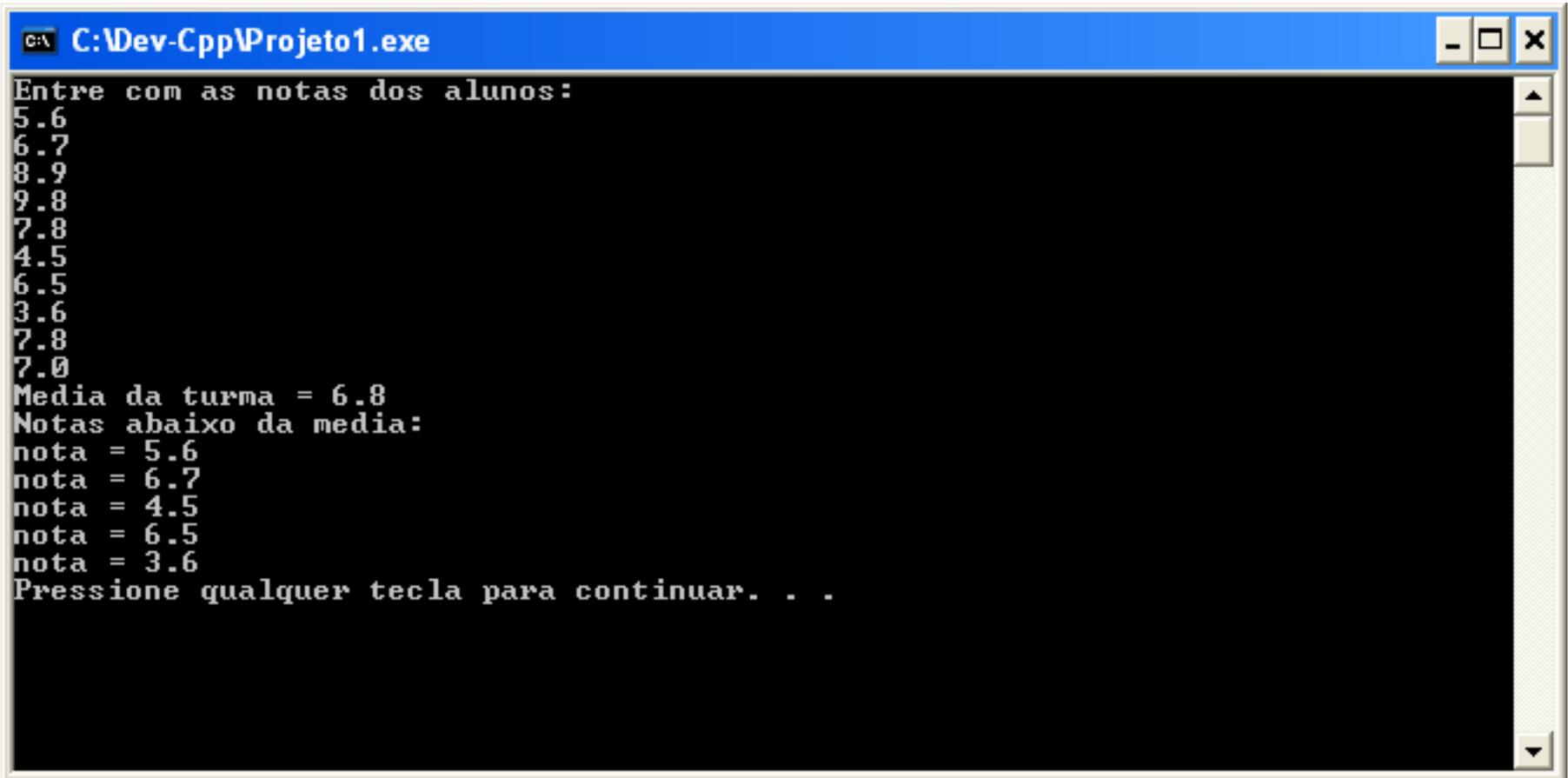
```
printf("Notas abaixo da media:\n");  
for(i=0; i<MAX; i++)  
    if (nota[i] < media)  
        printf("nota = %.1f\n", nota[i]);
```

Escrita de todas
as notas

```
system("PAUSE");
```

```
return 0;
```

```
}
```



```
C:\Dev-Cpp\Projeto1.exe
Entre com as notas dos alunos:
5.6
6.7
8.9
9.8
7.8
4.5
6.5
3.6
7.8
7.0
Media da turma = 6.8
Notas abaixo da media:
nota = 5.6
nota = 6.7
nota = 4.5
nota = 6.5
nota = 3.6
Pressione qualquer tecla para continuar. . .
```

Exercício 1: em classe

- Ler e armazenar um conjunto de **N** números inteiros (N é lido e é menor que 100).
- Encontrar e exibir o maior deles.

Exercício 2: em classe

- Elaborar um algoritmo que lê um conjunto de **30 valores inteiros** e os coloca em um vetor. Calcular e mostrar:
 - Os números pares;
 - A quantidade de números pares;
 - Os números ímpares
 - A quantidade de números ímpares;

Exercícios propostos

1. Escrever um algoritmo que lê **dois vetores** de 10 elementos inteiros e multiplica os elementos de **mesmo índice**, colocando o resultado em um terceiro vetor. No final, mostrar os dois vetores lidos e o vetor resultante.
2. Faça um algoritmo que lê um **vetor de 30 números** inteiros e um número **n** a ser procurado no vetor. Escrever **quantas vezes n** aparece no vetor e **em quais posições**.
3. Desenvolva uma solução (pode ser somente os passos) para **ordenar** um vetor de 100 números.

