Arrays

1. Objective

- How to store and manage a set of ***n*** variables at the same time. Would you create ***n*** variables? What if ***n*** is large.
- Store multiple values at the same location
- Arrays data structure
- Access values in an array.
- Multi-dimensional arrays
- How to manage strings of characters.

2. Introduction

- An array is a collection of data elements that are of the same type (e.g., integers, characters, doubles, etc.).
- Think of arrays as Matrices in Math.
- Assume you need a set of n values. Instead of declaring individual n variables, such as var1, var2,..., and varn, you can use one single declare one array variable such as vars and store and access all values: var[0], vars[2], ..., vars[n-1].
- A specific element in an array is accessed by an index.

3. Array Declaration & Initialization

- Syntax:

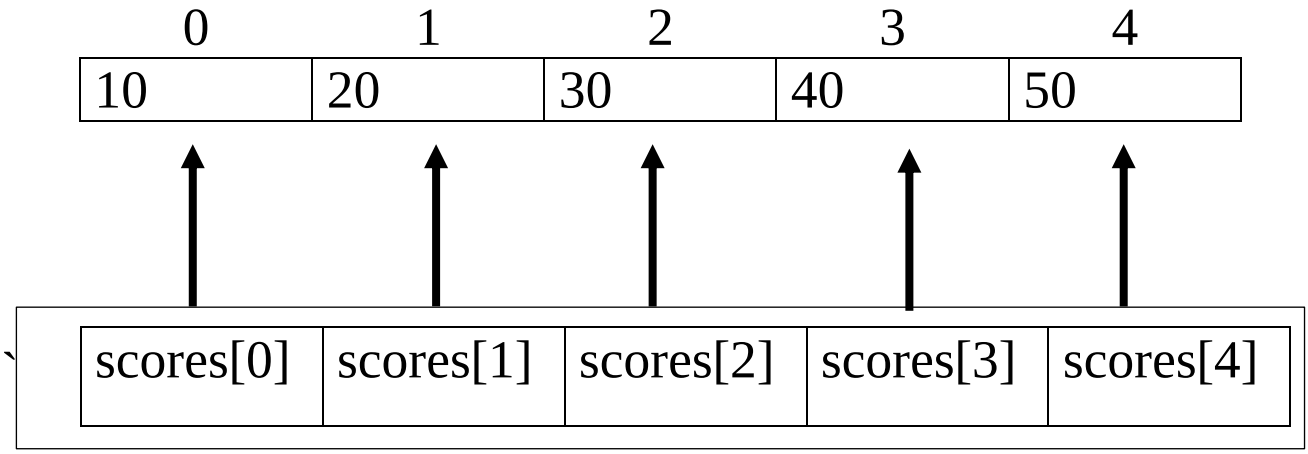
  type arrayName[**array_size**]

  Where

  - **arrayName** is a single-dimensional array.
  - The array elements are all values of the type **type** and **type** can be any valid C data type.
  - The size of the array is indicated by **array_size**, the number of elements in the array and **array_size** must be an **int constant or a constant expression**.

- Note that an array can have multiple dimensions.

- Example:
  - Declare an array of 10 integers:
    - int scores[10];



- Other examples:

  char str[10];

- Initializing an array:

  int scores[5] = {10,20,30,40,50};
  char str[10]={‘H’,‘E’,‘L’};

- How to access element of an array?
  - An element is accessed by indexing the array name. This is done by placing the index of the element within square brackets after the name of the array.
  - For example

```
//gcc 5.4.0

#include  <stdio.h>

void main(void)
{
   int n = 5;
   int scores[5] = {10,20,30,40,50};

   for (int i=0; i<n; ++i)
      printf("scores[%d] = %d\n", i, scores[i]);
}
```

- How to get the size of an array?
  - Use sizeof buil-in function.
  - Remember sizeof returns the number of bytes of the type of the array.
  - Example1:

```
//gcc 5.4.0

#include  <stdio.h>
```

```
void main(void)
{
   int n = 5;
   int scores[5] = {10,20,30,40,50};

   for (int i=0; i<sizeof(scores); ++i)
      printf("scores[%d] = %d\n", i, scores[i]);
}
```

Output:

```
scores[0] = 10
scores[1] = 20
scores[2] = 30
scores[3] = 40
scores[4] = 50
scores[5] = 32767
scores[6] = 1241097216
scores[7] = -1022739658
scores[8] = 0
scores[9] = 0
scores[10] = 2044827696
scores[11] = 32693
scores[12] = 0
scores[13] = 0
scores[14] = 752499464
scores[15] = 32767
scores[16] = 2050915488
scores[17] = 1
scores[18] = 4195520
scores[19] = 0
```

- Example2:

```
//gcc 5.4.0

#include  <stdio.h>

int main(void)
{
   char myarr[10] = {'a','b','c'};
   for (int i = 0; i <sizeof(myarr) ; i++) {
      printf("myarr[%d]=%c\n",i, myarr[i]);

    }

   return 0;
}
```

- Example2: Read a string of characters.

```
//gcc 5.4.0
#include  <stdio.h>
void main(void)
{
   int n = 5;
   char scores[] = "Hello World";

   for (int i=0; i<sizeof(scores); ++i)
      printf("scores[%d] = %c\n", i, scores[i]);
}
```

4. Questions/Practice

- Write a C program that copies the content of an array into another array. Check if the two arrays are of same length

```
//gcc 5.4.0

#include  <stdio.h>

int main(void)
{
   int myarr1[5] = {1,2,3,4,5};
   int myarr2[5];
   printf("%d\n", sizeof(myarr1)/4);
   if (sizeof(myarr1) == sizeof(myarr2)){
      for (int i = 0; i <sizeof(myarr1)/4 ; i++) {
         myarr2[i]= myarr1[i];

      }
      for (int i = 0; i <sizeof(myarr2)/4 ; i++) {
         printf("myarr2[%d]=%d\n",i, myarr2[i]);

      }
   }
   else
      printf("Arrays are not of the same length!!!!\n");
   return 0;
}
```

- Write a C program that prints the content of an array in reverse. If the content of the array is

  10 20 30 40 50

  The output should be:

  50
  40
  30
  20
  10

```
//gcc 5.4.0
#include  <stdio.h>
#define n  5
int main(void)
{
   int scores[n] = {10,20,30,40,50};
   for (int i=n-1; i>=0; --i)
      printf("scores[%d] = %d\n", i, scores[i]);

   return (0);
}
```

- Multiply the content of two one-dimensional array and save the output in a new array. For example, if the two arrays are:

10 20 30 40 50

And

1 2 3 4 5

The output should be:

10 40 90 160 250

```
//gcc 5.4.0

#include  <stdio.h>

#define n  5

int main(void)
{
   int array1[n] = {10,20,30,40,50};
   int array2[n] = {1,2,3,4,5};
   int array3[n] = {1,2,3,4,5};

   for (int i=0; i<n; ++i)
      printf("array3[%d] = %d\n", i, array1[i]*array2[i]);

   return (0);
}
```

- Write a C program that computes the sum of all the values in an array. Assume that he values are numbers.
- Write a C program that computes the average of all the values in an array. Assume that he values are all integers.
- Write a C program that finds the largest number in an array. Assume that he values are all integers.
- Write a C program that finds the smallest number in an array. Assume that he values are all integers.
- Write a C program that finds the second largest number in an array. Assume that he values are all integers.
- Write a C program that increments each value in an array. Assume that he values are numbers.

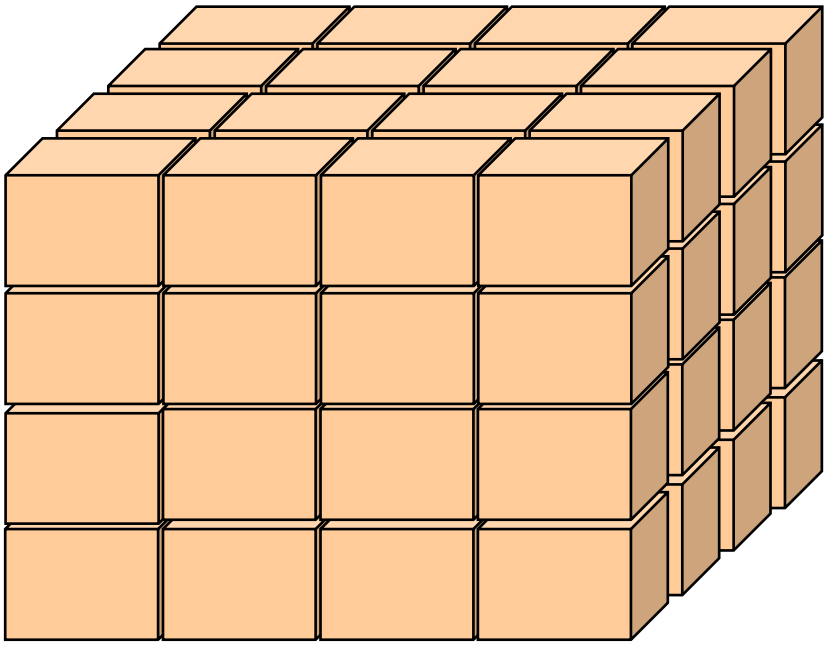
5. Multi-dimensional Arrays

- C language supports multidimensional arrays.
  - Two dimensional array
  - Three dimensional array
  - Four dimensional array etc…

- Two dimensional arrays:
  - The simplest form of the multidimensional array is the two-dimensional array
  - Two dimensional array is nothing but array of array.

	Col 0	Col 1	Col 2	Col 3
Row 0	A[0][0]	A[0][01]	A[0][02]	A[0][3]
Row 1	A[1][0]	A[1][01]	A[1][02]	A[1][3]
Row 2	A[2][0]	A[2][01]	A[2][02]	A[2][3]

- Three dimensional arrays:



- Syntax of declaration of a n-dimensional array:

  type array-name[d_1][d_2]…[d_n]

- Example :
  - int myarr[2][3];
  - double myarr[2][3][3];

- Initialization:

```
int a[2][3] = { {10, 11, 12}, {13, 14, 15} };
```

```
//gcc 5.4.0

#include  <stdio.h>

#define n  5
#define m  5


int main(void)
{

   int scoreBoard[n][m] = {{10,20,30,40,50}, {11,21,31,41,51},
{12,22,32,42,52}, {13,23,33,43,53},{14,24,34,44,54}};

   for(int i=0; i<n; ++i)
      for(int j=0; j<m; ++j)
      printf("scores[%d][%d] = %d\n", i, j, scoreBoard[i][j]);

   return (0);
}
```

6. Questions/Practice

- Write a C program that adds two matrices. Here an example:

$$\begin{bmatrix} 2 & 3 \\ 5 & 4 \end{bmatrix} + \begin{bmatrix} 10 & -1 \\ 5 & 0 \end{bmatrix} = \begin{bmatrix} 12 & 2 \\ 10 & 4 \end{bmatrix}$$

- Write a C Program to find the sum of diagonal elements of a two-dimensional matrix.

- Write a C program that finds the largest element of a three dimensional array.

- Write a C program that calculates the sum of all the elements of a multi-dimensional array. Implement your program for a 3-dimensional array.

- Write a C program that takes a two dimensional matrix as input and converts every entry to either 0 or 1 depending whether the value is even or odd. For example:

$$\begin{bmatrix} 4 & 5 & 3 \\ 0 & 1 & 2 \\ 11 & 12 & 10 \end{bmatrix}$$

The output should be:

$$\begin{bmatrix} 0 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$