***NAME : Himanshu Dixit***

***ENROLL NO. : 21103262***

***BATCH : B11***

***Data Structure [15B11CI311]***

***Tutorial Sheet***

***Week 4***

***Q1.*** *Write pseudo code to print first 50 natural numbers using recursion. Now remove this recursion and implement explicitly using stack.*

|  |
| --- |
| Start the program.  Declare a function int natural (int n)  if n!=0  return natural(n-1);  else  return;  Print start;  From the main function call the function natural (50).  End the program. |

***Q2.*** *Write pseudo code to print Sum of Natural Numbers Using Recursion.*

|  |
| --- |
| Start the program.  Declare the function int sum (int n)  if n!=0  return n+ sum(n-1);  else  return n;  In the main function, ask the number of natural numbers from the user and assign it to a variable n.  Then, call the function sum(n).  End the program. |

***Q3.*** *Write pseudo code to find sum of all digits using recursion.*

|  |
| --- |
| Start the program.  Declare the function int sum (int n)  if n!=0  return n%10+ sum(n/10);  else  return n;  In the main function, ask the number from the user and assign it to a variable n.  Then, call the function sum(n).  End the program. |

***Q4.*** *Write pseudo code that takes a string and returns if the string is a palindrome.*

|  |
| --- |
| Start the program.  Declare a function bool palindrome(string s, int start, int end)  if (end-start==1 or end==start)  return true;  else if (s[start]==start[end])  return palindrome(s,start+1,end-1);  else  return false;  In the main function, ask the user to enter a string and store in a variable s.  Call the function palindrome (s, 0, s.length()-1) and store it in a variable check.  If check==true  Print that the string is a palindrome.  Else  Print that the string is not a palindrome.  End the program. |

***Q5.*** *A list of students is given in ascending order of their height. Find all students with a height just above 5 feet.*

|  |
| --- |
| Start the program.  Declare a function int countHeight( int arr[], int n)  if (arr [n]<=5 or n==0)  return 0;  else  return 1+ countHeight(arr,n-1);  In the main function, let the arr of height be arr, and the no. of students be n.  Call the function countHeight(arr,n) and store its value in a variable count.  Print the value of count.  End the program. |

***Q6.*** *In a list of 400 students, find*

***a)*** *the student details having 4th largest marks in Data Structures T1 if not sorted*

|  |
| --- |
| Start the program.  Sort the list in the decreasing order on the basis of marks.  Print the details of the 4th element of the list.  End the program. |

***b)*** *the student details having 4th smallest marks in Data Structures T1 if not sorted.*

|  |
| --- |
| Start the program.  Sort the list in the non-decreasing order on the basis of marks.  Print the details of the 4th element of the list.  End the program. |

***c)*** *the students’ names having marks greater than 15 if sorted in descending order of T1 mark.*

|  |
| --- |
| Start the program.  Declare a function int marksMoreThan15( int arr[], int n)  if (arr [n]<=15 or n==400)  return 0;  else  Print the name of the student.  return marksMoreThan15 (arr,n+1);  In the main function, let the arr of height be arr, and the no. of students be n.  Call the function marksMoreThan15(arr,0).  End the program |