

## **Fundamentals of Programming**

### **Additional Exercises**

#### **Topic 3**

#### **Arrays and Pointers**

#### **Arrays**

##### **Q0**

**Write a program to prompt the user to enter 5 integers and save the data into the array `a[]`. Next, find the total of all the odd integers in the array `a[]`.**

**(Hint: Use the modulus function (%) to determine if an integer is odd. For example,  $3\%2 = 1$  and  $4\%2 = 0$ . So a remainder value of 1 from the modulus function will indicate that the integer is odd.)**

**Q1 Write a C program to ask the user to input 4 test marks, and store the data in array `marks[]`. Find the minimum and maximum test marks, output the results to the terminal screen. (Note: Test marks are from 0 to 100.)**

**Q2 Write a program with array a[5][5] containing any integers. Determine the sum for every row and column in the array. Print the results to the terminal screen.**

**(not do)**

```
int a[5][5] = {
    {1,2,3,4,5},
    {2,3,4,5,6},
    {3,4,5,6,7},
    {4,5,6,7,8},
    {5,6,7,8,9}};
```

**Example,**

```
Sum of Row 0 = 15
Sum of Row 1 = 20
Sum of Row 2 = 25
Sum of Row 3 = 30
Sum of Row 4 = 35
Sum of Col 0 = 15
Sum of Col 1 = 20
Sum of Col 2 = 25
Sum of Col 3 = 30
Sum of Col 4 = 35
```

**Q3 Write a C program that will calculate the sum and average of each row in array a[3][5] below.**

```
int a[3][5] = {
    {12, 15, 10, 8, 7},
    {19, 13, 18, 11, 9},
    {14, 23, 21, 32, 5}};
```

**Example,**

```
Sum for Row 1 = 52
Average for Row 1 = 10
Sum for Row 2 = 70
```

Average for Row 2 = 14  
Sum for Row 3 = 95  
Average for Row 3 = 19

- Q4** Write a C program to ask the user to input five (5) integers. Store the data in array `a[]` and sort the data in the array in ascending order. Print the sorted array data to the terminal.

(Note: Use nested FOR loops as the control structure. Also, focus on how to sort between three integers at a time, before applying it to the nested FOR loop.)

**Example,**

```
Enter integer 1: 34
Enter integer 2: 51
Enter integer 3: 2
Enter integer 4: 18
Enter integer 5: 40
Unsorted: 34 51 2 18 40
Sorted: 2 18 34 40 51
```

- Q5** Write a C program to convert an 8-bit char value into binary. (Note: Signed values are represented by two's complement. Range of char values is -128 to 127.)

**(not do)**

**Example,**

```
Enter number (-128 to 127): 87
Result = 01010111
```

```
Enter number (-128 to 127): -59
Result = 11000101
```

## Pointers

- Q6** Write a C program to read 10 integers from the user, store the data in array `a[]`. Using a pointer, output values from only the odd elements of the array `a[]`.

(Note: Use any loop to read from the user's input, but use a **WHILE** loop for the printing of values in `a[]`. Remember when using a pointer to output the contents in an array, the size of array `a[]` should be +1 to accommodate the pointer's index from elements 1, 3, 5, ..., 11; so that the NULL character (`\0`) may be used to stop the **WHILE** loop.)

**Example,**

```
Enter value 0: 5
Enter value 1: 2
Enter value 2: 8
Enter value 3: 6
Enter value 4: 3
Enter value 5: 11
Enter value 6: 7
Enter value 7: 9
Enter value 8: 4
Enter value 9: -1
2 6 11 9 -1
```

- Q7** Write a C program to generate 12 numbers from the Fibonacci sequence that is to be stored in array `fibonacci[]`. Use the pointer to print only the even numbers of the Fibonacci sequence to the terminal screen.

(Note: Even numbers are found using the formula: `Number%2 == 0`. The Fibonacci sequence does not have to use pointers to generate the numbers, but saving the data using a pointer is recommended.)

**Example,**

```
0 1 1 2 3 5 8 13 21 34 55 89
2 8 34
```

- Q8 Write a C program to read five (5) characters from the user using a pointer. The program needs to determine the minimum and maximum ASCII values in the characters, where the smallest value is 'a' and largest value is 'z'.**

**(Note: Use a FOR or WHILE loop to read the characters from the user. The pointer should be initialised to the first element in array a[]. Text input from the user must not have any spaces.)**

**Example,**

**Enter 5 char [no spaces]: zqdag**

**Min char = a**

**Max char = z**

- Q9 Write a C program to prompt the user for four (4) integers, and store in the 2-dimensional array a[2][2] using a single pointer. Find the determinant of the matrix a[][].**

**(Note: Use only one FOR loop to read values from the user with the pointer. Do not use nested FOR loops. Calculation of the determinant need not use the pointer.)**

**Example,**

**Enter integer 1: 5**

**Enter integer 2: 2**

**Enter integer 3: 3**

**Enter integer 4: 4**

**Determinant = 14.00**

**Q10 Write a C program to determine the gradient of a straight line graph. Given the two pairs of x- and y-coordinates are stored in array**

$$\text{coord}[2][2] = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \end{bmatrix}$$

**The program must read from the user the values for two x- and y-coordinates using a single pointer, then calculate the gradient. Output the result to the terminal screen.**

**Example,**

**Enter x1: 1**

**Enter y1: 12**

**Enter x2: 15**

**Enter y2: -24**

**Gradient = -2.57**