Reg. No.										

MANIPAL ACADEMY OF HIGHER EDUCATION

FIRST SEMESTER B.TECH. EXAMINATIONS – FEBRUARY-MARCH 2022

SUBJECT : CSE 1051/CSE_1051 : PROBLEM SOLVING COMPUTERS (DTQ)

Wednesday, February 23, 2022

Time: 03:20 – 05:00 Hrs. Maximum Marks: 40

PART – B

1A. Write a single complete program to check whether the given input is a letter or digit. [use only if statements]

i) If it is a lower-case letter convert it into an uppercase letter

ii) If it is the upper case then convert it into lower case

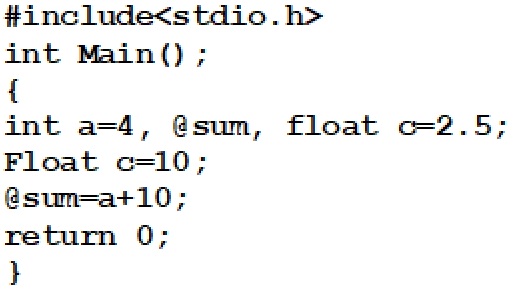
iii) If it is a digit then convert the following C expression into an equivalent if statement(s). (read all required variable values)

j=(k<5)? ((!x)? 100:500):1000)

1B. Draw a flowchart to read an n-digit integer number, extract even digits from the number and form a new number from the extracted digits.

Example: Input: 12346 Output: 246

1C. Identify the errors if any and give the reason for the same in the following C program.



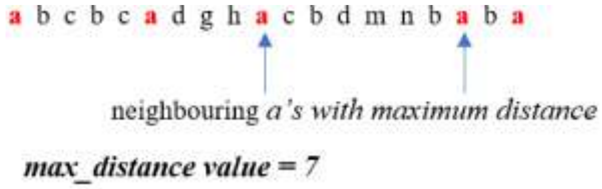
```
#include<stdio.h>
int Main();
{
int a=4, @sum, float c=2.5;
Float c=10;
@sum=a+10;
return 0;
}
```

(4+3+3 = 10 marks)

2A. Consider an elevator with a maximum capacity of 10 people. People can always leave the elevator anytime. However, people cannot enter the elevator if the total number of people (including those who are already in the elevator) exceeds the maximum of 10. Write a C program using a switch statement to perform the following operations:

1. *Add people*:
   Reads the number of persons who want to enter the elevator. Allow them if the maximum capacity is not exceeded. Partial addition of persons into the elevator is not allowed. Display the total number of people in the elevator after adding them.
2. *Remove people*:
   Removes people from the elevator. Display the total number of people in the elevator after removing them.
   Continuously display the above options until the user enters -1 to stop.

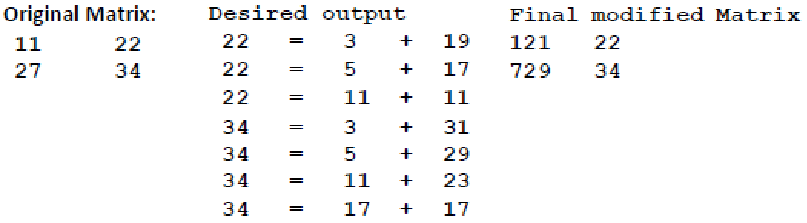
2B. Develop a C program to find the maximum distance between two neighboring **a**'s in the input string using ***for*** loop.


a b c b c a d g h a c b d m n b a b a
neighbouring *a's with maximum distance*
max_distance value = 7

2C. Illustrate the working of binary search to find 33 in the given set of numbers **6, 13, 14, 25, 33, 43, 51, 53, 64, 72, 84, 93, 95, 96, 97**. Show all intermediate computations.

(4+3+3 = 10 marks)

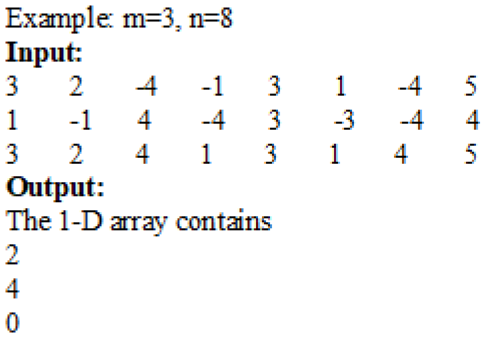
3A. Write a C function **sum2PN()** which takes each element from the 2D integer array as a parameter and print each element expressed as the sum of 2 prime numbers (all possible combinations). In the main function, read the 2-dimensional integer array, replace each element in the original array with its square if it cannot be expressed as the sum of 2 prime numbers, otherwise retain the element and display the resultant in matrix form.



```
Original Matrix:    Desired output         Final modified Matrix
 11      22          22  =  3   +  19      121    22
 27      34          22  =  5   +  17      729    34
                     22  =  11  +  11
                     34  =  3   +  31
                     34  =  5   +  29
                     34  =  11  +  23
                     34  =  17  +  17
```

3B. A programmer wants to write a program that will perform reversing the elements of an integer array. Write a C function, **exchange(int *p, int *q)**, that exchanges two integer values without any additional variables, which takes two pointers p and q containing the addresses/locations of the two integer elements of an array and then it exchanges the values in those two locations. Write a C program to read an integer array with n elements, reverse the array elements using the above function **exchange** and display the resulting reversed array. Do not use any additional functions.

3C. A cumulative-zero-sum is a point where the sign of the cumulative sum of a mathematical function is zero. Example: the function values 3, 2, -4, -1, 3, 1, -4, 5 has cumulative-zero-sum at the value -1 (i.e. 3 + 2 -4 -1) and -4. The number of cumulative-zero-sum is 2 in this case. Write a C function which takes a matrix, its dimension m, n, and a pointer to a 1-D array as arguments to compute the number of cumulative-zero-sums for each row. The number of cumulative-zero-sum for each row must be stored in the 1-D array.



Example: m=3, n=8

Input:

```
3   2   -4  -1  3   1   -4  5
1   -1  4   -4  3   -3  -4  4
3   2   4   1   3   1   4   5
```

Output:

The 1-D array contains

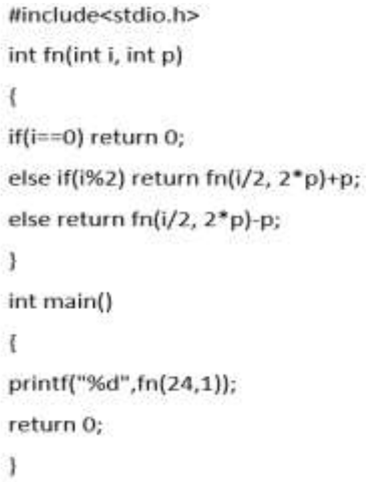
```
2
4
0
```

(4+3+3 = 10 marks)

4A. Write a C program with the following requirements:
Create a structure '**address**' to read the address of the employee with ***House_No***, ***Street_Name***, ***Place_name*** as members. Create another structure 'employee' with name, position, salary, address (structure) as its members. Using an array of structures, read 10 employee details. Read a place name from the user and display the name of the employees who belong to that place.

(4 marks)

4B. With the help of recursion call tree, predict the output of the following C program. What are the different cases in a recursive function?



```
#include<stdio.h>
int fn(int i, int p)
{
if(i==0) return 0;
else if(i%2) return fn(i/2, 2*p)+p;
else return fn(i/2, 2*p)-p;
}
int main()
{
printf("%d",fn(24,1));
return 0;
}
```

(2+1 = 3 marks)

4C. Define the type of cybercrime involved in the “Creation of email message with forged sender address”.

Write a C program to create a 1D array of 4 elements. E.g. X = {1,2,3,4}. Create a 2D array Y(3x4) size in which the second and third row are respectively, Row 2= 2 x (row1) Row 3=3 x (row1). i.e. **Y= {1, 2, 3, 4; 2, 4, 6, 8; 3, 6, 9, 12}**. Use pointer to print the address of each 1D array and also print the values of each element.

(1+2 = 3 marks)