



LAB 7 ARRAYS I

**Faculty of Electronic Engineering Technology
Universiti Malaysia Perlis**

1. OBJECTIVES:

- 1.1 To introduce the array data structure.
- 1.2 To be able to define an array, initialize an array and refer to individual elements of an array.
- 1.3 To be able to use arrays to store and sort data in a program.

2. INTRODUCTION:

Array is a collection or a data structure of a fixed number of components or elements where in all of the components or elements are of the **same type**.

To declare an array type of data structure, the command we use as below format.

2.1 (1-D) One dimensional array

Array declaration format :

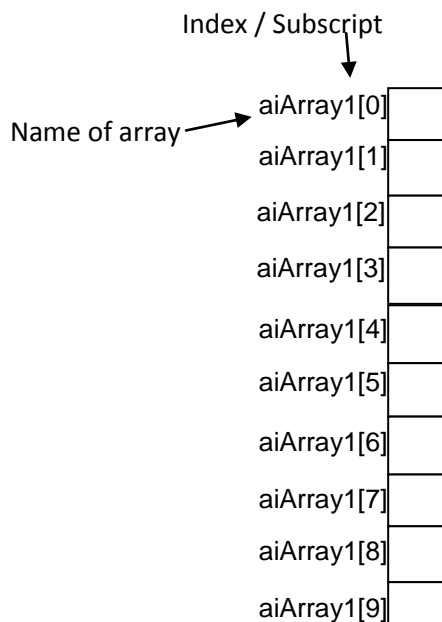
<data type> <variable_name>[subscript/index]

Subscript or index shall start with 0.

Example of an array data structure named **aiArray1**, which has 10 components of type integer:

```
int aiArray1[10];
```

The illustration of the above array:



2.2 (2-D) Two dimensional array

Array declaration format :

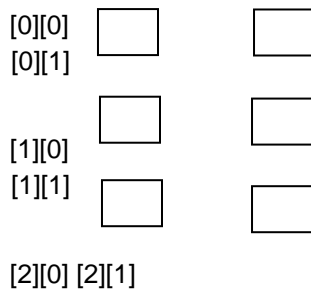
<data type> <variable_name>[subscript/index] [subscript/index]

Subscript or index shall start with 0.

Example of an array data structure named **aiArray2**, which has 6 components of type integer.

```
int aiArray2[3][2];
```

The illustration of the above array:



3. TASKS:

3.1 Declare the below array type variable.

- a. Variable name is **mark**, consists of 20 components with data type of **float**.

```
float mark [2][10];
```

- b. Variable name is **terracehouse**, consists of 15 components with data type of **integer**.

```
int terracehouse [3][5]
```

- c. Variable name is **matrix1**, consists 5 rows and 5 columns with data type of **double**.

```
double matrix1 [5][5]
```

- d. Variable name is **flathouse**, consists of 15 rows and 10 columns with data type of **integer**.

```
int flathouse [15][10]
```

3.2 For Questions 3.2 (a - c) assume the following environment.

```
#define MAX 50 int  
a[MAX], i, j, temp;
```

a. What is the effect of this program segment?

```
for (i = 0; i < MAX / 2; ++i)  
{ temp = a[i];  
  a[i] = a[MAX - i - 1];  
  a[MAX - i - 1] = temp;  
}
```

The segment effecting by filling half from data size from the array. Then the array slot number, "i" are defined by variable. The MAX number are subtracted by "i" and 1. The value of the subtract are store in a[i] array.

b. What is the effect of the following program segment?

```
for (i = 0; i < MAX - 1; ++i)  
if (a[i] > a[i + 1])  
{ temp = a[i];  
  a[i] = a[i + 1];  
  a[i + 1] = temp; }
```

The following segment are affecting the program by scanning the value of variable below the (MAX -1). And the program will be finding if the of the array is bigger than the value of array in the next slot. If the value is bigger than the next slot, the program will swap the place.

c. What is the effect of the following program segment?

```
temp = 0;  
for (i = 1; i < MAX; ++i)  
if (a[i] > a[0]) ++temp;
```

The following segment are affecting the program by scanning the value of variable on "i" array where to get see if the value in "i" array are bigger than the array in slot 0. If the condition is satisfied, the temp value will increase by 1.

d. How many numbers can be stored in the array declared below?

```
double arr[10][5][6];
```

300 number can be insert in the 3D array.

e. What will be the values of k[1] and k[3] after execution of the code segment below using the data shown?

```
int k[6] = {0, 0, 0, 0, 0, 0};  
for (i = 3; i < 6; ++i)  
{ scanf("%d", &n);  
  k[n] = i;  
}
```

Data: 2 0 1

K[1]= 3

K[3] = 0

For Questions f - h, refer to the declarations and initializations below. Indicate whether each of the statements is valid. If the statement is valid, indicate what value is displayed. If the statement is invalid, explain why.

```
double x[8] = {16.0, 12.0, 6.0, 8.0, 2.5, 12.0, 14.0, -54.5}; int
j = 5;
```

f. `printf("%.2f\n", x[j] + 1);`

13.0

g. `printf("%.2f\n", x[j + 1]);`

14.0

h. `printf("%.2f\n", x[j * j]);`

0

i. Which code fragment below fills M x N matrix m with the sums of corresponding elements of M x N matrices p and q?

i. `m = p + q;`

ii. `for (i = M; i < N; ++i)m[i]
= p[i] + q[i];`

iii. `for (i = 0; i < M; ++i)for
(j = 0; j < N; ++j) m[i][j]
= p[i][j] + q[i][j];`

ii

iv. `for (i = 0; i < M; ++i)for
(j = i; j < N; ++j) m[j][i]
= p[j][i] + q[j][i];`

j. What is accomplished by this code fragment if m is a matrix with r rows and c columns?

```
for (i = 0; i < r; ++i)
{ sum[i] = 0; for (j = 0; j <
c; ++j) sum[i] += m[i][j]; }
```

The first for will consider the row array and the second array will fill in the column array.

k. If m is a 7 x 7 integer matrix, what is displayed by this code fragment?

```
for (i = 0; i < 7; ++i)
printf("%8d", m[3][i]);
printf("\n");
```

The fragment will fill the array of row in the 3 column.

- l. Write a C program segment that takes a single M x N integer matrix argument, and finds and display the largest value in the matrix.

```
for (int i = 1; i < n; ++i)
{
    if (arr[0] < arr[i])
    {
        arr[0] = arr[i];
    }
}
```

- m. Assume that v is a 5-element array, m is a 5 x 4 matrix, and r is a 4-element array. All three arrays contain type double values. Write a code fragment that multiplies v by matrix m producing result r.

3.3 Write a program that declares and initializes an array of 10 elements, it is a (1-D) one dimensional integer array named **temperature**. Use the following temperatures to initialize the array:

78 89 65 90 35 20 88 101 56 99

Then, display the contents of the array on the screen and calculate and display the mean (average) of the temperatures.

```
#include <stdio.h>
int main ()
{
    int temprature[100],i,sum;
    double avg;
    printf("Please enter 10 temprature:");
    // Get the 10 temperature
    for (i=0;i<10;++i)
    {
        scanf("%d",&temprature[i]);
    }
```

```
    // Display the array
    for (i=0;i<10;++i)
    {
        printf("%d\t",temprature[i]);
    }

    // Calculate the average
    for (i=0;i<10;++i)
    {
        sum += temprature[i];
    }

    avg = sum / 10;

    printf("\n\nAverage %.2lf", avg);

    return 0;
}
```

3.4 Write a program that reads five (5) numbers and stores it to an array named **number**; then calculate the total of the numbers and prints the numbers in reverse order. The output of your program shall look like this:

Sample output:

```
Enter five numbers: 12 76 34 52 89

The sum of the numbers is: 263

The numbers in reverse order are: 89 52 34 76 12
```

```
#include <stdio.h>
int main ()
{
    int number [10], i, temp, sum;
    printf("Enter five numbers : ");
    for (i=0;i<5;++i)
    {
        scanf("%d", &number[i]);
    }

    // Sum of the number
    for (i=0;i<5;++i)
    {
        sum += number[i];
    }

    // Re-Arange
    for (i=0;i<5/2;i++)
    {
        temp = number [i];
        number [i] = number [5 - i - 1];
        number [5 - i - 1] = temp;
    }

    // Print
    printf ("The sum of the numbers is : %d\n",sum);
    printf ("The numbers in reverse order are :");
    for (i=0;i<5;++i)
    {
        printf ("%d\t",number [i]);
    }

    return 0;
}
```


Additional Tasks

You have been asked to write one part of a detector analysis software package for a telescope. Your program takes as an input the brightness [0 -13] of each point in a two-dimensional array representing an image of the sky. Use a 5 x 5 integer array for this image. Find and display the x and y coordinates and the value of the brightest pixel. If more than one pixel has this highest value, information for all highest-valued pixels should be displayed.

Sample output:

```
Enter brightness data for coordinate [0][0]: 4
Enter brightness data for coordinate [0][1]: 6
Enter brightness data for coordinate [0][2]: 7
Enter brightness data for coordinate [0][3]: 3
Enter brightness data for coordinate [0][4]: 10
Enter brightness data for coordinate [1][0]: 5
Enter brightness data for coordinate [1][1]: 12
Enter brightness data for coordinate [1][2]: 9
Enter brightness data for coordinate [1][3]: 7
Enter brightness data for coordinate [1][4]: 12
Enter brightness data for coordinate [2][0]: 6
Enter brightness data for coordinate [2][1]: 8
Enter brightness data for coordinate [2][2]: 7
Enter brightness data for coordinate [2][3]: 4
Enter brightness data for coordinate [2][4]: 3
Enter brightness data for coordinate [3][0]: 2
Enter brightness data for coordinate [3][1]: 6
Enter brightness data for coordinate [3][2]: 8
Enter brightness data for coordinate [3][3]: 7
Enter brightness data for coordinate [3][4]: 6
Enter brightness data for coordinate [4][0]: 5
Enter brightness data for coordinate [4][1]: 12
Enter brightness data for coordinate [4][2]: 4
Enter brightness data for coordinate [4][3]: 6
Enter brightness data for coordinate [4][4]: 8
There were 3 pixels with max brightest 12 as follows:
Pixel at [1][1]
Pixel at [1][4]
Pixel at [4][1]
```

```

#include <stdio.h>

int main()
{
    int bright[5][5], i, j, max = 0, count = 0;

    for(i=0; i<5; i++)
    {
        for(j=0; j<5; j++)
        {
            printf("Enter value for disp[%d][%d]:", i, j);
            scanf("%d", &bright[i][j]);

            if (bright[i][j] > 14 || bright[i][j] < 0)
            {
                printf("Invalid Please Enter amount above 0 and below 14\n");
                printf("Enter value for disp[%d][%d]:", i, j);
                scanf("%d", &bright[i][j]);
            }
        }
    }

    // finding the max
    for(i=0; i<5; i++)
    {
        for(j=0; j<5; j++)
        {
            if (bright[i][j] > max)
            {
                max = bright[i][j];
            }
        }
    }
    printf("%d", max);

    // Count frequency
    for(i=0; i<5; i++)
    {
        for(j=0; j<5; j++)
        {
            if (bright[i][j] == max)
            {
                count++;
            }
        }
    }
    printf("\n%d", count);

    for(i=0; i<5; i++)
    {
        for(j=0; j<5; j++)
        {
            if (bright[i][j] == max)
            {
                printf("\nPixel at [%d][%d] = %d", i, j, max);
            }
        }
    }

    return 0;
}

```