

## Programming – DT282/1 & DT228/1

### Lab 7 – Tuesday, November 8<sup>th</sup>, 2016

**Note:** You are expected to finish all programmes in your own time if you do not get these done during the lab session. This is your own responsibility.

### Arrays (part 2)

**Remember:** Use Symbolic names in your programs. Do not hard-code.

Write separate programs to:

1. What is in a[8] after the following code is executed?

```
for (i = 0; i < 10; i++)
{
    a[i] = 9 - i;
}
for (i = 0; i < 10; i++)
{
    a[i] = a[ a[i] ];
}
```

Include this code in a full program, compile and run it. Display the contents of a[8] to see its contents. Do you understand how it works and what is happening? Step through the process on paper to understand it.

2. Write a program that uses a 3x2 (2-D) array. Your program must do the following:
  - a) Enter values into the array
  - b) Display the values in the array
  - c) Find the smallest & largest value and display these
  - d) Calculate the average of the values and display this
3. Chapter 7 - Q4 (use a 1-D array), Q5, Q9

## Exercises

1. What are the subscript ranges of the following arrays?

- (a) `int array1[6] ;`
- (b) `float array2[] = { 1.3, 2.9, 11.8, 0 } ;`
- (c) `int array3[6][3] ;`
- (d) `int array4[][4] = { { 6, 2, 1, 3 } , { 7, 3, 8, 1 } } ;`

2. Write statements to define each of the following:

- (a) a one-dimensional array of floating-point numbers with ten elements
- (b) a one-dimensional array of characters with five elements
- (c) a two-dimensional array of integers with seven rows and eight columns
- (d) a 10 by 5 two-dimensional array of double precision numbers
- (e) a 10 by 8 by 15 three-dimensional array of integers.

3. What is the output from the following program segment?

```
int i, c1 = 0, c2 = 0 ;
int a[] = { 6, 7, 3, 13, 11, 5, 1, 15, 9, 4 } ;
for ( i = 0; i < 10; i++ )
{
    if( i%2 == 0 )
        c1++ ;
    if ( a[i]%2 == 0 )
        c2++ ;
}
printf( "c1=%d c2=%d\n", c1, c2 ) ;
```

4. Write a program to read in fifteen numbers and display them as follows:

- (a) each number on a separate line
- (b) on one line, each number separated by a single space
- (c) as in (b) but in the reverse order to which they were input.

5. Write a program to input numbers to two one-dimensional arrays, each having five elements, and display the result of multiplying corresponding elements together.

6. The number of users logging into a network every hour is input from the keyboard into a 24-element integer array. Write a program to display a report of the form:

Time	Number of logins	Percentage of total
0:00 - 1:00	1	0.3
1:00 - 2:00	2	0.7
...etc		
9:00 - 10:00	25	8.3
10:00 - 11:00	27	9.0
11:00 - 12:00	28	9.3
...etc		
23:00 - 0:00	8	2.7

Maximum logins 28 occurred between 11:00 and 12:00  
 Minimum logins 1 occurred between 0:00 and 1:00

7. The following two arrays represent the fixed and variable costs involved in producing each of eight items:

```
float fixed[] = { 11.31, 12.12, 13.67, 11.91, 12.30,
                  11.8, 11.00, 12.00 } ;
```

```
float variable[] = { 1.12, 1.13, 3.14, 1.35, 2.20, 1.28,
                     1.00, 2.10 } ;
```

Write a program to input an item number in the range 1 to 8 along with the number of units produced. The program should then display the cost of producing that number of units.

8. Use two `for` loops to set all the diagonal elements of a 9 by 9 integer array to 1 and all the elements not on a diagonal to 0.
9. Write a program to input values to a 4 by 5 array, search the array for values that are less than 0 and display these values along with their row and column indices.
10. Write a program to input ten integer values into an array `unsorted`. Your program should then loop through `unsorted` ten times, selecting the lowest value during each pass. For each pass through the loop, the element in `unsorted` containing the lowest value is replaced with a large value (e.g. 9999) after copying it into the next available element of another integer array `sorted`.

This is illustrated below:

`unsorted` at the start: 14 22 67 31 89 11 42 35 65 49

`sorted` at the start:

`unsorted` after the first pass: 14 22 67 31 89 9999 42 35 65 49

`sorted` after the first pass: 11

`unsorted` after the second pass: 9999 22 67 31 89 9999 42 35 65 49

`sorted` after the second pass: 11 14

etc.

Display the values in `sorted`. (Hint: see program P7C to determine the smallest value.)

11. In a magic square the rows, columns and diagonals all have the same sum. For example:

17	24	1	8	15
23	5	7	14	16
4	6	13	20	22
10	12	19	21	3
11	18	25	2	9

and

4	9	2
3	5	7
8	1	6

Write a program to read in a two-dimensional integer array and check if it is a magic square.