Giri's Tech Hub Pvt.Ltd, Pune

Programming (Machine) Test

Batch: July-25

Date: 14/11/2025
Time: 02:30 to 05:30 Pm

Instructions:

1. Solve any 8 questions.
2. Input should be from user.
3. Indentation and comments mandatory.
4. Each program 1 Marks and all comments 2 Marks.
5. Do not use any inbuilt functions.

Q1. Write a java program to print this pattern.

```
        1
      1   1
    1   2   1
  1   3   3   1
1   4   6   4   1
```

Q2. Write a java program to print this pattern.

```
A                       A
A   B                   B   A
A   B   C           C   B   A
A   B   C   D   D   C   B   A
A   B   C           C   B   A
A   B                   B   A
A                       A
```

Q3. Write a java program to swaps the first two digits with the last two digits, and prints the result.

Input : 12345
Output : 45312

Q4. Write a java program to find the frequency of each digit in a given integer using function recursion.

Q5. Write a program in java to count the number of inversion in a given array

Expected Output : The given array is : 1 9 6 4 5
The inversions are: (9, 6) (9, 4) (9, 5) (6, 4) (6, 5)
The number of inversion can be formed from the array is: 5

Q6.Given a sorted array (may contain duplicates), find the first and last index of a given number x using binary search. Example: arr = [2, 4, 4, 4, 6, 7, 9], x = 4 Output: First = 1, Last = 3

Q7. Write a Java program to implement binary search using function overloading.
You must create two overloaded methods with the same name binarySearch() that perform different operations:

int binarySearch(int[] arr, int key)

- **This method should perform normal binary search on a sorted array.**
- **It should return the index of the key if found, otherwise return -1.**

int binarySearch(int[] arr, int key, boolean first)

- **This overloaded method must find the first occurrence of the key using recursion.**
- **If duplicates exist, it should return the leftmost index where the key appears.**

Q8. Create a class Employee with:

- **empId**
- **empName**
- **baseSalary**
- **performanceRating**

Create a parameterized constructor and a display() method.
Create another class BonusCalculator with:

- **setEmployee(Employee[] arr)**
- **calculateBonus() →**
  Bonus Rules:
  - **Rating 5 → 25% bonus**
  - **Rating 4 → 15% bonus**
  - **Rating 3 → 10% bonus**
  - **Rating 2 → 5% bonus**
  - **Rating 1 → No bonus**

Q9. Write a java program to create a
Base class: ArrayOperation → contains two arrays
Child class: ZigZagMerge → merge arrays in zig-zag manner:
Example:
A1 = {11,22,33,44}
A2 = {55,66,77,88}
Output: 11 55 22 66 33 77 44 88

Q10. Write a Java program to check whether a square matrix is a Magic Square.
Description:
A matrix is called a Magic Square if:

- **Sum of all rows is equal**
- **Sum of all columns is equal**
- **Both diagonals also have the same sum**

-----ALL THE BEST-----