Program 1

1. WAP in Java to generate a decimal number from 100 to 200 and convert it equivalent binary, octal (without using String & array) & hexadecimal form (without using array). Test your program for the following data & some random data. (Use minimum complexity to convert the base value & don't use any inbuilt methods)

Sample Input: **Enter the number: 123**

Sample Output: **Equivalent Binary form: 1111011**

Equivalent Octal form: 173

Equivalent Hexadecimal form: 7B

Program 2

2. WAP in Java to display all the composite Fibonacci numbers between 1 to ‘n’.
Composite Fibonacci No = 8, 21, 55 [The no. that is a Composite & also Fibonacci]

Class: Comp_Fibo

<u>Instance Variable:</u>

int n : to store the upper limit

Rest of the member variables can be assumed as required.

<u>Instance Method:</u>

Comp_Fibo ( ) : to initialize the variables.

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

boolean chk_Composite (int n) : to check the **‘n’ i**s Prime or not, using best method & if it is Prime return **false** otherwise return **true.**

int chk_Fibo (int n) : to check the **‘n’** is Fibonacci or not (without using 3rd variable) if true return **1** otherwise return **0.**

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

Program 3

3. A positive natural number, (for e.g. 27) can be represented as follows:

2 + 3 + 4 + 5 + 6 + 7
8 + 9 + 10
13 + 14

Here every row represents a combination of consecutive natural numbers, which add up to **27**.Write a program which inputs a positive natural number **N** and prints the possible consecutive number combinations, which when added give **N.**

Test your program for the following data & some random data.

Sample Input: N = 9

Sample Output:
```
2 3 4

4 5
```

Sample Input: N = 15

Sample Output:
```
1 2 3 4 5

4 5 6

7 8
```

Program 4

4. Develop an object oriented 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 row_Wise_Sum (int , int) : to find the **row-wise sum** of all the elements.
void right_uptriangle(int , int) : to display **right upper triangular elements** of **a[ ][ ]**
int max_2(int , int) : to return the **2nd maximum** element of the Matrix without sorting.
void getData ( ) : to accept the elements of the matrix & the order of the Matrix.
void display (int , int) : to display the elements of the matrix using one loop only.
int calculate(int n) : to pass the matrix **a[ ][ ]** & the order of the matrix i.e. **n.**

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

Program 5

5. WAP in Java to compute the following series using the concept of function overloading:

S=1-(x2/2!)+(x4/4!)-(x6/6!)+ ---------- +(xn/n!)

Class Name: Series

Class variables:

int n : to accept a 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 & value of x using Scanner class.
void sum_of_series( ) : to calculate the sum of the series.
void display (double) : to display the sum of the series.
double calculate (int , int) : to calculate the power of x without iteration.
int calculate (int n) : to calculate the factorial of the denominator.

Declare the **main ( )** in another class named as Sum_Series and inside the **main ( )** call the methods. You are not supposed to use any in-built Math functions.

Program 6

6. A class **String_Op** is designed to sort the words of a sentence alphabetically.

Data member:

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) : to convert Upper case to Lower case & vice versa without using any String function.
String sort (String) : to sort the words of the sentence alphabetically using Bubble Sort
void display(String) : to display the sorted string & new String after case \ conversion.

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

Program 7

7. An integer array **a [ ]** consists of **N** different elements in unsorted order. We want to rearrange them according to the given instruction: Find the maximum value of the array and put it in the center position. Find the next largest value and put it to its right. Then find the next largest and place it to its left and so on alternating right and left until all integers in the array replaced. Assume that there are **at most 20 different elements in the array** (There are **no duplicate elements in the array**). For e.g. if the element of the array initially**: 7, 3, 1, 6, 4, 2, -1 and 5** then after rearranging it becomes **1, 3, 5, 7, 6, 4, 2 and -1.**

Write a program to accept N different element in an array & rearrange them according to the given instruction.

Sample Input:
```
Enter the Limit of the array: 5
Enter the Number: 7
Enter the Number: 9
Enter the Number: 2
Enter the Number: 5
Enter the Number: 6
```

Sample Output:
```
Original array:
          7 9 2 5 6
Rearrange Array:
         2 6 9 7 5
```

Program 8

8. Write a program in Java to accept the elements in a square matrix of order n x n then transpose the elements of the matrix & finally display the mirror image of that transpose form along with the original form.

<u>Sample Input:</u> Enter the order of the matrix (n) = 3

```
                                                          Mirror Image
The Matrix is: 1 2 3    Transpose form: 1 4 7               7 4 1
               4 8 9                    2 8 6               6 8 2
               7 6 5                    3 9 5               5 9 3
```

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

Program 9

9. Write a program in Java to display all the **Fibonacci numbers** from **1** to **N** using recursion and without a third variable. Accept the value of ‘**N**’ from the user.

Program 10

10. Write a program in Java to accept 10 numbers in an array and sort them **ascending** order using **Insertion sort** technique. Display the results with proper message.

Program 11

11.Write a program in Java to calculate the addition of two Complex number using passing and returning object concept. The (x + i.y) is a form of a Complex Number. Where '**x**' is the value of the real part and '**y**' is the value of the imaginary part. Design a class
Complex with the following details:

Class name : Complex

Data Members :
x : stores the real part
y : stores the imaginary part

Member Functions:
Complex (…) : Parameterized constructor to assign value to data members.
void display ( ) : To display the Complex Number.
Complex add (Complex obj) : calculates and returns the sum of the two complex numbers.

Write a **main** function to create an object and call the functions accordingly to calculate the sum of two complex numbers that is given by the user from another class **Sum**.

Program 12

12. Write a program in Java to convert a **binary number** to its **equivalent decimal form** & vice versa using **recursive function**.

Program 13

13. A positive whole number ‘**n**’ that has ‘**d**’ number of digits is squared and split into two pieces, a right-hand piece that has ‘d’ digits and a left-hand piece that has remaining ‘**d**’ or ‘**d-1**’ digits. If the sum of the two pieces is equal to the number, then ‘**n**’ is a **Kaprekar number**. The first few **Kaprekar numbers** are: **9**, **45**, **297** ……..

Example 1:
9 and 9^2 = 81, right-hand piece of 81 = 1 and left hand piece of 81 = 8
Sum = 1 + 8 = 9, i.e. equal to the number.

Example 2:
45 and 45^2 = 2025, right-hand piece of 2025 = 25 and left hand piece of 2025 = 20
Sum = 25 + 20 = 45, i.e. equal to the number.

Write a program in Java to display all **Kaprekar numbers** from **1** to **1000** and also calculate their **sum** & display the result with a proper message.

Program 14

14. Write a program in Java to accept a **Sentence ends** with (**.**) & display the smallest word from that sentence & finally reverse each word of that sentence at its position.

Sample Input: Enter the sentence: I have a pen.
Sample Output: Smallest Word: I a
Sample Output: After reverse each word at its position: I evah a nep.

Program 15

15. Write a program in Java to store ‘**N**’ numbers in a text file **“Number.txt”** then display all the **Prime numbers** and **Palindrome numbers** with a proper message.