

Done in class 1 Write a PL/SQL code to store the first n positive integers along with their cubes in an already created table. 'n' should be taken as an input from the user. The program will also display the output of the table.

Done in class 2 Write a PL/SQL code to check whether a given number is a palindrome or not. The number should be taken as an input from the user. Use proper Exception Handling.

3 Using PL/SQL implement the functions ADD, SUB, DIVIDE and MULTP in a single program, on two values taken as an input from the user.

TODO 4 Write a Function FAREA to calculate the Area of a Square or Area of a Circle. It accepts a number and a character parameter. The character parameter is either 'C' to compute area of a Circle or 'S' to compute the Area of a Square. Raise an Exception in case of invalid input.

Done in class 5 Write a PL/SQL program which will accept an integer from 1-20 and print the Factorial of the number. If the number provided is out of range then raise an Exception

Duplicate of 15 6 Write a PL/SQL code to create a table called Number which will store all the numbers within a given range in the following way:
No Type

1 Even
2 Odd
and so on till 1-n, n is to be taken as an input from the user. (where No and Type are the attributes of the table Number).

Done in class 7 Calculate the area and perimeter of a circle for a value of radius varying from 5 to 15. Store the radius and the corresponding values of calculated area and perimeter in a table *areaperi*.

8 Print the Fibonacci and Non-Fibonacci series till n, where n is taken as an input from the user.

Duplicate of 15 9 Write a PL/SQL code to store the first n positive odd integers along with their squares in table. 'n' should be taken as an input from the user. The program will also display the rows of the table.
If the number supplied by the user is a negative number, a proper exception will be thrown. (Create suitable table.)

10 Write a PL/SQL code to check whether a given number is a palindrome or not. The number should be taken as an input from the user.

11 Write a PL/SQL program to check whether a given number is divisible by 5 or not. If the number is divisible by 5 then find the square of that number; if the number is divisible by 2 then find the square root of it else find the cube of that number.

TODO 12 Write a PL/SQL program to add the sum of the square of the digits of a positive number. The number should be of at least of 2 digits. If the number of digits in the number is more

than 2, then add the sum of the digits of the number. The number should be taken as an input from the user. Use proper exception handling in case of an invalid input.

- 13 Write a PL/SQL program which will accept two different strings from the user. Using Oracle string functions do the following:
- (i) Find the length of each string.
 - (ii) concatenate the strings.
 - (iii) change the case of the strings.

- 14 Write a PL/SQL program which will find the sum, average and standard deviation of first 'n' numbers, where n is supplied by the user. The range of n should be within 10 to 100, throw an exception if an invalid range is supplied.

- 15 Write a PL/SQL code to create a table called Number which will store all the positive numbers from 21 to a given range (n) in the following way:

No	Type
21	Odd
22	Even
23	Odd
24	Even
..

And so on... (Where No and Type are the attributes of the table Number).The input 'n' should be provided by the user. Use exception handling in case of an invalid input.

- 16 Write a PL/SQL program to check whether a given number is an Armstrong number or not. The number should be taken as an input from the user. (A number is Armstrong if the sum of cubes of individual digits of a number is equal to the number itself. For example 371 is an Armstrong number as $3^3 + 7^3 + 1^3 = 371$.)

- 17 Write a PL/SQL program to multiply the digits of a positive number. The number should be of at least 3 digits. Use proper exception handling in case of an invalid input.

- 18 Write a PL/SQL code to check whether a given string is a palindrome or not. The string should be taken as an input from the user.

- 19 Write a PL/SQL code to check whether a given number is a perfect number or not. The number should be taken as an input from the user.
(A perfect number is a positive integer that is equal to the sum of its proper positive divisors, that is, the sum of its positive divisors excluding the number itself. The first perfect number is 6, because 1, 2, and 3 are its proper positive divisors, and $1 + 2 + 3 = 6$. . The next perfect number is $28 = 1 + 2 + 4 + 7 + 14$.)

- 20 Write a PL/SQL program to check whether an input positive number is even or odd. If the number is odd, square the number and add it to the cube of that number, if the number is even find the square root of that number. Use Exception handling in case of an invalid input is given.

- 21 Write a PL/SQL code to check whether a given number is a prime or not. The number should be taken as an input from the user.

-
- | | |
|----|---|
| 22 | Write a PL/SQL code to store the first n positive integers along with their squares in an already created table. „n” should be taken as an input. The program will also display the content of the table. |
| 23 | Write a PL/SQL code to reverse an input string. |
-