**Section B**

**Attempt any four questions [10 x 4=40]**

**Question 1.**

An emirp number is a number which is prime backwards and forwards. Example: 13 and 31 are both

prime numbers. Thus 13 is an emirp number.

Design a class Emirp to check if a given number is Emirp number or nt. Some of the members of the

class are given below:

Class Name : Emirp

Data Members

n : stores the number

rev : stores the reverse of the number

f : stores the divisor

Member functions

Emirp(int nn) : to assign n=nn, rev=0, and f=2

int isprime(int x) : check if the number is prime using the

recursive

technique and return 1 if prime otherwise

return 0.

void isEmirp() : reverse the given number and check if both

the

original number ad the reverse number are

prime, by invoking the function isprime(int)

and display the result with an appropriate

message.

Specify the class Emirp giving details of the constructor(int), int isprime(int) and void isEmirp().

Define the main function to create an object and call the methods to check for Emirp number.

[ 10]

**Question 2.**

Design a class Exchange to accept a sentence and interchange the first alphabet with the last

alphabet for each word in the sentence, with single letter word remaining unchanged. The words in

the input are separated by a single blank space and terminated by a full stop.

Example: Input: It is a warm day.

Ouput: tI si a marw yad

Some of the data members and member functions are given below:

Class Name : Exchange

Data members

sent : stores the sentence

rev : to store the new sentence

size : stores the length of the sentence

Member functions

Exchange() : default constructor

void readsentence() : to accept the sentence

void exfirstlast() : extract each word and interchange the first and last

alphabet of the word and form a new sentence.

void display() : display the original sentence along with the new

changed sentence.

Specify the class Exchange giving details of the constructor(), void readsentence(), void exfirstlast()

and void display(). Define the main() function to create an object and call the functions accordingly

to enable the task.

[ 10]

**Question 3.**

A class Matrix contains a two dimensional integer array of order [ m x n ]. The maximum value

possible for both m and n is 25. Design a class Matrix to find the difference of the two matrices. The

details of the members of the class are given below:

Class name : Matrix

Data members

arr[][] : stores the matrix element

m : integer to store the number of rows

n : integer to store the number of columns

Member functions:

Matrix(int mm, int nn) : to initialize the size of the matrix m=mm

and

n=nn

void fillarray() : to enter the elements of the matrix

Matrix SubMat(Matrix A) : subtract the current object from the matrix

of parameterized object and return the resulting object.

void display() : display the matrix elements.

Specify the class Matrix giving details of the constructor(int,int), void fillarray(),Matrix

SubMat(Matrix) and void display(). Define the main() function to create an object and call the

functions accordingly to enable the task.

[ 10]

**Question 4.**

A class Combine contains an array of integers which combines two arrays into a single array including

the duplicate elements, if any, and sorts the combined array. Some of the members of the class are

given below:

Class Name : Combine

Data members

com[ ] : integer array

size : size of the array

Member functions/methods

Combine(intnn) : parameterized constructor to assign size = nn

voidinputarray( ) : to accept the array elements

void sort() : sorts the elements of combined array in ascending

order using the selection sort technique.

void mix(Combine A, Combine B) : combines the parameterized object arrays and stores

the result in the current object array along with the

duplicate elements , if any.

void display( ) : displays the array elements.

Specify the class Combine giving details of the constructor(int ), void inputarray( ), void sort(), void

mix(Combine, Combine) and void display( ). Also define the main function to create an object and call

the methods accordingly to enable the task.

[ 10 ]

**Question 5.**

Design a class VowelWord to accept a sentence and calculate the frequency of words that begin with a

vowel. The words in the input string are separated by a single blank space and terminated by a full stop.

The description of the class is given below:

Class Name : VowelWord

Data members

str : to store a sentence

freq : to store the frequency of words beginning with a

vowel.

Member functions

VowelWord() : constructor to initialize data members to legal initial

values.

voidreadstr() : to accept a sentence.

void freq\_vowel( ) : counts the frequency of the words beginning with a

vowel.

void display() : to display the original string and the frequency of the words that begin with a vowel.

Specify the class VowelWord giving details of the constructor( ), void readstr(), void freq\_vowel() and

void display(). Also defing the main function to create an object and call the methods accordingly to

enable the task.

[ 10 ]

**Question 6.**

A happy number is a number in which the eventual sum of the square of the digits of the number is

equal to 1.

Example :

28 = (2)2 + ( 8 )2 = 4 + 64 = 68

68 = (6)2 + ( 8)2 = 36 + 64 = 100

100 = ( 1 )2 + ( 0 )2 + ( 0 )2 = 1 + 0 + 0 = 1

Hence 28 is a happy number.

Example :

12 = (1)2 + (2)2 = 1 + 4 = 5

Hence 12 is not a happy number.

Design a class Happy to check if a given number is a happy number. Some of the members of the class

are given below:

Class Name : Happy

Data Members

n : stores the number

Member functions:

Happy( ) : constructor to assign 0 to n

voidgetnum(intnn) : to assign the parameter value to the number n = nn

intsum\_sq\_digits(int x) : returns the sum of the square of the digits of the

number x, using the recursive technique.

voidishappy() : checks if the given number is a happy by calling the

functionsum\_sq\_digits(int) and displays an appropriate

message.

Specify the class Happy giving details of the constructor( ), void getnum( int), intsum\_sq\_digits(int) and

void ishappy(). Also define a main function to create an object and call the methods to check for a happy

number.