Week-1

1. The following method shows how to reverse the character from a string.

Step-1: START

Step-2: Input a String values and store in input value

Step-3: Declare new String variable reverse

Step-4: Declare I as length of input values

Step-4: for loop until i is greater than equal to the 0 and i--

rev=rev + str. charAt(i)

next loop

Step-5: OUTPUT reverse

2.The following pseudocode is the pseudocode for finding duplicate in an array.

Step-1: START

Step-2: Declare array as Main\_array

Step-3: DECLARE array Duplicate\_array [length. Main\_array]

Step-4: DECLARE temp, i=0, j=i+1

Step-5: for loop until I is less than Main\_array. length

temp=Main\_array[i]

for loop until j is less than Main\_array length

if temp=Main\_array[i+1]

OUTPUT temp

j++

next loop

i++

next loop

Step-6: END

Week-2

3.

Step-1: Declare firsts polynomial as array1 [1,2,3,4,5]

Step-2: Declare second polynomial as array2 [9,8,7,6]

Step-3: Declare third array as sum [array1.length]

Step-4: Declare I=0

Step-4: for loops until I is less than size of array1

sum[i]=array1[i]+array2[i]

i++

next loops

Step-5: output sum

Step-6: End

4.

Step 1: START

Step 2: INPUT number

Step 3: DECLARE remainder, reverse\_number=0

Step 4: while loop until number is equal to 0

remender=number%10

reverse\_number=reverse\_number\*10+rem

number=number/10

Step 5: if(number=reverse\_number)

DISPLAY (“The given input is palindrome”)

Else

DISPLAY (“The given input isn’t palindrome”)

Step 6: END

Time complexity:

It is the complexity which describe the amount of time taken by an algorithm to run. There are four types of time complexity.

1. Constant time complexity
2. Linear time complexity
3. Quadratic time complexity
4. Logarithmic time complexity

The time complexity of this program is O(n).