

10/9/25

Task: 0.7

Implement conditional, control and looping statements
Aim: to implement conditional control and looping statement using Python.

Algorithm:

1. Start the program

2. print a welcome message: outputs a simple greeting

3. Determining and print the number of students uses `len()` to find the number of element in the `Students_name` list

4. print the type of list: uses `type()` to show the type of the `student_names` and `student_grades` lists.

5. Find and print high and lowest grades: uses `max()` and `min()` and to determine the highest and lowest value in students.

6. print sorted list of grades: uses `sorted()` to sort the grades.

7. print reversed list of grades: uses `reversed()` to reverse the sorted list and convert it is a list

8. Generate and print a range indices uses `range()` to create a list of indices from 0 to the number of students

9. stop.

Program:

```
def analyze-student-grades():  
#sample data
```



```
Student-name = ("Alica", "Bob", "Charlie", "Diane")  
Student-grades = [85, 92, 78, 90]
```

```
# 1. print a welcome message.
```

```
print("welcome to the Student Grades Analyzer!!\n")
```

```
# 2. Determine and print the number of students
```

```
num_students = len(student-names)
```

```
print("Number of students", num-students)
```

```
# 3. Print the type of the student names list and  
the grade list.
```

```
print("\nType of Student-name list:", type(student-  
name))
```

```
print("Type of Student-grades list:", type(student-  
-grades))
```

```
# 4. Find and print the highest and lowest grade
```

```
highest-grade = max(student-grades)
```

```
lowest-grade = min(student-grades)
```

```
print("\n Highest grade", highest-grade)
```

```
print("lowest grade", lowest-grade)
```

```
# 5. Print the list of grades
```

```
sorted-grades = sorted(student-grades)
```

```
print("\nsorted grades", sorted-grades)
```

```
# 6. print the list of grades in reverse order
```

```
reversed-grades = list(reversed(sorted-grades))
```

```
print("Reversed grades", reversed-grades).
```


output:

Enter the starting no: 1

enter the ending no: 50

enter the step value: 5

Output:

welcome to the student's grades analyzer:

no of students = 4

Type of student names list: <class 'list'>

Type of student-grades list: <class 'list'>

Highest grade = 92

Lowest grade = 78

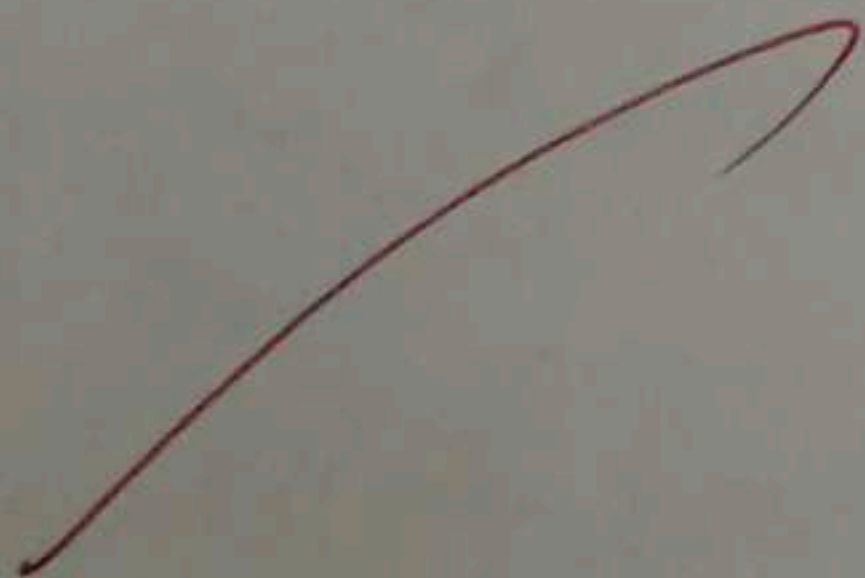
Sorted grades: [78, 85, 90, 92]

Reversed grades: [92, 90, 85, 78]

Grade indices from 1 to number of students:
[1, 2, 3, 4]


```
# 7. Generate and print a range of grade indices  
from 1 to the number of student grade-indices  
= list(range(1, num-student+1))  
print("In Grade indices from 1 to number of  
student's grade-indices")
```

```
# Run the analysis  
analyze-student-grades()
```



Task: #2

Aim: It is to create small Calculator application to help users perform basic arithmetic operations.

Algorithm:

1. Start the program
2. User Input for Number: The program prompts the user to enter two numbers.
3. User Input for operation the program prompts the user to choose an arithmetic operation.
4. perform operation based on the user choice the program perform the chosen arithmetic operation using the defined function.
5. display Result: The program display the result of the operation.
6. Stop.

program:

```
def add(a,b):  
    """Return the sum of two number: """  
    return a+b  
def subtract(a,b):  
    """Return the difference between two numbers"""  
    return a-b  
def multiply(a,b)  
    """Return the product of two number """
```



```

return a+b
def divide (a,b):
    """ Return the quotient of two number Hardless
    division by zero """
    if b!=0
        return a/b
    else :
        return "Error Division by zero".
def greet(name):
    return f"Hello {name}! welcome to program"
def main()
# Demonstrating the uses of user-defined function
# Arithmetic operations.
num1 = 10
num2 = 5
print("Arithmetic operations:")
print(f"Sum of {num1} and {num2} : "add {num1,
num2})
print(f" Difference between {num1} and {num2} :
Sub tract (num1,num2)"
print(f"product of {num1} and {num2} : "multiply
(num1,num2))
# Greeting the user
user-name = "Alice"
print("In Greeting.")
print(gree (user-name))

```


output:

Arithmetic operations

Sum of 10 and 5: 15

Difference between 10 and 5: 5

product of 10 and 5: 50

Quotient of 10 and 5: 2

greeting:

Hello, Alice! welcome to the program


```
#Run the main function  
if __name__ == "__main__":  
    main()
```

VEL TECH	
EX No.	
PERFORMANCE (5)	7
RESULT AND ANALYSIS (5)	5
VIVA VOCE (3)	3
RECORD (4)	3
TOTAL (15)	4
SIGN WITH DATE	15

Result: Thus the python program using function's concept was successfully executed and the output was verified.