**Class Exercise on Arrays**

**Set A**

1. Create an array of integer with size n. Return the difference between the largest and the smallest value inside that array.

**Definition of Done:**

a) The program should ask user to enter the size and elements of the array.

b) The program should display the difference b/w the largest & smallest element.

1. Write a program that initializes an array with ten random integers and then prints four lines of output, containing:

* Every element at an even index
* Every odd element
* All elements in reverse order
* Only the first and last element

**Defining of done:**

1. Program should ask the user to input 10 integer values
2. It should display the output line by line as asked in the question
3. Finally, should ask the user to continue (Y) or quit (N)
4. Write a program to read numbers in an integer array of size 5 and display the following:
5. Sum of all the elements
6. Sum of alternate elements in the array
7. Second highest element in the array

**Definition of Done**

1. The program should ask user to enter the elements of the array.
2. The program should display a menu with the above choices and ask the user to choose one of the choices.
3. The program should display the result.

4. Write a program that count occurrence of a given number in the array and its frequency.

**Definition of Done:**

1. The program should ask user to enter the elements of the array
2. The program should ask the user to enter a number to search and display the frequency of the searched element.

5. Write a program that reads a 2-D integer array with each element of maximum 2 digits and prints the following:

a) Elements of the entered array.

b) Elements of the array after each element is multiplied by 2 if it is an odd number.

**Definition of Done:**

1. The program should generate an error message if the number of digits in the array is more than two and ask the user to re-enter the number.
2. The program should use *continue* statement for ignoring even elements.
3. The array elements should be displayed in tabular form with width set to

**Set B**

6. Given an array of integers, return the number of times that two 6's are next to each other in the array. Also count instances where the second element is 7.

* arrayFunc([6, 6, 2]) → 1
* arrayFunc([6, 6, 2, 6]) → 1
* arrayFunc([6, 7, 2, 6]) → 1

**Definition of Done:**

1. Program should ask the user the array size & then input integer values
2. In the next line It should print the number of occurrences of two 6’s
3. It should also print the occurrences of ‘6’ followed by ‘7’

7. Write a method called *swapPairs*() that accepts an array of integers and swaps the elements at adjacent indexes. That is, elements 0 and 1 are swapped, elements 2 and 3 are swapped, and so on. If the array has an odd length, the final element should be left unmodified. For example, the list {10, 20, 30, 40, 50} should become {20, 10, 40, 30, 50} after a call to your method.

**Definition of Done:**

1. Program should ask the user to input array size and integer values
2. In the next line, it should print the swapped elements as desired

8. Write a method called *median*() that accepts an array of integers as its argument and returns the median of the numbers in the array. The median is the number that will appear in the middle if you arrange the elements in order. Assume that the array is of odd size (so that one sole element constitutes the median) and that the numbers in the array are between 0 and 99 inclusive.

For example, the median of {5, 2, 4, 17, 55, 4, 3, 26, 18, 2, 17} is 5 and the median of {42, 37, 1, 97, 1, 2, 7, 42, 3, 25, 89, 15, 10, 29, 27} is 25.

Similarly, in case of even size, the median would be the average of middle two numbers.

**Definition of Done:**

1. Program should ask the user to enter the array size and input integer values
2. The median() method should return the median of the array based on its size

9. Given an array of pairs of integers. The task is to sort the array with respect to second element of the pair.

Example:

Input: [(2, 1), (5, 3), (6, 2), (1, 7)]

Output: [(1, 2), (3, 5), (2, 6), (1, 7)]

**Definition of Done:**

1. Create a class Pair and store the pairs in an array
2. Ask the user to enter the array size and elements into it
3. Define a static function sort that sorts an array of pairs w.r.t the second element of the pair

10. Write a program that accepts an array of Strings as its argument and indicates whether that array is a palindrome—that is, it reads the same forwards as backwards.

* For example, the array {"alpha", "beta", "gamma", "delta", "gamma", "beta", "alpha"} is a palindrome.

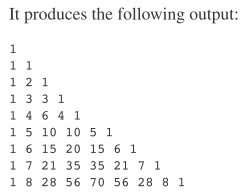
**Definition of Done:**

1. Program should ask the user to enter the array size and input String values
2. In the next line, it should print whether the entered array is palindrome or not

**Set C**

11. Write a program that constructs a jagged two-dimensional array to represent pascals triangle. It takes advantage of the fact that, each value other than the 1s that appear in the beginning and at the end of each row is the sum of two values from the previous row.

For example:



**Definition of Done:**

1. Program should ask the user to enter the number of rows to print
2. Finally, it should display the pascals traingle

12. Use a two-dimensional array to solve the following problem: A company has four salespeople (1 to 4) who sell five different products (1 to 5). Once a day, each salesperson passes in a slip for each type of product sold. Each slip contains the following:

a) The salesperson number

b) The product number

c) The total dollar value of that product sold that day

Thus, each salesperson passes in between 0 and 5 sales slips per day. Assume that the information from all the slips for last month is available. Write an application that will read all this information for last week’s sales and summarize the total sales by salesperson and by product.

**Definition of Done:**

1. All totals should be stored in the two-dimensional array sales.
2. Results should be displayed in a tabular format, with each column representing a salesperson and each row representing a particular product.
3. Cross-total of each row should get the total sales of each product for last week.
4. Cross-total of each column should get the total sales by the salesperson for last week. The output should include these cross-totals to the right of the totalled rows and to the bottom of the totalled columns.