Green University of Bangladesh
Department of Computer Science and Engineering (CSE)

Faculty of Sciences and Engineering

Semester: (Spring, Year: 2025), B.Sc. in CSE (Day)

LAB REPORT NO 01

Course Title: Data Mining Lab

Course Code: CSE 436 Section: 213 D4

Lab Experiment Name: Implementation of Sum of Odd/Even Numbers and Triangle Validity Check

<u>Student Details</u>

	Name	ID
1.	Md. Rabby Khan	213902037

Lab Date	:	**10/02/2025**
Submission Date	:	**16/02/2025**
Course Teacher's Name	:	**Md. Jahid Tanvir**

[For Teachers use only: Don't Write Anything inside this box]

<u>Lab Report Status</u>

Marks: ………………………………… **Signature: ....**

Comments: .... **Date: ....**

1. TITLE OF THE LAB EXPERIMENT

- Write a Python program to find the sum of odd and even numbers from a set of numbers.
- Write a Python program to Check Triangle is Valid or Not

2. OBJECTIVES/AIM

- To develop skills in using loops and conditionals in python.
- To understand how to classify numbers as even or odd and compute their sums.
- To apply the triangle inequality theorem to determine triangle validity.
- To write well structured, readable python code.

3. PROCEDURE / DESIGN

Program 1: Sum of Odd and Even Numbers

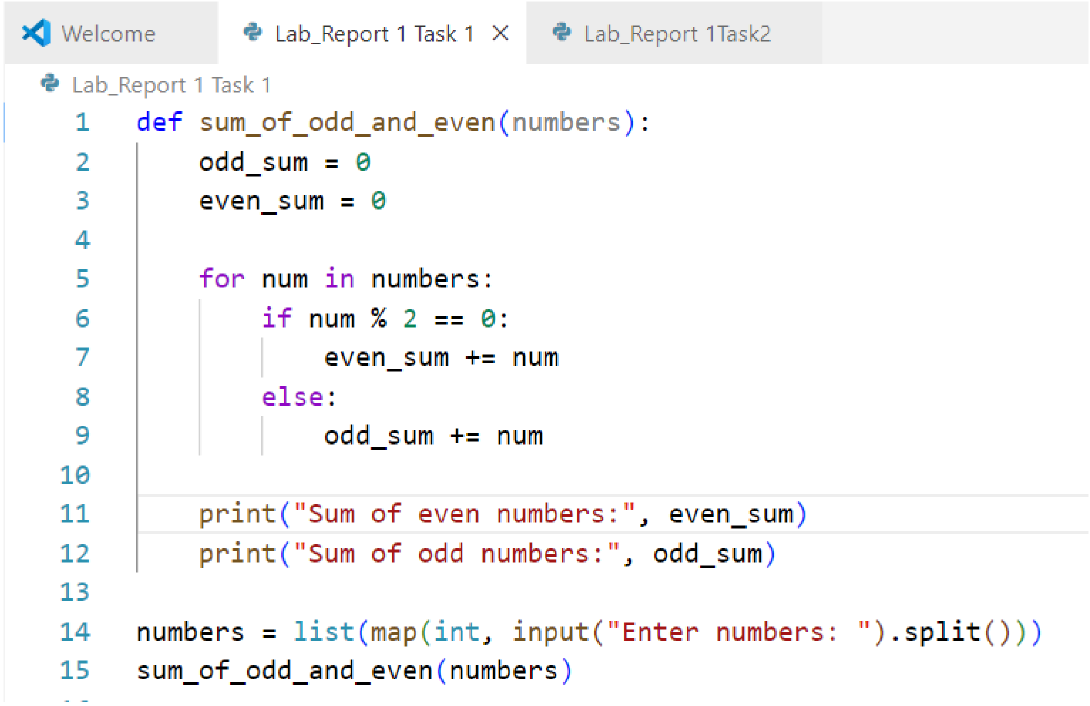
1. Take **input** from the user as a **space-separated** string of numbers.
2. **Convert** the string input to a list of **integers** using the map function.
3. Initialize variables **odd_sum** and **even_sum** to zero.
4. **Iterate** through each number in the list:
   - If the number is divisible by 2 (num % 2 == 0), add it to **even_sum**.
   - Otherwise, add it to **odd_sum**.
5. **Print** the sums of even and odd numbers.

Program 2: Triangle Validity Check

1. **Take Input**:
   Take three float inputs from the user representing the sides of a triangle.
2. **Apply Triangle Inequality Theorem**:
   The sum of any two sides must be greater than the third side for a valid triangle.
3. **Check Conditions**:
   Use conditional statements to check if:
   - $a + b > c$
   - $a + c > b$
   - $b + c > a$
4. **Print Result**:
   Print whether the triangle is valid or not based on the conditions.

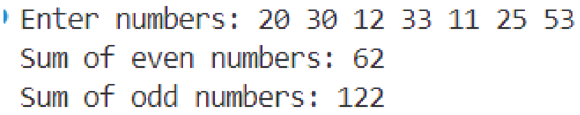
4. IMPLEMENTATION

LAB REPORT 1 TASK 1



```python
def sum_of_odd_and_even(numbers):
    odd_sum = 0
    even_sum = 0

    for num in numbers:
        if num % 2 == 0:
            even_sum += num
        else:
            odd_sum += num

    print("Sum of even numbers:", even_sum)
    print("Sum of odd numbers:", odd_sum)

numbers = list(map(int, input("Enter numbers: ").split()))
sum_of_odd_and_even(numbers)
```

OUTPUT:

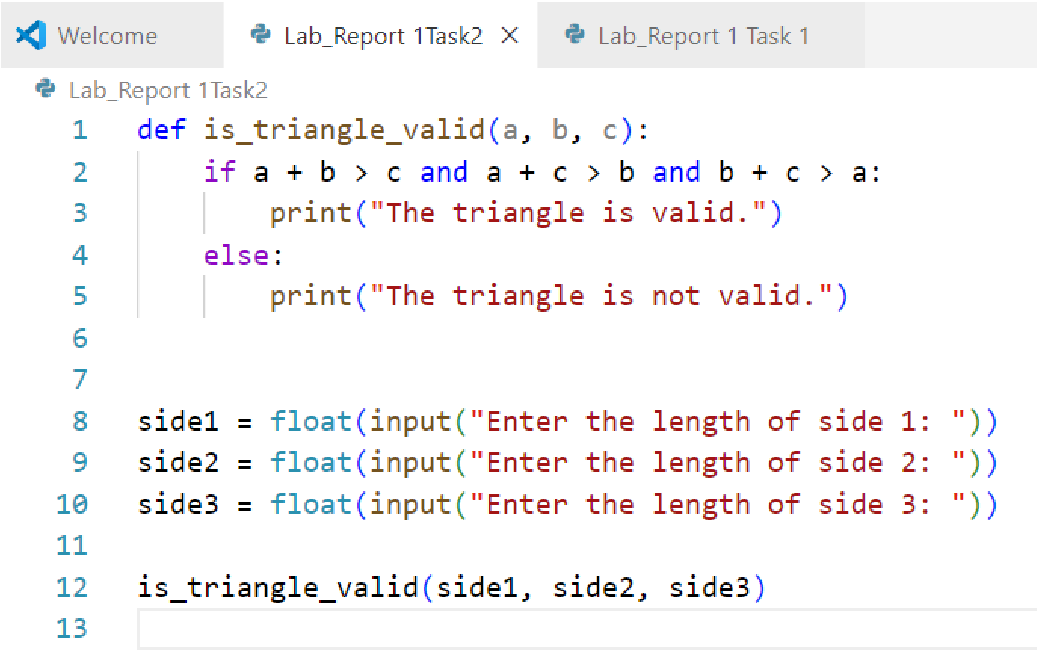


```
Enter numbers: 20 30 12 33 11 25 53
Sum of even numbers: 62
Sum of odd numbers: 122
```

OUTPUT EXPLANATIONS:

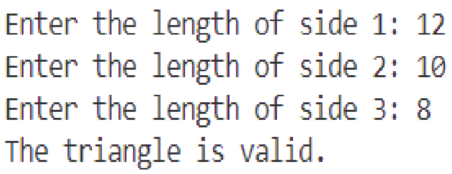
The program identified even numbers (20, 30, 12) and summed them to get 62. It then identified odd numbers (33, 11, 25, 53) and calculated their sum as 122. The final output shows the sums of both even and odd numbers.

LAB REPORT 1 TASK 2

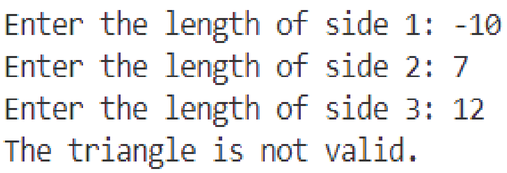


```python
def is_triangle_valid(a, b, c):
    if a + b > c and a + c > b and b + c > a:
        print("The triangle is valid.")
    else:
        print("The triangle is not valid.")


side1 = float(input("Enter the length of side 1: "))
side2 = float(input("Enter the length of side 2: "))
side3 = float(input("Enter the length of side 3: "))

is_triangle_valid(side1, side2, side3)

```

OUTPUT:



```
Enter the length of side 1: 12
Enter the length of side 2: 10
Enter the length of side 3: 8
The triangle is valid.
```



```
Enter the length of side 1: -10
Enter the length of side 2: 7
Enter the length of side 3: 12
The triangle is not valid.
```

OUTPUT EXPLANATIONS:

The program uses the Triangle Inequality Theorem to check if any two sides' sums are greater than the third side. For sides 12, 10, and 8, all three conditions hold true (22 > 8, 20 > 10, 18 > 12). Therefore, the output correctly declares the triangle as valid.

The triangle is invalid because one side is -10, which is impossible in geometry. Additionally, the triangle inequality is violated: -10 + 7 > 12 and -10 + 12 > 7 are both false. Thus, the program correctly concluded the triangle cannot be formed.

6. ANALYSIS AND DISCUSSION

Both programs work well for their intended purposes. The **Sum of Odd and Even Numbers** program correctly separates numbers into odd and even categories, but it lacks input validation, which could lead to errors if non-numeric or empty inputs are provided. Improving input handling would make it more reliable. The **Triangle Validity Check** program successfully checks if the sides form a valid triangle using the triangle inequality theorem. However, it doesn't account for negative or zero-length sides, which could give wrong results, and might face issues with floating-point precision. Both programs demonstrated the practical use of conditionals and loops in solving real-world problems.

7. SUMMARY

The lab experiment involved writing and testing two Python programs. The first program calculated sums of odd and even numbers from a list, while the second program determined the validity of a triangle. Both programs were implemented and tested successfully.