

Algoritmos e Estruturas de Dados I

Prof. Lúcio Mauro Pereira

07/10/2024

Arquivos: outras formas de *i/o*



Modos de abertura de um arquivo com **fopen()**:

w Cria um fluxo para escrita no arquivo - *sobrescreve*

a Cria um fluxo para escrita no arquivo – *ao final*

r Cria um fluxo para leitura do arquivo – *NULL se não houver*

w+ Acrescenta permissão para leitura

a+ Acrescenta permissão para leitura

r+ Acrescenta permissão para escrita



Modos de abertura de um arquivo com **fopen()**:

wb Cria um fluxo para escrita no arquivo - *sobrescreve*

ab Cria um fluxo para escrita no arquivo – *ao final*

rb Cria um fluxo para leitura do arquivo – *NULL se não houver*

wb+ ou **w+b** Acrescenta permissão para leitura

ab+ ou **a+b** Acrescenta permissão para leitura

rb+ ou **r+b** Acrescenta permissão para escrita



Escrevendo com *fwrite()*:

Amplia parâmetros de manipulação de arquivo

Permite manipular coleção de registros

Origem do dado: Posição da memória

Número de bytes por linha: Tamanho do registro

Quantidade de linhas: Número de registros

Arquivo: Ponteiro para o arquivo



Escrevendo com ***fwrite()***:

Origem do dado: Posição da memória

Número de bytes por linha: Tamanho do registro

Quantidade de linhas: Número de registros

Arquivo: Ponteiro para o arquivo

```
FILE* arquivo = fopen("exemplo.dat", "w");
```

```
int x = 123;
```

```
fwrite(&x, sizeof(int), 1, arquivo);
```



Lendo com *fread()*:

Origem do dado: Posição da memória

Número de bytes por linha: Tamanho do registro

Quantidade de linhas: Número de registros

Arquivo: Ponteiro para o arquivo

```
FILE* arquivo = fopen("exemplo.dat", "rb");
```

```
int x;
```

```
fread(&x, sizeof(int), 1, arquivo);
```

Introdução aos arranjos

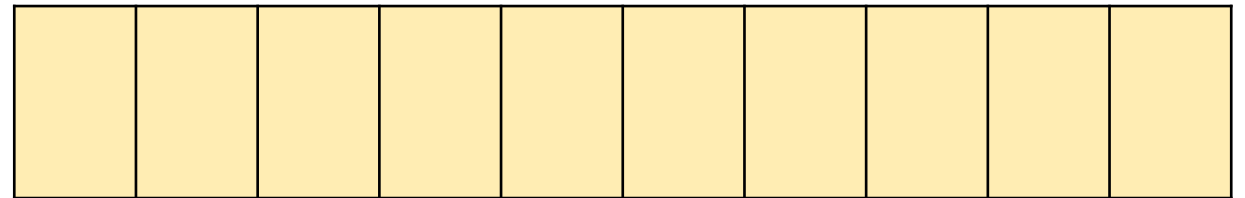
Estruturas de dados estáticas

Como representar uma coleção de dados?

Uma variável pode representar uma coleção de dados

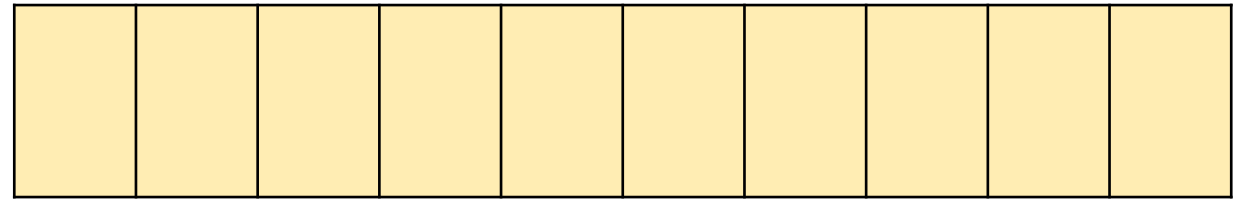
Cada elemento é identificado por sua posição na coleção (arranjo)

Observe a necessidade de serem *bytes* contíguos



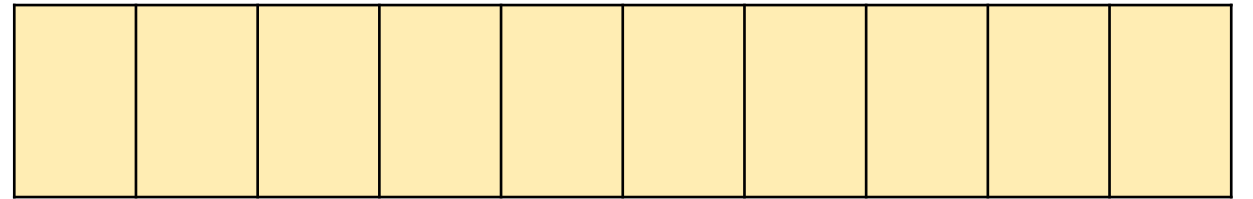
Arranjo unidimensional, vetor, *array*

Declarando um vetor em C



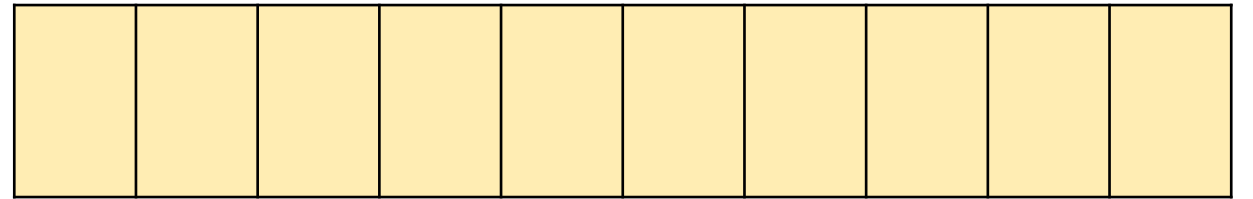
Arranjo unidimensional, vetor, array

Declarando um vetor em C e inicializando-o



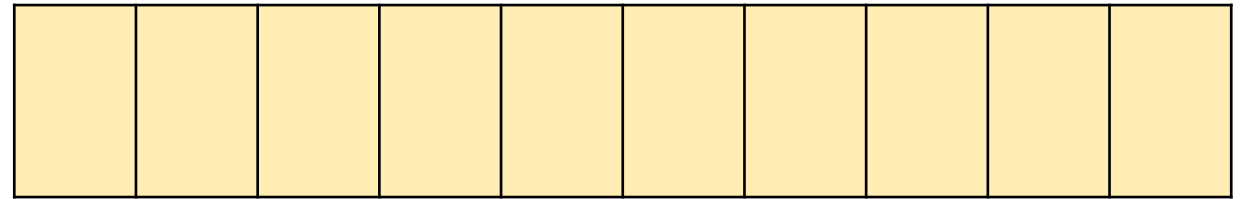
Arranjo unidimensional, vetor, array

Declarando um vetor em C e lendo-o



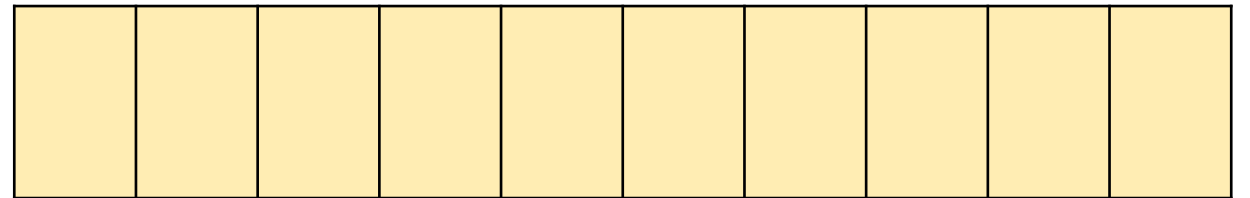
Arranjo unidimensional, vetor, array

Escrevendo um vetor em C



Arranjo unidimensional, vetor, array

Passando um vetor como parâmetro



Primeiro exemplo

Ler idade da turma (10 alunos).
Identificar o número de idades acima da média.

