Exercises

**1.** Write a program that will generate and print N random values where each value can vary between 1 and 100. At the end, the program should also display the number of odd values and even values generated. For example, if N=10 and if the program generates 5, 48, 73, 29, 33, 99, 44, 45, 78 and 88 then it will report odd values = 6 and even values = 4. You should use at least one value returning function in your program.

**2.** Write a function called is\_prime that takes in a positive integer as argument and returns a Boolean to indicate if the number is prime or not. A simple algorithm for prime checking is given below. Test the function by calling it from the main().

Let us say we want to check 83. Start from the number 2 and check if 83 is divisible by 2 (use remainder operator: %). If not divisible by 2, move to next divisor, 3. If divisible by 3, stop and report the number as non-prime. Otherwise continue checking with the next divisors 4, 5, 6…  . If the divisor value exceeds the square root of 83, stop iterating and report the number as prime.

**3.** Write a module called ‘customer.py’ which will have a function called calculatePoints(). The function will take an input called total\_purchases as a parameter and return the loyalty points earned by a customer. The points are calculated based on the total purchases (rounded figure) according to the following table.

|  |  |
| --- | --- |
| **Total purchases** | **Points** |
| $0-$100 | 10 |
| $100-$500 | 20 |
| more than $500 | 50 |

Now write a program called ‘company.py’ that will ask the user to enter the total purchases of a customer. Note that the customer module should be imported in this program. The program should then call the calculatePoints() function by passing the amount of total purchases and get the points earned by the customer. Finally, the program should display the points earned by the customer.

4. The aim of this task is to create a list for storing student records. Consider a student record consists of following fields: *ID, Last Name, First Name*, and *Marks*. Write a program that gets student data from the user and stores those as records in a list called students. You must validate the marks input so that it only contains a number between 0 and 100. In the beginning, the program should also prompt the user for the number of student records.

**Note**: The student list will have a structure like this: ['123', 'Lu', 'Kevin', 85.0, '429', 'Jane', 'Mary', 60.0, ...] (four elements per student)

Extend the program to display details of all students scoring 85+ marks. You must use slicing operations to extract information of a student from the full list.