Perform PL/SQL Programs:

Part – A

1. Write a PL/ SQL program to print a welcome message on a screen.

```
PRINT 'Welcome to SQL Server!';
```

2. Write a PL/SQL program to addition of two numbers.

```
declare @n1 int = 10;
declare @n2 int = 5
        set @n1 = 20;
        set @n2 = 56;
declare @n3 int;
        set @n3 = @n1 + @n2;
print 'sum of two number is: '+convert(varchar , @n3);
```

3. Write a PL/SQL program to print maximum number out of three numbers.

```
declare @m1 int = 10;
declare @m2 int= 20;
declare @m3 int = 30;
declare @max_num int;
        set @max_num =
        case
                when @m1 >= @m2 and @m1 >= @m3 then @m1
                when @m2 >= @m3 and @m2 >= @m1 then @m2
        else @m3
        end
print 'max number is: '+convert(varchar,@max_num);
```

4. Write a PL/ SQL program to print number from 1 to 10. (Using while loop)

```
declare @a1 int = 1
while @a1 <= 10
begin
        print 'number :' +convert(varchar,@a1);
        set @a1 = @a1 +1;
end
```

5. Write a PL/ SQL program to check where given number is ODD or EVEN.

```
declare @p1 int;
        set @p1 = 2
if @p1 % 2 = 0
        print 'number ' +cast(@p1 as varchar)+ 'is even';
else
        print 'number '+convert(varchar,@p1)+ 'is odd'
```

Part – B

6. Write a PL/ SQL program to print ODD numbers between 1 and 10.

```
DECLARE @odd_num INT = 1;
WHILE @odd_num <= 10
BEGIN
        IF @odd_num % 2 != 0
        BEGIN
                PRINT 'Odd num: ' + CONVERT(VARCHAR, @odd_num);
        END
        SET @odd_num = @odd_num + 1;
END
```

7. Write a PL/ SQL program to print Sum of 1 to 50 numbers.

```
declare @num int = 1
declare @summ int = 0;
WHILE @num <= 50
BEGIN
        set @summ = @summ + @num
        set @num = @num + 1
END
print 'sum is ' +convert(varchar,@summ)
```

8. Write a PL/ SQL program to print Sum of even numbers between 1 to 20.

```
declare @even_number_sum float = 0.0
declare @i int = 1;
WHILE @i <= 20
BEGIN
        if @i % 2 = 0
                begin
                        set @even_number_sum = @even_number_sum + @i
                end
        set @i = @i +1
END
print 'sum of even number is ' +convert(varchar,@even_number_sum);
```

9. Write a PL/ SQL program to inserting even numbers into even table & odd numbers into odd table between 1 to 50.

```
CREATE TABLE EvenTable (
   Number INT
);
CREATE TABLE OddTable (
   Number INT
);
```

```
DECLARE @NUMBER INT =1;
BEGIN
WHILE @NUMBER <= 50
   BEGIN
        IF @NUMBER % 2 =0
                BEGIN
                        INSERT INTO EvenTable values(@NUMBER)
                END
        ELSE
                BEGIN
                        INSERT INTO OddTable values(@NUMBER)
                END
                set @NUMBER = @NUMBER + 1

   END
END
```

Part – C

10. Write a PL/ SQL program to calculate the factorial of N number and display the result.

```
DECLARE @N INT= 5;
DECLARE @FACTORIAL INT= 1;
DECLARE @J INT= 1;
WHILE @J <= @N
BEGIN
        set @FACTORIAL = @FACTORIAL * @j
        set @j = @J +1
END
PRINT 'THE FACTORIAL IS:' +CONVERT(VARCHAR,@FACTORIAL)
```

11. Write a PL/ SQL program to check weather given number is prime or not.

```
DECLARE @NUMB INT = 13; -- Number to check
DECLARE @K INT = 2; -- Starting divisor
DECLARE @ISPRIME BIT = 1; -- 1 means true (assume prime)

WHILE @K < @NUMB
 BEGIN
          IF @NUMB % @K = 0
                BEGIN
                        SET @ISPRIME = 0; -- Not a prime number
                        BREAK; -- Exit the loop since we found a divisor
                END
                SET @K = @K + 1; -- Increment @K to check the next divisor
END
IF @ISPRIME = 0
        BEGIN
           PRINT 'THE GIVEN NUMBER ' + CONVERT(VARCHAR, @NUMB) + ' IS NOT PRIME';
```

```
END
ELSE
BEGIN
    PRINT 'THE GIVEN NUMBER ' + CONVERT(VARCHAR, @NUMB) + ' IS PRIME';
END
```

12. Write a PL/ SQL program to reverse a string and display the reversed string.

```
DECLARE @STR VARCHAR(100)= 'HELLO WORLD'
PRINT 'REVERSE STRING : '+( REVERSE(@STR))
```

OR

```
DECLARE @InputString NVARCHAR(100) = 'HelloWorld';   -- Input string
DECLARE @ReversedString NVARCHAR(100) = '';          -- Reversed string
DECLARE @Length INT;                                 -- Length of the input string
DECLARE @Index INT;                                  -- Loop index

-- Get the length of the input string
SET @Length = LEN(@InputString);

-- Initialize the loop index to the length of the string
SET @Index = @Length;

-- Loop through the string in reverse order
WHILE @Index > 0
BEGIN
   -- Concatenate each character to the reversed string
   SET @ReversedString = @ReversedString + SUBSTRING(@InputString, @Index, 1);

   -- Decrement the index
   SET @Index = @Index - 1;
END;

-- Display the results
PRINT 'Original String: ' + @InputString;
PRINT 'Reversed String: ' + @ReversedString;
```

13. Write a PL/ SQL program to generate the Fibonacci series up to N number and display the series.

```
DECLARE @TERM INT= 10;
DECLARE @PREV INT = 0;
DECLARE @CURR INT = 1;
DECLARE @NEXT INT;
DECLARE @COUNTER INT = 1;
PRINT @PREV
```

```
PRINT @CURR

WHILE @COUNTER < @TERM
BEGIN
        SET @NEXT = @PREV + @CURR
        PRINT @NEXT
        SET @PREV = @CURR
        SET @CURR = @NEXT
        SET @COUNTER = @COUNTER + 1
END
```

14. Write a PL/ SQL program to check given year is leap year or not.

```
DECLARE @YEAR INT = 2014
IF (@YEAR % 4 = 0 AND @YEAR % 100 != 0) OR @YEAR % 400 = 0
BEGIN
        PRINT 'THE GIVEN YEAR '+CONVERT(VARCHAR,@YEAR)+ ' LEAP YEAR'
END
ELSE
BEGIN
        PRINT 'THE GIVEN YEAR '+CONVERT(VARCHAR,@YEAR)+ ' NOT LEAP YEAR'
END
```