Class Test – 2 – Section D9

1. Differentiate between the **`break`** and **`continue`** statements, with examples. **2**

2. Mention the output produced by the following **`for`** statement. Then convert the **`for`** statement into an equivalent **`while`** statement. **3**

```
for (i = 5, j = i-1; j > 0; --i, j = i-1)
    printf(“%d ”, i);
```

3. Write a complete C program (with proper indentation and variable naming convention) that asks the user for **a time in 24-hour format**, then displays **the time in 12-hour format**. Remember, 12:00 PM is not 0:00, but 12:00 AM is 0:00. The program output will be exactly as follows: **5**

```
Enter a 24-hour time:-     22:14
Equivalent 12-hour time:-     10:14 PM
```

4. Write a complete C program (with proper indentation and variable naming convention) that asks the user for an integer **n, and prints the following pattern**: *[here n = 4 given by user in this example]* **5**

```
1
1122
111222333
1111222233334444
```

Class Test – 2 – Section D10

1. Differentiate between the **`break`** and **`goto`** statements with examples. **2**

2. Mention the output produced by the following **`for`** statement. Then convert the **`for`** statement into an equivalent **`while`** statement. **3**

```
for (i = 10; i >= 1; i /= 2)
    printf(“%d ”, i++);
```

3. Write a complete C program (with proper indentation and variable naming convention) that asks the user for **a year as input** and then **prints whether the year is a Leap Year or not**. [*Hint: a year is considered as a Leap Year when it is divisible by 4 and it is not divisible by 100. A year is considered as a Leap year also when it is divisible by 400. The algorithm is given below*: **5**

Step 1 → Take integer variable year
Step 2 → Assign value to the variable
Step 3 → Check if year is divisible by 4 but not 100, DISPLAY "leap year"
Step 4 → Check if year is divisible by 400, DISPLAY "leap year"
Step 5 → Otherwise, DISPLAY "not leap year"]

4. Write a complete C program (with proper indentation and variable naming convention) that asks the user for an integer **n, and prints the following pattern**: *[here n = 4 given by user in this example]* **5**

```
   4
  43
 432
4321
```