Vetores e Strings

1. Vetores

Vetores nada mais são que matrizes.

Matriz (Definição da Álgebra) -> Arranjo retangular de elementos de um conjunto.

É importante notar que vetores, ou melhor, matrizes de qualquer dimensão são caracterizadas por terem todos os elementos pertencentes ao mesmo tipo de dado. Forma geral para se declarar um vetor unidimensional:

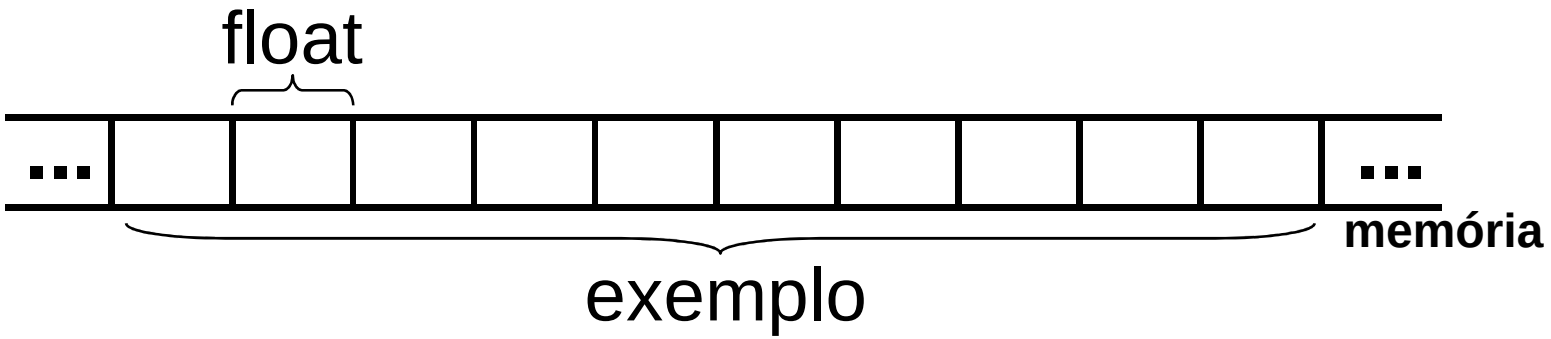
tipo_da_variavel nome_da_variavel [tamanho];

Vetores e Strings

1. Vetores (continuação)

Exemplo:

float exemplo [10];



Vetores e Strings

1. Vetores (continuação)

Na linguagem C a numeração dos índices começa sempre em zero. Isto significa que, no exemplo acima, os dados serão indexados de 0 a 9. Para acessá-los vamos escrever:

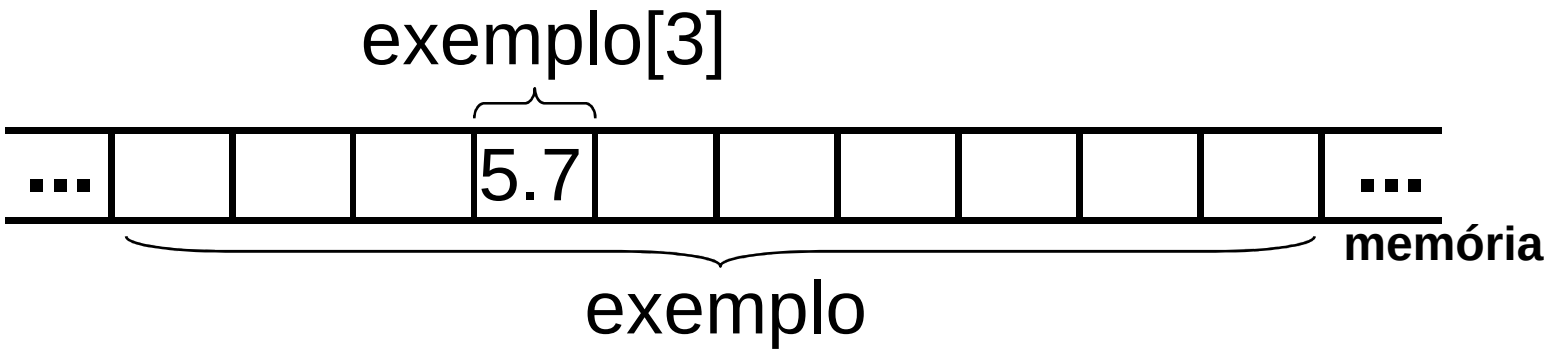
exemplo[**0**]
exemplo[**1**]
.
.
.
exemplo[**9**]

Vetores e Strings

1. Vetores (continuação)

Exemplo:

exemplo [3]=5.7;



Vetores e Strings

1. Vetores (continuação)

Mas ninguém o impede (programador) de escrever:

exemplo[30]

exemplo[103]

exemplo[-2]

Pois, o compilador C não verifica se o índice que você usou está dentro dos limites válidos. **Este é um cuidado que *você* deve tomar.** Se o programador não tiver atenção com os limites de validade para os índices ele corre o risco de ter variáveis sobreescritas ou de ver o computador travar. Inúmeros *bugs* podem acontecer.

Vetores e Strings

1. Vetores (continuação)

Exercício:

Construa um programa que declare um vetor de inteiros com 10 elementos e o inicialize com números fornecidos pelo usuário, através da entrada padrão.

```
#include <stdio.h>
main()
{
   int vetor[10], indice;
   for (indice=0; indice<10; indice++)
   {
      printf("\nVetor[%d]: ",indice+1);
      scanf("%d",&vetor[indice]);
   }
}
```

1. Vetores (continuação)

Exercício:

Construa um programa, com base no exercício anterior, que declare um vetor de inteiros com 10 elementos, o inicialize, com números fornecidos pelo usuário através da entrada padrão, e que através de uma pesquisa nos elementos do vetor, retorne na saída padrão os elementos de menor e maior valor, respectivamente.

```
#include <stdio.h>
main()
{
   int vetor[10], indice, menor, maior;
   for (indice=0; indice<10; indice++)
   {
      printf("\nVetor[%d]: ",indice+1);
      scanf("%d",&vetor[indice]);
   }
   for (indice=0;indice<10;indice++)
      if (!indice)
         menor=maior=vetor[indice];
      else
         if (menor>vetor[indice])
            menor=vetor[indice];
         else
            if (maior<vetor[indice])
               maior=vetor[indice];
   printf("\nO menor valor dos elementos do vetor eh: %d",menor);
   printf("\nO maior valor dos elementos do vetor eh: %d",maior);
}
```

```
#include <stdio.h>
main()
{
   int vetor[10], indice, menor, maior;
   for (indice=0; indice<10; indice++)
   {
      printf("\nVetor[%d]: ",indice+1);
      scanf("%d",&vetor[indice]);
   }
 menor=maior=vetor[0];
   for (indice=1;indice<10;indice++)
      if (menor>vetor[indice])
         menor=vetor[indice];
      else
         if (maior<vetor[indice])
            maior=vetor[indice];
   printf("\nO menor valor dos elementos do vetor eh: %d",menor);
   printf("\nO maior valor dos elementos do vetor eh: %d",maior);
}
```

Vetores e Strings

1. Vetores (continuação)

OBS.: Um vetor pode ser inicializado na declaração, exemplo:

```
int vetor[10]={0,1,2,3,4,5,6,7,8,9};
```

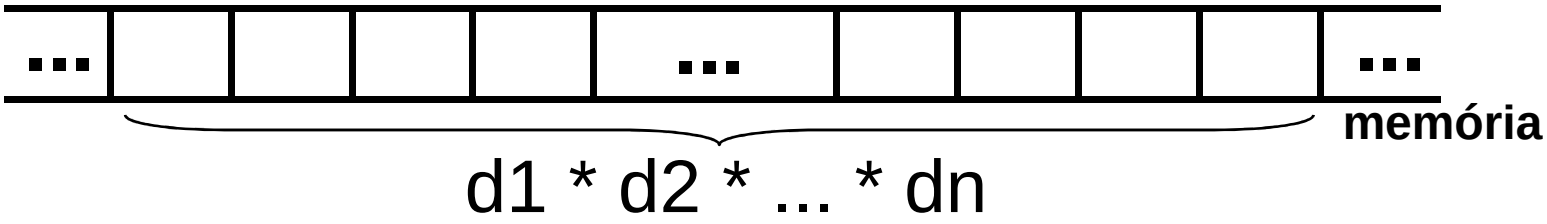
E ainda pode-se deixar em aberto o número de elementos, o qual será preenchido pelo número de valores fornecidos no momento da declaração, ou melhor, durante a inicialização.

```
int vetor[]={0,1,2,3,4,5,6,7,8,9};
```

```
int vetor[];  /* Não é permitido! */
```

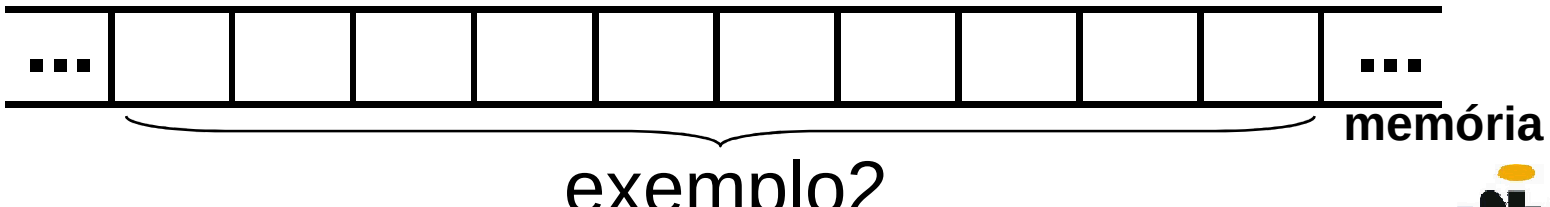
2. Vetores Multidimensionais

tipo_da_variavel nome_da_variavel [d1][d2]...[dn];



Exemplo:

int exemplo2 [2][5];

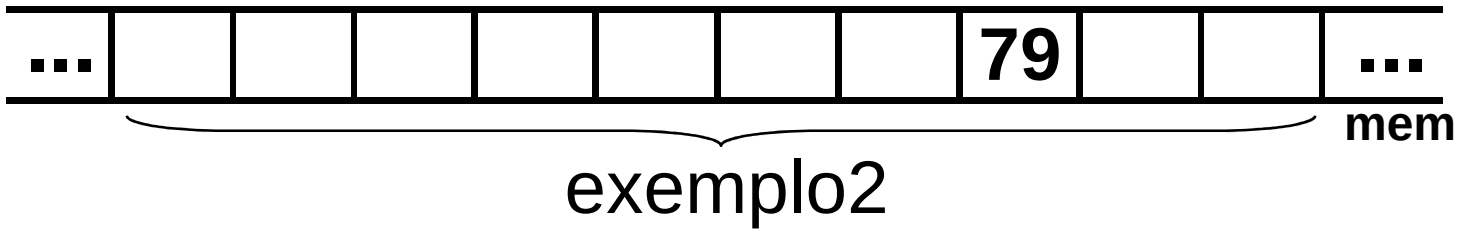


Vetores e Strings

Exemplo:

exemplo2 [1][2]=79;

O armazenamento de vetores multidimensionais se dá da seguinte forma: na primeira posição armazena-se o elemento com todos os índices zero, no caso do exemplo acima o elemento referenciado por exemplo2[0][0], seu sucessor é o elemento com o índice mais à direita incrementado em uma unidade (exemplo2[0][1]); quando o referido índice chegar ao seu valor máximo (exemplo2[0][4]) é incrementado o índice que o antecede e o seu valor é zerado (exemplo2[1][0]) e assim sucessivamente. Sendo assim, temos



Vetores e Strings

Exemplo:

O programa a baixo declara uma matriz 3x4 de inteiros e a inicializa com valores fornecidos pelo usuário.

```
#include <stdio.h>
#define n_l 3
#define n_c 4
main()
{
   int matriz[n_l][n_c], i, j;
   for (i=0;i<n_l;i++)
      for (j=0;j<n_c;j++)
      {
         printf ("\nEntre com matriz[%d][%d]=",i+1,j+1);
         scanf ("%d",&matriz[i][j]);
      }
}
```

2. Vetores Multidimensionais (continuação)

Assim como os vetores unidimensionais os vetores multidimensionais também podem ser inicializados na declaração.

Exemplo:

float matriz [3][4]={1,2,3,4,5,6,7,8,9,10,11,12};

float matriz [ ][2]={1,2,3,4,5,6,7,8,9,10,11,12};

/* Uma dimensão pode ser omitida quando a inicialização se dá na declaração */

~~*float matriz [ ][ ]={1,2,3,4,5,6,7,8,9,10,11,12};*~~

/* Mais de uma dimensão não podem ser omitidas quando a inicialização se dá na declaração */

2. Vetores Multidimensionais (continuação)

Exercício: Construa um programa, na linguagem C, que declare uma matriz 7x4 de números em ponto flutuante, a inicialize com valores fornecidos pelo usuário através da entrada padrão e a retorne na saída padrão com o layout a seguir:

	x.xx	x.xx	...	x.xx
10	x.xx	10 x.xx	...	x.xx
	.	.	.	.
	.	.	.	.
	.	.	.	.