

Introdução à Computação

Vetores

Prof. Dr. Marcos Paulino Roriz Junior (marcosroriz@ufg.br)



UFG

UNIVERSIDADE
FEDERAL DE GOIÁS



ENGENHARIA DE
TRANSPORTES

FCT
FACULDADE DE
CIÊNCIAS E TECNOLOGIA



UFG
UNIVERSIDADE
FEDERAL DE GOIÁS

Até agora...

Para diversas aplicações, os tipos primitivos não são suficientes para representar ou armazenar todas as possíveis informações em um programa

Uma variável simples armazena apenas 1 valor de um certo tipo

- E se quisermos guardar, para usar no programa, um conjunto de valores relacionados ?

Exemplos:

- a velocidade, ano e peso de um veículo?
- as notas de prova, testes, trabalho de um aluno



Problemática

Imagine o seguinte problema

“leia as velocidade de cinco veículos e depois imprima as velocidades que são maiores do que a média”

Um algoritmo para esse problema poderia ser o mostrado a seguir.



Imagine o seguinte problema

“leia as velocidade de cinco veículos e depois imprima as velocidades que são maiores do que a média”

Um algoritmo para esse problema poderia ser o mostrado a seguir.

```
#include <stdio.h>

int main() {
    float v1, v2, v3, v4, v5;
    printf("Digite as 5 velocidades");
    scanf("%f", &v1);
    scanf("%f", &v2);
    scanf("%f", &v3);
    scanf("%f", &v4);
    scanf("%f", &v5);

    float media = (v1 + v2 + v3 + v4 + v5) / 5;
    if (v1 > media) { printf ("Vel: %f\n", v1); }
    if (v2 > media) { printf ("Vel: %f\n", v2); }
    if (v3 > media) { printf ("Vel: %f\n", v3); }
    if (v4 > media) { printf ("Vel: %f\n", v4); }
    if (v5 > media) { printf ("Vel: %f\n", v5); }
}
```



Problemática

- O algoritmo anterior apresenta uma solução possível para o problema apresentado
- Porém, essa solução é inviável para grandes quantidades de veículos
- Imagine se tivéssemos de processar 1000 veículos



Problemática

Para 1000 veículos, precisamos de:

- Uma variável para armazenar a velocidade de cada veículo
 - 1000 variáveis
- Um comando de leitura para cada velocidade
 - 1000 scan
- Um somatório de 1000 velocidades
- Um comando de teste para cada velocidade
 - 1000 comandos if.
- Um comando de impressão na tela para cada velocidade
 - 1000 print



Vetores (Arrays)

As variáveis têm relação entre si

- todas armazenam notas de velocidades
- Podemos declará-las usando um ÚNICO nome para todos os 1000 veículos

velocidades: vetor de 1000 valores acessados por um índice

- Isso é um tipo de vetor/arranjo!
- No C é denominado (array)

0 2 3 4 5 6 ... 998 999

velocidades =



Vetores

Arrays são agrupamentos de dados adjacentes na memória **do mesmo tipo**.

Vetor é um **array de uma dimensão**.

Declaração:

- **tipo_dado** nome_array[tamanho];

O comando acima define um array de nome *nome_array*, capaz de armazenar tamanho elementos adjacentes na memória do tipo tipo_dado

- Ex: **float** velocidades[1000];
- **int** notas[100];



Vetores

- O tamanho do vetor/array deve ser um valor inteiro constante

```
int main() {  
    int const M = 20;  
  
    int v[M];  
    int v2[100];  
}
```



Vetores (Array/Arranjos)

O padrão C99 possui arrays de comprimento variável

- O tamanho pode ser especificado em tempo de execução
- É um recurso de implementação opcional no C11

```
int main() {  
    int n;  
    printf("Digite o número de veículos: ");  
    scanf("%d", &n);  
  
    int velocidades[n];  
}
```

Vetores (Array/Arranjos)

- Assim como em variáveis podemos atribuir um valor inicial a um vetor, esta inicialização deve ser feita no momento da criação do vetor.
- Exemplo:

```
#include <stdio.h>

int main() {
    int velocidades[4] = {20, 55, 22, 33};

    for (int i = 0; i < 4; i++) {
        printf("%d\n", velocidades[i]);
    }
}
```



Pseudocódigo?

- **DECLARE** nome_vetor[tamanho] tipo

Exemplos:

- **DECLARE** A[5] NUMÉRICO
- **DECLARE** VELOCIDADES[1000] LÓGICOS

Como referenciar?

- **A[1] <- leia**



Acessando elementos do vetor

- Em um vetor, os elementos são acessados especificando o índice desejado entre colchetes []
- A numeração começa no número 0
- Isto significa que um array de 1000 elementos terá índices de 0 a 999:
- velocidades[0], velocidades[2], velocidades[3], ..., velocidades[999]



Acessando elementos do vetor

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

```
int main() {  
    int velocidades[1000];  
  
    velocidades[0] = 20;  
    velocidades[1] = 50;  
    velocidades[2] = 30;  
    // ...  
    velocidades[999] = 44;  
}
```

0 1 2 3 4 5 ... 998 999

20	50	30							44
----	----	----	--	--	--	--	--	--	----



Observações

- Se o usuário digitar mais de 1000 elementos em um array de 1000 elementos, o programa tentará ler normalmente.
- Porém, o programa os armazenará em uma parte não reservada de memória, pois o espaço reservado para o array foi para somente 1000 elementos.
- Isto pode resultar nos mais variados erros durante a execução do programa.



Vetores

Cada elemento do array tem todas as características de uma variável e pode aparecer em expressões e atribuições (respeitando os seus tipos)

- `velocidades[2] = 10 + velocidades[3];`
- `if (velocidades[2] > 60)`
- Ex: somar todos os elementos de velocidades:

```
int soma = 0;  
int i = 0;  
for (i = 0; i < 1000; i++) {  
    soma = soma + velocidades[i];  
}
```



Exemplo

Podemos usar um comando de repetição (for, while e do-while) para percorrer um array

- Exemplo: somando os elementos de um array de 5 elementos

```
#include <stdio.h>

int main() {
    int velocidades[4] = {20, 55, 22, 33};

    for (int i = 0; i < 4; i++) {
        printf("%d\n", velocidades[i]);
    }
}
```



Vetores (operação)

- Cada elemento do array tem todas as características de uma variável e pode aparecer em expressões e atribuições (respeitando os seus tipos)
- `velocidades[2] <- 10 + velocidades[3]`
- `if (velocidades[2] > 60)`
- Ex: somar todos os elementos de velocidades:



Vetores

Características básicas de um Vetor

- Estrutura homogênea, isto é, é formado por elementos do mesmo tipo.
- Todos os elementos da estrutura são igualmente acessíveis, isto é, o tempo e o tipo de procedimento para acessar qualquer um dos elementos do array são iguais.
- Cada elemento do array tem um índice próprio segundo sua posição no conjunto



Acessando elementos do vetor

- Em um vetor os elementos são acessados especificando o índice desejado entre colchetes []
- DECLARE A[5] NUMÉRICO
- DECLARE N = 2
- A[0] <- 20
- A[2] <- 30
- A[3] <- A[1] + A[2]
- A[4] <- A[3] + A[n]
- A[n] <- A[n] * A[4]



Inicializar um vetor

Pode-se inicializar um vetor utilizando a chave `{ }`

- `float notas[5] = {7, 8, 9.5, 9.9, 5.2};`
- `int notas[10] = {0}; // zera todos os itens`



Exemplos

```
PARA  $i \leftarrow 1$  ATÉ 5 FAÇA
```

```
INÍCIO
```

```
    ESCREVA "Este é o ",  $i$ , "º número do vetor"
```

```
    ESCREVA  $X[i]$ 
```

```
FIM
```



Questão 1. Considere o seguinte trecho de código C:

```
1  #include <stdio.h>
2  int main() {
3      int arr[4] = { 1, 2, 3, 4 };
4      int i = 2;
5      printf("%d", arr[i]);
6  }
```

Pode-se afirmar que a saída do programa na linha 5 é:

- (a) 1
- (b) 2
- (c) 3
- (d) 4
- (e) Erro, o programa não irá executar adequadamente.



```
1  #include <stdio.h>
2
3  int main() {
4      int a[8] = {9, 8, 7, 2, 3, 11, 5, 17};
5      int b = 100;
6      int i = 0;
7      for (i = 0; i < 8; i = i + 2) {
8          if (a[i] < b) {
9              b = a[i];
10         }
11     }
12     printf("%i", b);
13 }
```

Questão 1. Pode-se afirmar que a saída do programa na linha 12 é:

- (a) 9
- (b) 7
- (c) 2
- (d) 3
- (e) 5



Exercício

- O professor Arquimedes precisa da sua ajuda para descobrir qual é a nota mais frequente entre as notas que os alunos dele tiraram na última prova. A turma tem N alunos e seu programa deve imprimir a nota que aparece mais vezes na lista de N notas. Se houver mais de uma nota mais frequente, você deve imprimir a maior delas! Por exemplo, se a turma tiver $N = 10$ alunos e as notas forem $[20, 25, 85, 40, 25, 90, 25, 40, 55, 40]$, as notas mais frequentes são 25 e 40, ocorrendo três vezes cada. Seu programa, então, deve imprimir 40.



- **Entrada**
- A entrada consiste de duas linhas. A primeira linha contém um número inteiro N , o número de alunos na turma. A segunda linha contém N inteiros, que é a lista de notas dos alunos.
- **Saída**
- Seu programa deve imprimir apenas uma linha contendo apenas um número, a nota mais frequente da lista.





Exemplo de Entrada	Exemplo de Saída
10 20 25 85 40 25 90 25 40 55 40	40



Exercício



- Certa vez, numa aula, a professora passou um filme para os alunos assistirem. Durante este filme, ela passou uma lista de presença em sua sala para verificar a presença dos alunos, onde cada aluno deveria inserir apenas seu número de registro. Alguns alunos contudo, como possuem amigos que fogem da aula, decidiram ser camaradas e inseriram os números de registro de seus amigos fujões. O problema é que muitos alunos são amigos de alunos que fogem da aula e alguns números de registro acabaram sendo repetidamente inseridos na lista de presença. Além de tudo, alguns dos alunos que se esperava que não estivessem na aula de fato estavam!
- A professora, ao notar que a lista de presença continha alguns números repetidos, ficou sem entender, mas decidiu dar um voto de confiança e dar presença a todos os alunos cujos números de registro estavam na lista. Como são muitos alunos na sala e muitos números com repetição, ela pediu a sua ajuda para determinar o total de alunos que receberam presença na aula.



■ Entrada

- A primeira linha da entrada contém um número inteiro N ($1 \leq N \leq 105$), que informa a quantidade de números de registro que apareceram na lista de presença. Cada uma das N linhas seguintes contém um número de registro V_i ($0 \leq V_i \leq 106$) que foi inserido na lista de presença.

■ Saída

- Seu programa deve imprimir uma única linha, contendo apenas um número inteiro, o número de alunos que receberam presença.





Exemplos de Entrada

Exemplos de Saída

3
2
3
1

3

7
0
5
12
41
7
5
41

5



Obrigado!

Perguntas?

Marcos Roriz (marcosroriz@ufg.br)

**ENGENHARIA DE
TRANSPORTES**

FCT
FACULDADE DE
CIÊNCIAS E TECNOLOGIA



UFG
UNIVERSIDADE
FEDERAL DE GOIÁS

Fonte dos ícones utilizados: flaticons.com



Obrigado!

Perguntas?

Marcos Roriz (marcosroriz@ufg.br)

**ENGENHARIA DE
TRANSPORTES**

FCT
FACULDADE DE
CIÊNCIAS E TECNOLOGIA



UFG
UNIVERSIDADE
FEDERAL DE GOIÁS

