ESTRUTURAS DE DADOS

VARIÁVEIS COMPOSTAS

- VETORES

Variáveis Compostas

- São um conjunto de variáveis identificadas por um mesmo nome.
  - Homogêneas (vetores e matrizes)
  - Heterogêneas (estruturas)

Arranjos unidimensionais

- Utilizados para armazenar conjuntos de dados cujos elementos podem ser endereçados por um único índice.

- Também são conhecidos como vetores.

Variáveis Compostas Homogêneas

- Correspondem a posições da memória:
  - identificadas por um único nome
  - individualizadas por índices
  - cujo conteúdo é de um mesmo tipo

Notas:	6,1	2,3	9,4	5,1	8,9	9,8	10	7,0	6,3	4,4
Posição:	0	1	2	3	4	5	6	7	8	9

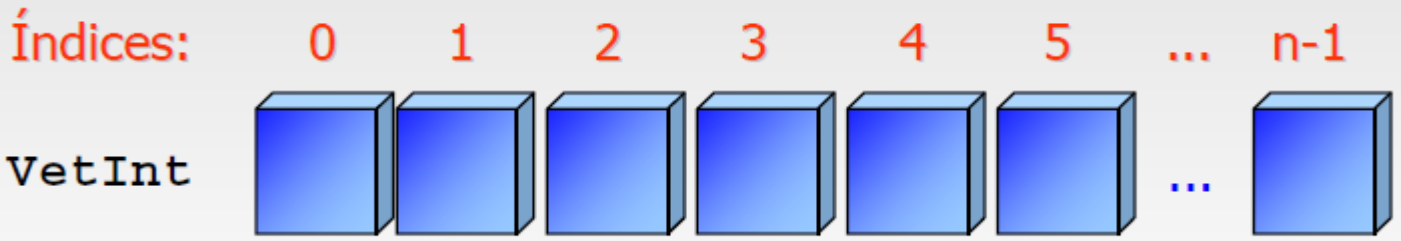
Vetores e Matrizes na Linguagem C

- O tamanho de um vetor ou matriz é pré-definido, ou seja, após a compilação, não pode ser mudado.

- Portanto, vetores e matrizes são chamadas estruturas de dados estáticas, pois mantém o mesmo tamanho ao longo de toda a execução do programa.

Vetores e Matrizes na Linguagem C

```
int VetInt[n];
```



Índice do primeiro elemento: zero
Índice do último elemento: n – 1
Quantidade de elementos: n

Declaração de um vetor

Tipo de dado <nome_da_variável> [tamanho];

Ex.:

```
int M [ 10] ;
char  nome[15];
float  notas[30]
```

- Atribuição de dados a um Vetor

No caso de vetores (variáveis indexadas), além do nome da variável deve-se necessariamente fornecer também o índice do componente do vetor onde será armazenado o resultado da avaliação da expressão. Exemplo:

M[0]←15
M[1]←13
M[2]←10
……
M[10]←35

- Operações Básicas com Vetores

O acesso individual a cada componente de um vetor é realizado pela especificação de sua posição na mesma por meio do seu índice. No exemplo anterior foi definida uma variável M capaz de armazenar 10 número inteiros. Para acessar um elemento deste vetor deve-se fornecer o nome do mesmo e o índice do componente desejado do vetor (um número de 1 a 10, neste caso).

Por exemplo:

float M [10];

7.5	8.2	5.5	9.0	8.3	4.5	7.7	3.0	6.5	8.0
0	1	2	3	4	5	6	7	8	9

M[0] indica o primeiro elemento do vetor,
M[1] indica o segundo elemento do vetor,
.....
.....
M[9] indica o último elemento do vetor.

- Leitura de Dados de um Tipo Vetor

```
int M [10], i ;

    for (i=0; i<10; i++)
      scanf("%i",&M[i]);
```

- Escrita de Dados de um Vetor

.....

```
for (p=0; p<10; p++)
 printf("\n O elemento %i foi: %i \n", p,  numeros[ p]);
```

.....

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

main()
{
 int i, j, p, veta[10],   vetb[10],    vetc[10];

 // Faz a entrada do vetor A
  printf("\n Entre com os elementos do vetor A.\n\n");
 for (i=0; i<10; i++)
 {
  printf("Entre com o %i elemento do vetor A : " ,i);
  scanf("%i",&veta[i]);
 }
    // Faz a entrada do vetor B
printf("\n Entre com os elementos do vetor B.\n\n");
 for (i=0; i<10; i++)
 {
  printf("Entre com o %i elemento do vetor B: ",i);
  scanf("%i",&vetb[i]);
   }
   // calcula a soma dos dois vetores
  for (p=0; p<10; p++)

  vetc[p]=veta[p]+vetb[p];

getche();
}
```

Pode-se utilizar um comando while

```
 int v[50];
i = 0;
printf("Digite os elementos da relação (máximo = 50 e -1 para encerrar)");

scanf("%d", &v[i]);
while (v[i] != -1)
  { i=i+1;
   scanf("%d", &v[i]);
  }
```