Arrays

Array

- Coleção de variáveis do mesmo tipo que é referenciada por um nome comum.
- O elemento específico de um array é acessado por um índice
- São posições contíguas na memória
- O endereço mais baixo corresponde ao primeiro elemento e o mais alto ao último elemento.

Array

- Arrays em linguagem de alto nível especifica:
  - Nome do array
  - número de elementos
  - tamanho do elemento
  - tipo do elemento
  - faixa de índices

Array

- Exemplo em C:

```
int     test_marks [10];
```

  - Nome do array (test_marks)
  - número de elementos (10)
  - tamanho do elemento (4 bytes)
  - tipo do elemento (int)
  - faixa de índices (0 a 9)
- O espaço necessário para armazenar o array:

  Espaço = numero elementos * tamanho em bytes

- No exemplo **espaço = 10 * 4 = 40 bytes**

Array

- Em assembly, arrays são implementados através da alocação do requerido espaço de armazenamento.
- O array do exemplo anterior pode ser declarado desta forma:

```
test_marks      resd    10
```

Array

- Para acessar um elemento de um array em assembly:

Deslocamento = númro do índice * tamanho em bytes

- Para acessar test_marks[5]:

Deslocamento = 5 * 4 = 20

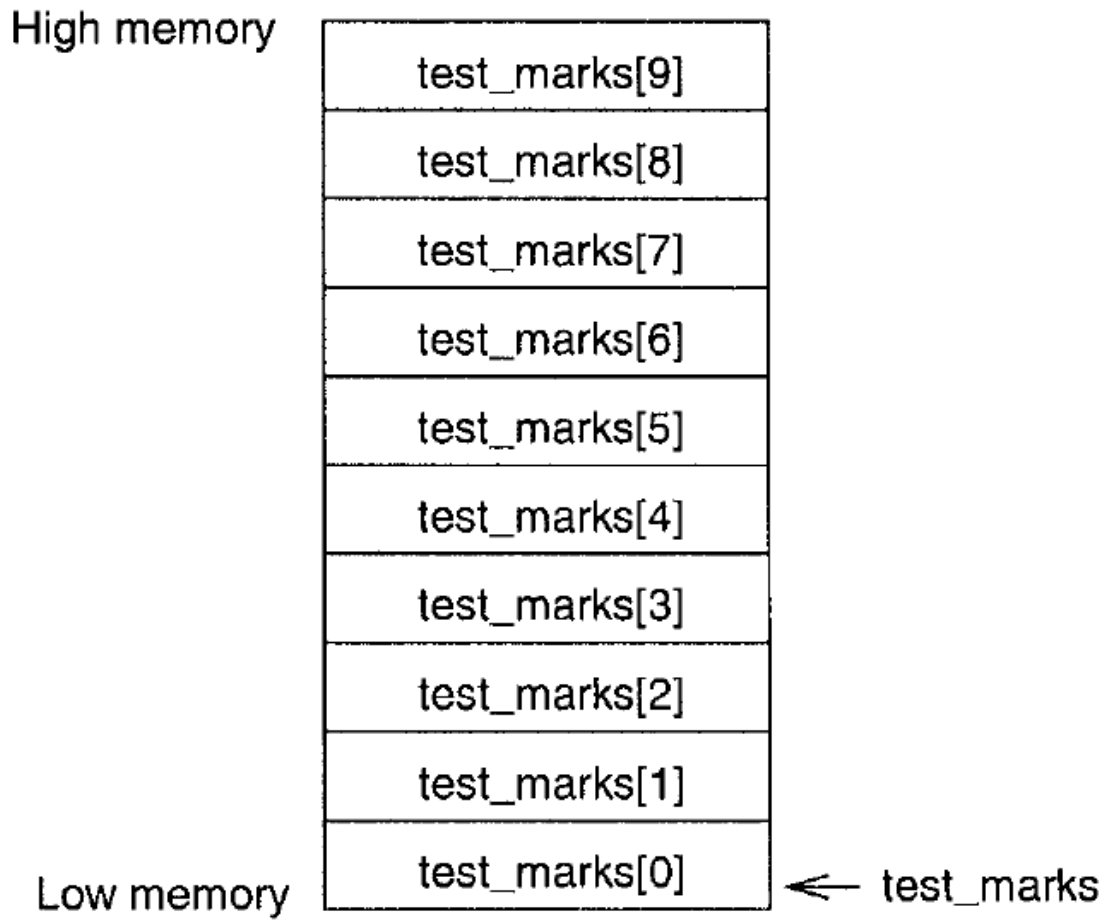
Declarações de Array

- Z DD 1, 2, 3 ; declara array com 3 elementos de 4 bytes cada
- bytes DB 10 DUP(?) ; declara 10 bytes não inicializados
- Arr DD 100 DUP(0) ; declara 100 palavras inicializadas por zero.
- Str DB 'hello',0 ; declara 6 bytes

Declarações de Arrays

- Arr resb 10 ;declara um array de 10 bytes
- Arr2 times 5 db 10 ; declara array de 5 bytes e os inicializa com o valor 10
- Arr3 times 10 db 1 ; declara array de 10 bytes e os inicializa com valor 1

Armazenando array na memória



Exemplo

- Soma de todos os elementos de um array:

```
.DATA
test_marks      DD  90,50,70,94,81,40,67,55,60,73
NO_STUDENTS     EQU ($-test_marks)/4      ; number of students
sum_msg         DB  'The sum of test marks is: ',0

.CODE
        .STARTUP
        mov     CX,NO_STUDENTS    ; loop iteration count
        sub     EAX,EAX           ; sum := 0
        sub     ESI,ESI           ; array index := 0
add_loop:
        mov     EBX,[test_marks+ESI*4]
        PutLInt EBX
        nwln
        add     EAX,[test_marks+ESI*4]
        inc     ESI
        loop    add_loop

        PutStr  sum_msg
        PutLInt EAX
        nwln
```

Array multidimensional

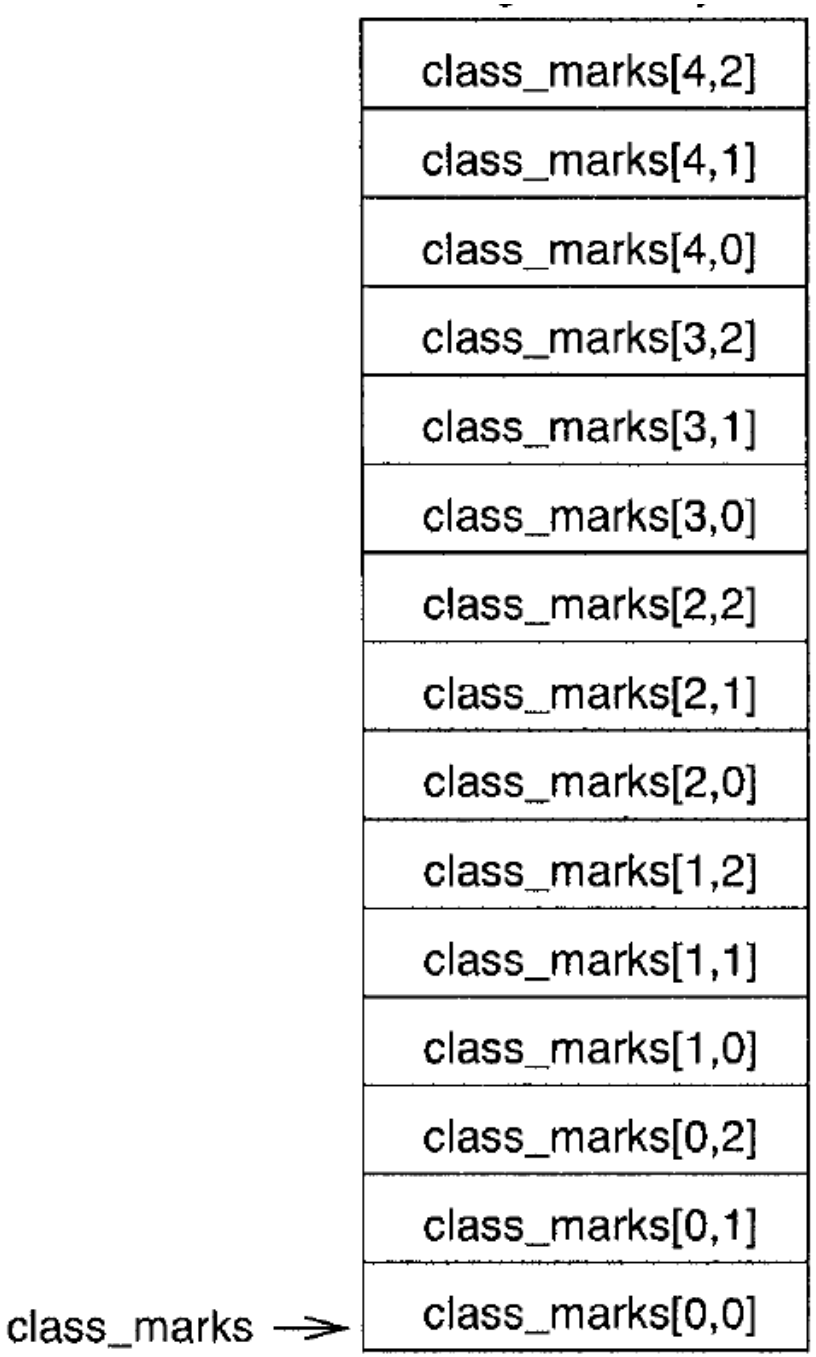
- Podemos criar um array de duas dimensões em assembly da seguinte forma:

```
class_marks     resd     5*3
```

- Cria um array de 60 bytes
- Abstrai o array multidimensional através de linhas *i* e colunas *j.*
- Desloca através da formula abaixo:

Deslocamento = (i * colunas + j) * tamanho

Array multidimensional na memória



Exemplo: encontra a soma dos elementos de uma coluna de um array multidimensional

Exemplo

```
.DATA
NO_ROWS          EQU  5
NO_COLUMNS       EQU  3
NO_ROW_BYTES     EQU  NO_COLUMNS * 2  ; number of bytes per row
class_marks      dw   90,89,99
                 dw   79,66,70
                 dw   70,60,77
                 dw   60,55,68
                 dw   51,59,57

sum_msg          db   "The sum of the last test marks is: ",0

.CODE
         .STARTUP
         mov      CX,NO_ROWS    ; loop iteration count
         sub      AX,AX         ; sum = 0
         ; ESI = index of class_marks[0,2]
         sub      EBX,EBX
         mov      ESI,NO_COLUMNS-1
sum_loop:
         add      AX,[class_marks+EBX+ESI*2]
         add      EBX,NO_ROW_BYTES
         loop     sum_loop

         PutStr   sum_msg
         PutInt   AX
         nwln
```

Exercício

- Dada uma seqüência de n números, imprimi-la na ordem inversa à da leitura.
- Dado dois arrays, A (5 elementos) e B (5 elementos), faça um programa em assembly que imprima todos os elementos comuns aos dois arrays.
- Dado dois arrays, A (5 elementos) e B (8 elementos), faça um programa em assembly que imprima todos os elementos comuns aos dois arrays.