

Ex: Ler as notas de 30 alunos.
Calcular e informar a média da turma.

Algoritmo MédiaTurma

{ Obtém as notas de 30 alunos e informa a média }

Entradas: 30 notas

Saída: média das 30 notas lidas

Processamento

1. Início
2. Inicializar somatório das notas
3. Repetir 30 vezes (para cada aluno)
 - 3.1 obter a nota de um aluno
 - 3.2 acrescentar esta nota ao somatório de notas
4. Calcular a média
5. Informar a média
6. Fim

Slide 1

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Ex: Ler as notas de 30 alunos.
Calcular e informar a média da turma.

```
//calcula média de uma turma
#include <stdio.h>
#include <stdlib.h>

int main ( )
{
    float nota; // nota de um aluno
    float media, soma;
    int aluno; // variável de controle da repetição
    soma = 0; // inicializa a soma das notas
    for (aluno = 1; aluno <= 30; aluno++) // para cada aluno da turma
    {
        scanf("%f", &nota); //obtem a nota do aluno
        soma = soma + nota;
    }
    media = soma / 30;
    printf("Media da turma: %7.2f", media);
    system("pause");
    return 0;
}
```

Slide 2

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Ex: Ler as notas de 30 alunos. Calcular e informar a média da turma. Informar, ainda, as notas que são superiores à média calculada.

Algoritmo MédianotasSup

{ Obtém as notas de 30 alunos.

Informa a média e as notas superiores à média }

Entradas: 30 notas

Saídas: média das 30 notas lidas
notas superiores à média

Processamento

1. Início
2. Inicializar somatório das notas
3. Repetir 30 vezes (uma vez para cada aluno)
 - 2.1 obter a nota de um aluno
 - 2.2 acrescentar esta nota ao somatório de notas
3. Calcular e informar a média
4. Informar as notas superiores à média
5. Terminar



Slide 3

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Ex: Ler as notas de 30 alunos. Calcular e informar a média da turma. Informar, ainda, as notas que são superiores à média calculada.

```
//calcula media de uma turma e notas superiores a media
#include <stdio.h>
int main ( )
{
    float n1, n2, n3, n4, n5, { ... }, n30 // uma variável para cada nota
    float media, soma;
    // obtém as 30 notas
    scanf("%f %f (...) %f", &n1, &n2, (...) &n30);
    soma = n1+n2+n3+n4+n5+n6+ (...) n30;
    media = soma / 30;
    printf("Media da turma: %7.2f", media);
    if (n1 > media) // testa cada nota !!!
        printf("%f", n1);
    if (n2 > media)
        printf("%f", n2);
    { ... }
    return 0;
}
```

É a única solução ?

Slide 4

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Arranjos

- 1 dimensão ←
- várias dimensões

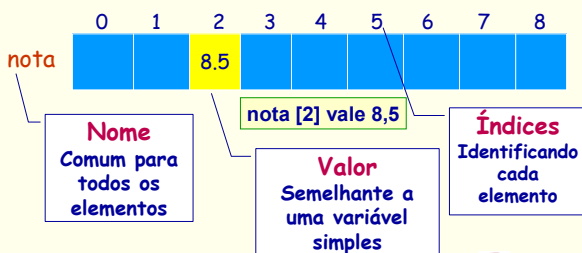
Slide 5

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Arranjo (vetor)

- conjunto ordenado de informações de mesma natureza.
- **elementos:** todos do mesmo tipo.
- acesso randômico.



Slide 6

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Arranjo

C - declaração:

tipo nome_do_arranjo [número de elementos da dimensão];

- número de elementos da dimensão: inteiro ou representar um valor inteiro
- tipo: *int*, *float*, *double* (entre outros).

Ex:

```
float nota[30]; // vetor de 30 elementos reais
int x[15]; // vetor de 15 elementos inteiros
```

ATENÇÃO: em C, strings (= cadeias de caracteres) são arranjos de caracteres. Serão vistos na próxima aula!

Slide 7

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Arranjo

C - utilização:

Variável indexada

nome_do_arranjo[indice]

Índice:

- Constante, nome de variável ou expressão que corresponda a um valor inteiro (ou resulte em) um valor inteiro;
- Sempre inicia em 0.
- Índice corresponde à posição - 1.

Slide 8

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Arranjo

C - utilização:

nome_do_arranjo[posição - 1]



Ex:

```
float nota[9]; // vetor de 9 elementos
scanf("%f", &nota[5]); // 6º elemento
nota[1] = 7.5;
nota[3] = nota[1] + 2;
if (nota[1] > 6.0)
    printf("Aprovado");
...
```

nota[5]

Slide 9

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Arranjo

C - inicialização de conteúdos:

Arranjos, como as demais variáveis, correspondem a locais identificados de memória.

O preenchimento de áreas de arranjos ocorrem através de leitura e atribuição.

Os conteúdos iniciais são imprevisíveis. Por isso, pode ser necessária a inicialização destes conteúdos.

Arranjos podem ser inicializados ao serem criados, quando assim especificado.

Slide 10

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

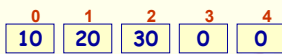
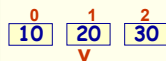
Arranjo

C - inicialização na declaração:

tipo nome[num_elem] = {valor0, valor1, valornum_elem-1};

Exemplos:

```
int v[3] = {10, 20, 30}; // inicializa todos
int x[5] = {10, 20, 30}; // zera 2 últimos
int nros[100] = {0}; // inicia com todos zerados
int vi1[3]={1,0,7}, vi2[3]={0}, vi3[3];
```



Slide 11

Profa. Cora H.F. Pinto Ribeiro

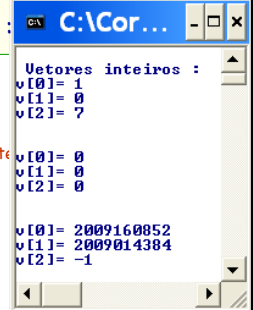
UFRGS Informática

Arranjo

C - inicialização na declaração :

```
.....
// declarações :
int vi1[3]={1,0,7}, vi2[3]={0}, vi3[3];
int i;

system("color f1"); // altera cores da tela
printf("\n Vetores inteiros : \n");
for (i=0;i<=2;i++)
    printf("v[%d]= %d \n",i,vi1[i]);
printf("\n\n");
for (i=0;i<=2;i++)
    printf("v[%d]= %d \n",i,vi2[i]);
printf("\n\n");
for (i=0;i<=2;i++)
    printf("v[%d]= %d \n",i,vi3[i]);
```



Slide 12

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Ex1: Preencher por leitura um arranjo **valor** de 100 elementos inteiros.

| Índice | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 99 |
|--------|---|---|---|---|---|---|---|---|---|-----|
| Valor | | | X | | | | | | | ... |

```
int valor[100]; // vetor de 100 elementos
int indice;
// 1º elemento: [0]; 100º elemento: [99]
for (indice = 0; indice < 100; indice++)
{ printf("Informe elemento %d", indice + 1);
  scanf("%d", &valor[indice]);
}
...
```

Ex2: Somar os elementos de um arranjo **x** de 200 elementos.

```
...
soma = 0; // variável acumuladora
// índice varia de 0 a 199!
for (indice = 0; indice < 200; indice++)
    soma = soma + x[indice];
...
```

Slide 13

RGS Informática

Ex: Ler as notas de 30 alunos. Calcular e informar a média da turma. Informar, ainda, as notas que são superiores à média calculada.

```
// Calcula média de uma turma e informa notas superiores a média
#include <stdio.h>
#include <stdlib.h>
int main ( )
{
    float nota [30]; // arranjo para as notas
    float media, soma;
    int aluno; // variável de controle da repetição
    soma = 0; // inicializa soma
    for (aluno = 0; aluno < 30; aluno++) // para cada aluno
    {
        printf("Informe nota do aluno %d:", aluno + 1); // identifica aluno
        scanf("%f", &nota[aluno]); // obtém sua nota
        soma = soma + nota[aluno]; // acumula a soma
    }
    media = soma / 30;
    printf("Média da turma: %.7.2f", media);
    for (aluno = 0; aluno < 30; aluno++) // mostra notas > media
        if (nota[aluno] > media)
            printf("\nAluno %2d com média: %.7.2f", aluno + 1, nota[aluno]);
    system("pause");
    return 0;
}
```

Slide 14

PROFA. LORA H.F. PINTO KIBEIRO

UF-RGS Informática

Ex: Os preços de mercadorias de uma loja são armazenados em um arranjo **preco**. Os índices do arranjo correspondem aos códigos das mercadorias (10 a 99). Informe o código da mercadoria cujo preço é R\$ 123,00 (supor todos os preços diferentes)

```
// código da mercadoria com preço determinado
#include <stdio.h>
#include <stdlib.h>
int main ( )
{
    float preco [90]; // existem 90 produtos
    int cod; // posição do elemento = cod - 10
    // obter preços, usando a variável de controle com o código:
    for (cod = 10; cod < 100 + 10; cod++)
    {
        printf("Informe preço do produto %d : ", cod);
        scanf("%f", &preco[cod-10]);
    }
    // procurar produto de preço R$ 123,00
    for (cod = 10; cod < 100 + 10; cod++)
        if (preco[cod - 10] == 123)
            printf("O produto %d custa R$123,00. ", cod);
    system("pause");
    return 0;
}
```

RS Informática

Ex: Os preços de mercadorias de uma loja são armazenados em um arranjo **preco**. Os índices do arranjo correspondem aos códigos das mercadorias (10 a 99). Informe o código da mercadoria cujo preço é R\$ 123,00 (supor todos os preços diferentes)

```
// código da mercadoria com preço determinado
#include <stdio.h>
#include <stdlib.h>
int main ( )
{
    float preco [90]; // existem 90 produtos
    int cod; // posição do elemento = cod
    // obter preços, usando a variável de controle com o código:
    for (cod = 0; cod < 100; cod++)
    {
        printf("Informe preço do produto %d : ", cod);
        scanf("%f", &preco[cod]);
    }
    // procurar produto de preço R$ 123,00
    for (cod = 0; cod < 100; cod++)
        if (preco[cod] == 123)
            printf("O produto %d custa R$123,00. ", cod + 10);
    system("pause");
    return 0;
}
```

Problemas desta solução:

- mesmo depois de encontrar, continua percorrendo o arranjo;
- não informa se não encontrou o valor procurado.

RS Informática

```
// solução 1 - com variável flag
// procura código de mercadoria com preço determinado
#include <stdio.h>
#include <stdlib.h>
int main ( )
{
    float preco[90];
    int cod;
    int achou; // 0 = falso, 1 = verdadeiro
    // obter preços
    for (cod = 0; cod < 90; cod++)
        scanf("%f", &preco[cod]);
    // procurar produto de preço R$ 123,00
    achou = 0; // inicializa como não encontrado
    cod = 0;
    do
    {
        if (preco[cod] == 123)
            achou = 1; // sinaliza ter encontrado
        else
            cod = cod + 1;
    } while (cod < 90 && achou == 0);
    if (achou == 1)
        printf("Código do produto = %d", cod);
    else
        printf("Não encontrou!");
    system("pause");
    return 0;
}
```

Informática

```
// solução 2 - usando posição final do laço
// código da mercadoria com preço determinado
#include <stdio.h>
#include <stdlib.h>
int main ( )
{
    float preco [90]; // existem 90 produtos
    int cod; // posição do elemento = cod
    // obter preços
    for (cod = 0; cod < 90 ; cod++)
    {
        printf("Informe preço do produto %d : ", cod + 10);
        scanf("%f", &preco[cod]);
    }
    // procurar produto de preço R$ 123,00
    cod = 0;
    // encerra laço onde encontrou ou fora do vetor
    while (preco[cod] != 123 && cod < 90)
        cod++; // incrementa índice
    if (cod < 90) // está apontando para o elemento procurado
        printf("Produto %d custa R$123,00.", cod + 10);
    else // ou posição está inválida - não encontrou
        printf("não existe produto com este preço.");
    system("pause");
    return 0;
}
```

3