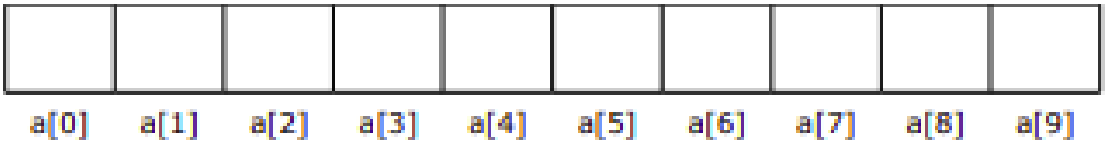
Vetores

Um vetor é um conjunto finito de elementos homogêneos. O conjunto é organizado de forma que exista um elemento zero, um primeiro elemento, um segundo e assim por diante. A declaração de um vetor unidimensional é feita da mesma forma que uma variável simples, e segue a sintaxe:

```
tipo nome_vetor [nro_de_elementos];
```

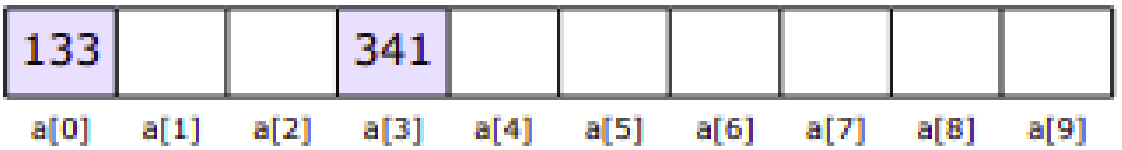
Assim, a declaração em C especifica um vetor de 10 inteiros:

```
int a[10];
```



Em C, os índices de um vetor com n elementos variam de 0 a n-1, sendo que o índice do primeiro elemento é sempre 0 (zero). Os índices podem ser utilizados para ler e armazenar dados no vetor:

```
int a [10];
a[0] = 133;
a[3] = 341;
```



As duas operações básicas para acesso aos elementos de um vetor são a **extração** e o **armazenamento**, e ambas utilizam um índice para localizar uma posição no vetor. O trecho de programa abaixo exemplifica as operações de declaração e atribuição do valor 0 à todas as posições do vetor, e em seguida imprime os valores armazenados:

```
#include <stdio.h>
int main(){
      int i, a[100];
      for (i=0;i<100;i++) // preenche o vetor com zeros
       {
            a[i]= 0;
       }
       for (i=0;i<100;i++) // impressão do vetor
       {
            printf("%d ",a[i]);
       }
   return 0;
}
```

Um vetor unidimensional é normalmente utilizado para manter uma grande quantidade de itens na memória, e para referenciar todos os itens de maneira uniforme. Uma declaração:

```
int vetor [100];
```

indica que são reservadas 100 posições sucessivas de memória, cada uma com tamanho definido para armazenar um número inteiro.

O trecho de programa abaixo exemplifica a geração aleatória de valores entre 0 e 10 a serem armazenados em um vetor de números reais, de tamanho 100, e o cálculo da média dos valores gerados:

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

int main(){
   float a[100];
   float soma=0.0;
   float media=0.0;
   int i;
   srand(10);
   //srand (time(NULL));
   for (i=0;i<100;i++){
       a[i]=rand()%11; // Gera número de 0 a 10 e armazena no vetor a
   }
   for (i=0;i<100;i++){
```

```
        soma += a[i];
    }
    media = soma/100;
    printf("Media = %.2f", media);
}
```

Pode-se declarar um vetor sem indicar a quantidade de elementos a ser reservada, desde que os elementos a serem armazenados estejam definidos na sua carga inicial. Neste caso, o compilador calcula o número de elementos que o vetor terá.

```
int a[] = {13,24,35,46};
```

Observação: Não se pode declarar vetores sem dimensão ou sem a dimensão calculada a partir da carga automática.

Exercício:

1- Faça um programa para gerar um vetor de inteiros randômicos com tamanho M. Implemente um menu com as seguintes opções para:
1- exibir os valores do vetor;
2- encontrar o maior valor deste vetor;
3- encontrar o maior e o menor valor;
4- calcular a média dos valores do vetor;
5- verificar se existem dois valores iguais no vetor retornando **verdadeiro** ou **falso**.

2- Faça um programa para trocar de posição dois elementos de um vetor inteiro usando uma variável auxiliar.

3- Para um vetor de N números reais, imprimir seus elementos na ordem inversa.

4- Ler 20 notas de uma determinada turma, calcular a média, mostrar as notas que estão acima da média (>7), na média e quais notas estão abaixo da média.

```
char nome[30];
printf("Digite o nome da turma: ");
 scanf("%s",nome);
 fflush(stdin);
```

Configuração da resposta:
Turma xxxxxxx
****

Notas abaixo da média:
x
y
z
w
****
Notas na média:
v
a
c
b
****
Notas acima média:
t
e
f
g

Matrizes

O que é uma matriz?

- Matriz é uma estrutura de dados do tipo vetor com duas ou mais dimensões.
- No curso, vamos trabalhar com duas dimensões, e chamaremos de Matriz bi-dimensional.
- Os itens de uma matriz tem que ser todos do mesmo tipo de dado.
- Na prática, as matrizes formam tabelas na memória.

Matriz na linguagem C
Exemplo de declaração de matriz com 2 dimensões usando linguagem C
float Media[2][3];
Onde:

- O valor 2 representa a quantidade de linhas.
- O valor 3 representa a quantidade de colunas.

Dizemos que esta matriz é do tipo 2 X 3.

Como temos 2 linhas com 3 posições de armazenamento em cada linha, temos capacidade para armazenar até 6 elementos (itens) do tipo float.

Será necessário utilizar um índice para cada dimensão da matriz, logo uma matriz bidimensional terá 2 índices, um para posicionar a linha, outro para a coluna e assim como no vetor, o índice da primeira posição é zero.

Como atribuir valores a uma matriz?

Suponha a matriz como declarada:

float Media[3] [2];

para atribuir uma valor precisamos identificar a posição usando os índices:

Media [0][0] = 5.0;

Preenchendo uma matriz bi-dimensional

Para fazer o preenchimento de uma matriz, devemos percorrer todos os seus elementos e atribuir-lhes um valor.

Isto pode ser feito tanto gerando valores para cada elemento da matriz, como recebendo os valores pelo teclado.

Um método interessante para percorrer uma matriz é usar duas estruturas de repetição for e duas variáveis inteiras, uma para a linha e a outra para a coluna.

Exemplo, preenchendo matriz com valores gerados.

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

int main(){
  int matriz[3][3];
  int l,c;
  for ( l=0; l < 3; l++ )
  {
          for ( c=0; c<3; c++ )
          {
                  matriz[l][c]= 2*l+c;
          }
  }//fim for
 for ( l=0; l<3; l++ )
  {
        for ( c=0; c<3; c++ )
        {
              printf("%d ", matriz[l][c]);
        }
          printf("\n");
  }//fim for
return 0;
}
```

Exemplo, preenchendo matriz com valores digitados pelo teclado:

```
#include <stdio.h>
#include <stdlib.h>
int main(){
 int qtlinha;
int qtcoluna;
printf(“digite a quantidade de linha:”);
scanf(“%d”, &qtdlinha);

printf(“digite a quantidade de coluna:”);
scanf(“%d”, &qtdcoluna);
  int matriz[qtdlinha][qtdcoluna];

  int l,c;
  for ( l=0; l<3; l++ )
  {
         for ( c=0; c<3; c++ )
         {
                  printf("digite o valor da posicao [%d][%d]:",l, c);
                  scanf("%d", &matriz[l][c]);
         }
  }//fim for
 for ( l=0; l<3; l++ )
  {
       for ( c=0; c<3; c++ )
       {
              printf("%d ", matriz[l][c]);
       }
          printf("\n");
  }
return 0;
}
```

Exercícios

1- Elabore um programa em C que leia duas matrizes 2x2 e as imprima na tela.

2- Elabore um programa em C que leia uma matriz 2x3 de inteiros, mostre na tela a matriz formatada preservando o número de linhas e colunas. Posteriormente, multiplique cada posição por um valor inteiro B (positivo ou negativo) informado pelo usuário via teclado, e imprima a matriz com os novos valores.

3- Seja A e B duas matrizes 3x3. Implemente um programa em c que calcule e mostre na tela a soma, subtração e multiplicação da matriz A por B.

$$A = \begin{bmatrix} 2 & 4 & 5 \\ 1 & 7 & -4 \\ 3 & 1 & 7 \end{bmatrix} \qquad B = \begin{bmatrix} 5 & 3 & 2 \\ 2 & 4 & 1 \\ 9 & 8 & 6 \end{bmatrix}$$

Dica: Crie uma matriz 4x3 auxiliar para realizar os cálculos, tanto adição, multiplicação e subtração terá a mesma nomenclatura, ou seja, 4 linhas e 3 colunas.

4- Elabore um programa C, que leia 3 matrizes A, B e C 3x3 quaisquer. Na sequência, o programa deve mostrar as matrizes informadas na tela, assim como deve também o resultado dos seguintes cálculos em função do usuário.

1) A + B + C
2) A + B -C
3) A - B - C

4- Verifique uma matriz **A** 5 x 4 quaisquer, se existe elementos que se repetem. Caso encontre um elemento repetido, informar quantas vezes o elemento se repetiu.