



Vetor

Prof. Lilian Berton

São José dos Campos, 2019

Vetores

- São estruturas indexadas utilizadas para armazenar dados de um mesmo tipo: int, char, float ou double. O exemplo a seguir é de um vetor de inteiros:

0	1	2	3	4	5		78	79
10	12	17	12	-4	-3	...	-42	34

- Declaração de vetores:
- O exemplo abaixo reserva 80 gavetas consecutivas na memória, que corresponde ao tamanho ou número de casas do vetor. Cada gaveta guarda um int.

`<tipo_do_vetor> <nome_do_vetor> [<tamanho_do_vetor>];`

Exemplos:

- `int v[80];` ————— 80 é o tamanho do vetor!

Vetores

- São usados índices para acessar uma casa de um vetor.
- Um índice é um número natural.
- O índice da primeira casa é sempre zero.

```
1      # include <stdio.h>
2
3      int main () {
4          int v[80], i;
5
6          v[3] = 4; /* casa de índice 3 do vetor v recebe o inteiro 4 */
7          i = 2;
8          v[i] = 3; /* casa de índice 2 do vetor v recebe o inteiro 3 */
9          v[v[i]] = 10; /* vc saberia dizer qual casa do vetor v
10                        * recebe o inteiro 10?
11                        */
12
13      return 0;
14  }
```

0	1	2	3	4	5		78	79
?	?	3	4	?	?	...	?	?

Percorrendo vetores

- **Percorrer um vetor significa varrer o vetor de casa em casa a partir do índice 0 (zero).**
- Ao percorrer um vetor, é necessário saber o número de casas que deve-se percorrer.
- Este número normalmente é guardado em uma variável inteira.

```
1      # include <stdio.h>
2
3      int main () {
4          float x[80];
5          int i;
6
7          for (i=0; i<80; i++)
8              x[i] = 0;
9
10         return 0;
11     }
```

Leitura de vetores

```
1  # include <stdio.h>
2
3  int main () {
4      float v[100];
5      int i, n;
6
7      printf ("Entre com 0<n<=100: ");
8      scanf ("%d" &n);
9
10     /* percorrer o vetor v de 0 a n-1 colocando o valor lido pelo teclado */
11     for (i=0; i<n; i++) {
12         printf ("Entre com v[%d] = ", i);
13         scanf ("%f", &v[i]);
14     }
15
16     return 0;
17 }
```

← Armazena valores no vetor

Impressão de vetores

```
1  # include <stdio.h>
2
3  int main () {
4      float v[100];
5      int i, n;
6
7      printf ("Entre com 0<n<=100: ");
8      scanf ("%d" &n);
9
10     /* percorrer o vetor v de 0 a n-1 imprimindo o valor de cada casa */
11     for (i=0; i<n; i++) {
12         printf ("v[%d] = %f\n", i, v[i]);
13     }
14
15     return 0;
16 }
```

← Imprime os valores
armazenados no vetor

Observações

- Ao desenvolver seus programas com vetores, preste atenção com relação aos seguintes detalhes:
- **índices inválidos:** tome muito cuidado, especialmente dentro de um while ou for, de não utilizar índices negativos ou maiores que o tamanho máximo do vetor.
- **definição do tamanho do vetor:** se faz na declaração do vetor. O tamanho dele é constante, só mudando a sua declaração é que podemos alterar o seu tamanho. Isso significa que podemos estar “desperdiçando” algum espaço da memória que fica no final do vetor.
- Não leia n, onde n seria o tamanho do vetor, e tente “declarar” o vetor em seguida.

Busca sequencial em vetores

- Procurar um valor numa lista ou vetor:
pesquisa 5 em [8,2,1,5,2,6] → True
pesquisa 5 em [8,2,1,3,2,6] → False

x é comparado sucessivamente com
 $a[0], a[1], \dots, a[n-1]$.

Se x for igual a algum dos $a[i]$, retorna-se True.
Senão, retorna-se False.


```

#include<stdio.h>
bool buscaSequencial (int x, int v[]);

int main ()
{
    int v[10], i, x;
    bool encontra;

    for(i = 0; i < 10; i++) {
        scanf("%d",&v[i]);
    }
    printf("Insira o numero que deseja procurar no vetor.\n");
    scanf("%d",&x);
    encontra = buscaSequencial(x,v);

    printf(encontra ? "true" : "false");

    return 0;
}

bool buscaSequencial (int x, int v[]) {
    int i;
    for(i = 0; i < 10; i++) {
        if(v[i] == x) {
            return true;
            break;
        }
    }
    return false;
}

```

Busca em lista telefônica

Como buscamos o número de telefone de uma pessoa em uma lista?



Busca binária em vetores

- Busca o elemento 10 num vetor de 0 a 14:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0...14	•	•	•	•	•	•	•	<u>•</u>	•	•	•	•	•	•	•
8...14	•	•	•	•	•	•	•	•	•	•	•	<u>•</u>	•	•	•
8...10	•	•	•	•	•	•	•	•	•	<u>•</u>	•	•	•	•	•
9...9	•	•	•	•	•	•	•	•	•	•	<u>•</u>	•	•	•	•
9...8 = \emptyset	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•

- O número de divisões ao “meio” é igual ao número de comparações, neste caso, igual a $4 = \log(n + 1)$.

Busca binária em vetores

- **Se a lista a está ordenada** – suponhamos por ordem crescente – há algoritmos muito mais eficientes de procurar x:
- x é procurado entre os índices $i1$ e $i2$.
- Inicialmente faz-se $i1=0$ e $i2=n-1$.
- Repetidamente, calcula-se o ponto médio $m=(i1+i2)/2$ (divisão inteira)
 - Se $x < V[m] \rightarrow i2=m-1$;
 - Se $x > V[m] \rightarrow i1=m+1$
 - Se $x == V[m] \rightarrow$ retorna true
- Se o intervalo $[i1,i2]$ acaba por ficar vazio ($i1 > i2$)
 - x não está na lista: \rightarrow retorna false.

```

int main () {
    int v[10], i, x;
    bool encontra;

    for(i = 0; i < 10; i++) {
        scanf("%d",&v[i]);
    }
    printf("Insira o numero que deseja procurar no vetor.\n");
    scanf("%d",&x);
    ordenaVetor(v);
    encontra = buscaBinaria(x,v);
    printf(encontra ? "true" : "false");

    return 0;
}

bool buscaBinaria (int x, int v[]) {
    int i1 = 0, i2 = 9, m;
    while (i1 <= i2) {
        m = (i1+i2)/2;
        if (x < v[m]) {
            i2 = m-1;
        } else if (x > v[m]) {
            i1 = m+1;
        } else {
            return true;
        }
    }
    return false;
}

```

Ordenação de elementos

- **Ordenar:** processo de rearranjar um conjunto de objetos em uma ordem ascendente ou decendente.
- A ordenação **visa facilitar a recuperação** posterior de itens do conjunto ordenado.



1	22	5	9	3	15	8	14	10	19
---	----	---	---	---	----	---	----	----	----

1	3	5	8	9	10	14	15	19	22
---	---	---	---	---	----	----	----	----	----

Exercícios

1. Leia dois vetores A e B (de tamanho 10 cada um). Depois faça duas funções que calcule:
 - a) a soma dos elementos do vetor A com os do vetor B (respeitando as mesmas posições) e escrever o resultado em um vetor Soma.
 - b) a subtração dos elementos do vetor A com os do vetor B (respeitando as mesmas posições) e escrever o resultado em um vetor Subtracao.

2. Faça um algoritmo para ler e armazenar em um vetor a temperatura média de todos os dias da semana. Calcule:
 - a) Menor temperatura da semana
 - b) Maior temperatura da semana
 - c) Temperatura média semanal

3. Faça um algoritmo para ordenar os elementos de um vetor.



$$\sum_{i=1}^n x_i$$



```
for (int i=0; i<n; i++)  
    sum+=X[i];
```