Name: **Roll No:**   **Machine No:**

Instructions:
  I.   *The source code should have proper indentation and comments. Ensure the source code should be readable and understandable.*
  II. *The source code should be more generic such that during execution it should be more user-friendly. The code should work for any input, it should not consider any specific input.*

---

1. Write a C program using pointers to carry out the following task:

    (a) Enter the float values say N through keyboard and print the same.
    (b) Determine the range of values entered by computing minimum and maximum of the values entered. Split the range into 4 equal intervals. Print the range of numbers entered and intervals computed.
    (c) Count the number of values present in each interval, print the intervals and number of values (count) associated to each interval.

    Example:
    Input : 55, 41, 62, 75, 13, 82, 4, 99, 32, 59, 7, 17, 24, 38, 43, 62, 71, 84, 97, 60

    Range = 4 – 99 (minimum = 4, maximum = 99)
    Interval-1 = 4 – 27
    Interval-2 = 28 – 51
    Interval-3 = 52 – 75
    Interval-4 = 76 – 99

    Interval-1 = 4 – 27   5
    Interval-2 = 28 – 51   4
    Interval-3 = 52 – 75   7
    Interval-4 = 76 – 99   4

2. Write a C program to perform the following tasks using linked lists:

    a) Create a linked list and enter N values into the list such that each node carries one value.
    b) Traverse the list and print the values in the sequence as they are entered.
    c) Access each node of linked list, if the value present in the node is negative, then remove the node from the list and place the value in the new list.
    d) Traverse the new list as well as updated old list and print the values present in the lists.

    Example: 23, 44, -12, 53, -7, 15, -45, 87
    Output: New List = -12, -7, -45;
            Updated old list = 23, 44, 53, 15, 87

3. Write a C program to evaluate the given postfix expression using stack. First define the structure of the stack and use the basic functions of the stack to evaluate the postfix expression. Consider the digits 1 to 9 for representing operands, four basic arthematic operations (+ - * /) as operators and maximum length of the expression is 20.

Example: 6 8 – 4 2 + / 8 9 / 5 8 * - *
Sequence of computations are as follows:
(1) 6 – 8 = -2
(2) 4 + 2 = 6
(3) -2 / 6 = -0.33
(4) 8 / 9 = 0.89
(5) 5 * 8 = 40
(6) 0.89 – 40 = -39.11
(7) -0.33 * -39.11 = 12.91

Name: **Roll No:**   **Machine No:**

Instructions:
  I.   *The source code should have proper indentation and comments. Ensure the source code should be readable and understandable.*
  II. *The source code should be more generic such that during execution it should be more user-friendly. The code should work for any input, it should not consider any specific input.*

1. Write a C program using pointers to carry out the following tasks:

    (a) Enter the sequence of characters (string) terminated by end of line and print the same.
    (b) Determine the count of vowels and consonants present in the input characters.
    (c) Determine the counts associated to successive 2 characters satisfy the following: (i) both vowels and (ii) both are same (repetition).

    Example: "asdeausdggkouu"
    Count of vowels = 7
    Count of consonants = 7
    Count of 2 successive characters to be vowels = 4
    Count of 2 successive characters to be same character = 2

2. Write a C program to perform the following task using linked lists:

    a) Create a linked list and enter N positive integer values into the list such that each node carries one value.
    b) Traverse the list and print the values in the sequence as they are entered.
    c) Traverse the list and insert the new nodes such that successive nodes should not have either both even or both odd values. Suppose, if 2 successive nodes have even values 64 and 72, then we need to insert a new node in between them and place the value as one more than the value of previous node (i.e., 65).
    d) Traverse the updated list and print the values present in the list.

    Example: 23, 44, 12, 53, 7, 15, 45, 87
    Output: 23, 44, 45, 12, 53, 54, 7, 8, 15, 16, 45, 46, 87

3. Write a C program to evaluate the given prefix expression using stack. First define the structure of the stack and use the basic functions of the stack to evaluate the prefix expression. Consider the digits 1 to 9 for representing operands, four basic arthematic operations (+ - * /) as operators and maximum length of the expression is 20.

Example: + / * 6 – 2 5 + 9 7 – / 4 3 + 2 4

(1) 4 + 2 = 6
(2) 3 / 4 = 0.75
(3) 6 – 0.75 = 5.25
(4) 7 + 9 = 16
(5) 5 – 2 = 3
(6) 3 * 6 = 18
(7) 16 / 18 = 0.89
(8) 5.25 + 0.89 = 6.14