Hands-On Activities – Milestone 3

Activity 1:

Design an algorithm to accept 15 numbers for an array from the user. The algorithm must identify and display only those numbers which are exactly divisible by 3. If there are no numbers which are exactly divisible by 3, print a message "**No numbers present**".

Activity 2:

The sum of two numbers is 25 and their difference is 13. Design an algorithm to find their product.

Activity 3:

Design an algorithm to accept 5 strings from the user and print them in lexicographical order.

For Example –
If the input strings are:
abc, a, cde, ccd, xyz

The output must be:
a, abc, ccd, cde, xyz

Activity 4:

The ratio between the perimeter and the breadth of a rectangle is 5: 1. If the area of the rectangle is 216 sq. cm, design an algorithm to find the length of the rectangle.

Activity 5:

Three friends working together earned Rs. 375. This amount is divided among them in such a way that if Rs. 4, Rs. 5, Rs. 6 is subtracted from their respective shares; the remainders will be in the ratio 3: 4: 5. Design an algorithm to calculate their individual shares.

Activity 6:

50 students enrolled in a MBA program, of the enrolled students 90% took the final exam. Two thirds of the students who took the final exam passed the final exam. Design an algorithm to find out how many students passed the final exam.

Activity 7:

Design an algorithm to accept 6 elements (only positive) for an array from the user. The algorithm must then find the number which is one number less than the least number in the array.

Note 1 - if the least number in the array is 0, then there is no number less than that and the output will be zero itself

Note 2 - the least number got need not be present in the array

For Example –

If input elements are:

5, 3, 6, 4, 8, 10

The least element in the array is 3 and the number less than 3 is 2

Activity 8:

Design an algorithm to accept an array of 5 positive integers. The algorithm must then find the smallest positive integer in the array which cannot be formed from the sum of 2 numbers in the array.

For Example –

If the input elements are:

5, 3, 2, 6, and 4

Method 1 (look at different methods to solve):

6 can be formed by 2 + 4 (both elements from the array)

5 can be formed by 3 +2 (both elements from the array)

4 cannot be formed by adding 2 elements in the array

3 cannot be formed by adding 2 elements in the array

2 cannot be formed by adding 2 elements in the array

The answer is 2 because it is the smallest integer which cannot be formed.

Activity 9:

Design an algorithm for the below puzzle –

Four people need to cross a bridge.

A maximum of two people can cross the bridge at a time.

There is only 1 lamp and the lamp must accompany them while crossing the bridge.

The two people have to walk together, i.e. the two people should walk at the speed of the slower person. Calculate the minimum time in which all four people can cross the bridge.

Below is the time taken by each individual to cross the bridge:

A – 1 minute
B – 2 minutes
C – 7 minutes
D – 10 minutes

Your program should be generic, i.e. it should be able to solve this puzzle for any number of people, say 7 people with below timings -

A – 1 minute
B – 2 minutes
C – 7 minutes
D – 10 minutes
E – 15 minutes
F – 20 minutes
G – 30 minutes

Activity 10:

Design an algorithm which accepts values of radius R and height H of a cone. The algorithm must then calculate the volume and surface area of the cone.
Note – The values entered for R and H are positive and non-zero. Also both radius and height belong to same unit (may be cm or meters)

Formulas –
Surface area = Pi * r * (r + sqrt(r*r + h*h2))
Volume = 1 / 3 * Pi * r*r *h

Value of Pi is 22/7

Activity 11:

Design an algorithm to calculate the sum of even numbers up to N (excluding N) and print it. The value of N is entered by the user and it must be positive and non-zero.

For Example-
If value of N is 6
Sum of even numbers till 6 is 2+4= 6

Activity 12: *Find First N Fibonacci Numbers*

Design an algorithm to find the first N Fibonacci Numbers and print it. The value of N is provided as an input by the user.

For Example-
If the value on N is 10

First 10 FIBONACCI numbers are:
0
1
1
2
3
5
8
13
21
34

Activity 13:

Design an algorithm to check whether a given character is present in a string (which is input by the user). The algorithm must display the position of occurrences and the number of times occurred.
If the character does not occur in the given string, then a message "**Character not present in the given string**" must be displayed.

For Example –

Enter character to be searched: r
Enter the string: programming

The output must be:
Positions of 'r' in the string "programming" are: 1 and 4
(**Position always starts from 0th place**)
Character 'r' occurred for 2 times.

Activity 14:

Design an algorithm which accepts a number (with at least 2 digits or greater) as input from the user and must generate the greatest number which can be formed from the same digits present in the number given by the user.

For Example –
If the input number is: 123

The greatest number that can be formed by these 3 digits is 321. The output displayed must be 321.

Note 1:
If the number entered by the user is already the greatest number that can be formed, then the same number must be printed as the output.
For Example –
If the input number is: 321
The output must be: 321

Note 2:
If the user enters a single digit number (like 1 or 2..), a message "**Please enter a number with at least 2 digits or more**" must be displayed.

Activity 15:

Design an algorithm to accept 2 strings **string1** and **string2** from the user. The algorithm must then remove from string2, all characters present in string1 which are also present in string2. The new string formed must be displayed as the output.
If characters present in string1 are not present in string2, then the string2 must be displayed as it is, as the output.

For Example-
If the input strings are as below
String1 = "hello"
String2 ="helloworld"

The output string will be "world"

Note – If a character from string1 occurs more than once in string2, only its first occurrence must be removed. Like in above example character 'o' occurs twice in string2 but only the first occurrence is removed.