1. Write a program in Java to accept a decimal number and convert it equivalent binary, octal (without using String & array) & hexadecimal form. Test your program forthe following data & some random data.
(Use minimum complexity to convert the base value & don't use any inbuilt methods)Class Name:

Number_System

Instance Method:

void binary (int dec) : to convert the decimal number to binaryvoid octal
(int dec) : to convert the decimal number to octal
void hexa (int dec) : to convert the decimal number to hexa-decimal

Sample Input: Enter the number: 123
Sample Output: Equivalent Binary form: 1111011Equivalent Octal
form: 173
Equivalent Hexadecimal form: 7B

2. Write a program in Java to display all the **composite Fibonacci number** between 1to 'n' (both inclusive).
Composite Fibonacci No = 8, 34, 55 [The number that is a non-prime number &also a Fibonacci Number]

Class Name: **Comp_Fibo**
Instance Variable: int n : to store the upper limit

Rest of the member variables can be assumed as required.

Instance Method:

Comp_Fibo ( ) : to initialize the variables.

void display (int num) : to display the Composite Fibonacci number.

boolean check_Composite (int n) : to check the 'n' is Prime or not, using bestmethod & if it is

Prime returns false otherwise returns true.

int chk_Fibo (int n) : to check the 'n' is Fibonacci or not (withoutusing 3rd

variable) if true returns 1 otherwise returns 0.

Write a main ( ) to test your class & call the methods using object.

3. Write a program in Java to compute the following
   void binary(int oct): accept octal number and prints it in binary numbervoid
   octal(int bin): accept binary number and print it in octal

Create a **main()** method for inputs and enable the above task.(follow the directmethods for conversions)

4. Write a program in Java to compute the following series using the concept offunction overloading:
$S = 1 - (x^2/2!) + (x^4/4!) - (x^6/6!) + ---------- + (x^n/n!)$

   Class Name: **Series**

   Class variables: int n : to accept number of terms to be added.int x : to

   accept the value of x.

   Rest of the member variables can be assumed as required.

   Public Member functions:
   void input ( ) : to accept the number of terms & the value of x.void

   sum_of_series( ) : to calculate the sum of the series. void display

   (double s) : to display the sum of the series.

   double calculate (int x, int p) : to calculate the power of x using iteration.int calculate

   (int n) : to calculate the factorial of the denominator.

Define the main ( ) in another class named as Sum_Series and inside the main ( )callthe methods.

You are not suppose to use any in-built Math functions.

5. A class **Stringop** is designed to sort the words of a sentence alphabetically.

Instance Variable: String txt : to store the sentence end with full stop.Rest of the member variables can be assumed as required.

Member function:

void readString ( ) : to accept the sentence

char caseConvert (char ch) : to convert upper case character to Lowercase & vice versa without using any String function.

String sort (String txt) : to sort the words of the sentence alphabeticallyusing Bubble Sort technique.

void display (String st) : to display the sorted string & new String aftercase conversion.

Write a main ( ) to call the methods using object.

6. Write a program in Java to do the following task:Class

Name: **Matrix**

Member variables:

int a [n] [n] : to store the elements of the matrix of (n x n) order.Rest of the

member variables can be assumed as required.

Public Member Functions:

void column_Sum (int , int) : to find the column-wise sum of elements a[ ][ ]. void right_Lower (int , int) : to display right lower triangular elements of a[ ][ ].int second_ max (int , int) : to return the 2nd maximum element of the Matrixwithout using any standard sorting technique.

void getData ( ) : to accept the elements of the matrix & the order of the Matrix.void display (int , int) : to display the elements of a[ ][ ] using one loop only.

int , int : to pass the matrix a[ ][ ] & the order of the matrix i.e. 'n'.

Test your class using the **main ( )** method.

7. Write a program in Java do the following task:class

name : **complement**

Member methods:

String binary(int n): accept a decimal number and return it's binaryString

eight_bits(int n): return 8 bits binary number

void diplay(): print 1's complement of input number

Create a **main()** method to input a number and enable the above task.

8. Write a program in Java to accept the elements in a square matrix of order **nxn** thentranspose the elements of the matrix & finally display the mirror image of that transpose form along with the original form.Sample Input:

Enter the order of the matrix (n) = 3 Mirror Image:

The Matrix is:			The transpose is:			The mirror image is:		
1	2	3	1	4	7	7	4	1
4	5	6	2	5	8	8	5	2
7	8	9	3	6	9	9	6	3

Output should be taken using both odd ordered and even ordered matrix.

9. A Vampire number is a composite natural number with an even number of digits that can be factored into two natural numbers each with half as many digits as the original number and not bothwith trailing zeros, where the two factors contain precisely all the digits of the original number, in any order of counting multiplicity.
Example: 1260 = 21 x 60 (where, 21 and 60 contain precisely all the digits of the number)
Thus, 1260 is a Vampire number.
Accept two positive integers m and n, where m is less than n and the values of both 'm' and 'n' must be greater than or equal to 1000 and less than or equal to 9999 as user input.
Display all Vampire numbers that are in the range between m and n (both inclusive) and output them along with the frequency, in the format specified below:
Test your program for the following data and some random data.
Example 1
INPUT:
m = 1002 n = 1640
OUTPUT: THE VAMPIRE NUMBERS ARE: 1260 1395 1435 1530
FREQUENCY OF VAMPIRE NUMBER IS: 4
Example 2 INPUT: m = 1810 n = 7800
OUTPUT: THE VAMPIRE NUMBERS ARE: 1827 2187 6880
FREQUENCY OF VAMPIRE NUMBER IS: 3
Example 3 INPUT: m = 8105 n = 9999
OUTPUT: THE VAMPIRE NUMBERS ARE: NIL
FREQUENCY OF VAMPIRE NUMBER IS: 0
Example 4
INPUT: m = 174 n = 4500
OUTPUT: INVALID INPUT

10. Write a program to declare a matrix A [ ] [ ] of order (M × N) where 'M' is the number of rows and 'N' is the number of columns such that both M and N must be greater than 2 and less than10.
Allow the user to input integers into this matrix. Display appropriate error message for an invalid input.
Perform the following tasks on the matrix.
(a) Display the input matrix
(b) Rotate the matrix by 270^0 degrees anti clock wise and display the resultant matrix (c) Calculate the sum of the odd elements of the matrix and display
Test your program for the following data and some random data:
Example 1
INPUT: M = 3 N = 4
ENTER ELEMENTS: 8, 7, 9, 3,-2, 0, 4, 5, 1, 3, 6, -4
OUTPUT:
ORIGINALMATRIX
8 7 9 3
-2 0 4 5
1 3 6 -4
ROTATED MATRIX (2700 ANTI CLOCK WISE)
1 -2 8 3
0 7 6 4
9 -4 5 3
SUM OF THE ODD ELEMENTS = 28

11. Write a program to declare a square matrix M[][] of order 'N'. Check if the matrix is a Doubly Markov matrix or not.

A matrix which satisfies the following conditions are Doubly Markov matrix

(i) All elements are greater than or equal to 0

(ii) Sum of each row is equal to 1.

(iii) Sum of each column is equal to 1.

Accept 'N' from the user where 3 <= N <= 9. Display an appropriate error message if 'N' is not in the given range or the entered numbers are negative.

Allow the user to create a matrix and check whether the created matrix is a Doubly Markov matrix or not Test your program for the following data and some random data: Example 1

INPUT: N=3

Enter elements in the matrix: 0.5, 0.25, 0.25, 0.25, 0.75, 0.0, 0.25, 0.0, 0.75

OUTPUT: IT IS A DOUBLY MARKOV MATRIX

Example 2

INPUT: N=3

Enter elements in the matrix: 1.5, 3, 0.15, 0.25, 4, 1.0, 0.25, 1.0, 3

OUTPUT: IT IS NOT A DOUBLY MARKOV MATRIX

Example 3

INPUT: N=2

Enter elements in the matrix: 0.8, -4.0, 0.9, 3.5

OUTPUT: NEGATIVE NUMBERS ENTERED.

INVALID ENTRY

Example 4 INPUT: N =12

OUTPUT: SIZE IS OUT OF RANGE.

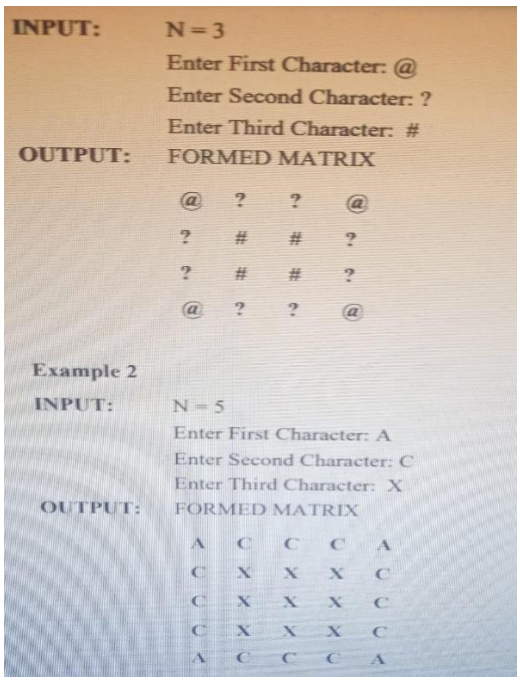
INVALID ENTRY

12. Write a program to declare a square matrix M [ ] [ ] of order 'N' where 'N' must be greater than 3 and less than Accept the value of N as user input. Display an appropriate message for an invalid input. Allow the user to accept three different characters from the keyboard and fill the matrix according to the instruction given below:

(i) Fill the four corners of the square matrix by character 1

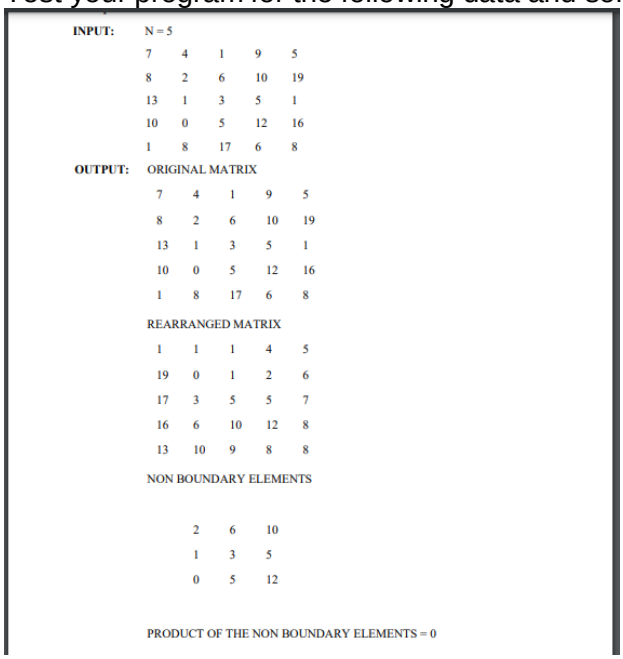
(ii) Fill the boundary elements of the matrix (except the four corners) by character 2

(iii) Fill the non-boundary elements of the matrix by character 3 (iv) Display the matrix formed by the above instructions Test your program for the following data and some random data:



13. An encoded text can be decoded by finding actual character for the given ASCII code in the encoded message. Write a program to input an encoded text having only sequence of ASCII values without any spaces. Any code or value which is not in the range 65-90 (A – Z) or 97-122 (a – z) or 32 (for space) will be ignored and should not appear in the output message.
Write a program to accept a CODE which contains only digits (0 to 9). Display an error message if the code contains any other character/s. Perform the following tasks:
(a) Decode the encoded text as per the above instructions
(b) The first alphabet of each word must be changed to UPPER case and the remaining alphabets to lower case
(c) Any consecutive sets of code 32 will be taken as only one blank space.
(d) Display it in the form of a sentence
Test your program for the following data and some random data:
Example 1
INPUT:
CODE: 10665771011153266797868
OUTPUT: James Bond
Example 2
INPUT: CODE: 667685693232837589
OUTPUT: Blue Sky
Example 3
INPUT:
CODE: 22-768#5693232375889
OUTPUT: INVALID INPUT

14. Write a program to declare a square matrix M[ ] [ ] of order (N x N) where `N' must be greater than 3 and less than 10. Allow the user to input positive integers into this matrix.
Perform the following tasks on the matrix:
(a) Sort the boundary elements in ascending order using any standard sorting technique and rearrange them in the matrix in clockwise manner.
(b) Calculate the product of the non-boundary elements.
(c) Display the original matrix, rearranged matrix and only the non-boundary elements of the rearranged matrix with their sum.
Test your program for the following data and some random data:



INPUT: N = 5

7	4	1	9	5
8	2	6	10	19
13	1	3	5	1
10	0	5	12	16
1	8	17	6	8

OUTPUT: ORIGINAL MATRIX

7	4	1	9	5
8	2	6	10	19
13	1	3	5	1
10	0	5	12	16
1	8	17	6	8

REARRANGED MATRIX

1	1	1	4	5
19	0	1	2	6
17	3	5	5	7
16	6	10	12	8
13	10	9	8	8

NON BOUNDARY ELEMENTS

2	6	10
1	3	5
0	5	12

PRODUCT OF THE NON BOUNDARY ELEMENTS = 0

15. Write a program to accept a sentence which may be terminated by either '. ' or ?' or '!' only. Any other character may be ignored. The words may be separated by more than one blank space and are in UPPER CASE. Perform the following tasks: (a) Accept a sentence and remove all the extra blank space between two words to a single blank space. (b) Accept any word from the user along with its position and insert the word in the given position. (The position is calculated by place value of each word where first word is in position 1, second word in position 2 and so on). c) Display the modified sentence. Test your program for the following data and some random data.:

Example
INPUT: MORNING WALK IS A BLESSING FOR THE WHOLE DAY.
WORD TO BE INSERTED: TOTAL
WORD POSITION IN THE SENTENCE: 5
OUTPUT: MORNING WALK IS A TOTAL BLESSING FOR THE WHOLE DAY.

16. A Composite Magic number is a positive integer which is composite as well as a magic number. Composite number: A composite number is a number that has more than two factors. Example: 10 Factors are: 1, 2, 5, 10 Magic number: A magic number is a number in which the eventual sum of the digits is equal to 1. Example: 28 = 2+8 = 10 = 1+0 = 1 Accept two positive integers m and n, where m is less than n as user input. Display the number of Composite Magic integers that are in the range between m and n (both inclusive) and output them along with the frequency, in the format specified below:
Test your program for the following data and some random data.
Example 1
INPUT: m = 40 n = 100
OUTPUT:
THE COMPOSITE MAGIC INTEGERS ARE: 46, 55, 64, 82, 91, 100
FREQUENCY OF COMPOSITE MAGIC INTEGERS IS: 6

Class name: Number
Data members/instance variables:
int num: to store thenumber
int c :to count factors
static int count: assign 0 for
count frequency
Methods/Member functions:
Number (int nn): parameterized constructor to initialize the data member num=nn
int count_of_factors(int i): returns the number of the factors (num), excluding itself, using a recursive technique
int sum_of_digits(int i): returns the sum of the digits of the number(num),using a recursive technique

void check(): checks whether the given number is Composite Magic number by invoking the functions and displays the result with an appropriate message
Define a main() function to create an object and call the functions accordingly to enable the task.