Task. 7 Utilizing 'Function's' concepts in python Programming.

Aim:-

To write the python program using 'Functins' concepts in python Programming.

7.1:- You are developing a small Python script to analyze and manipulate a list of student grade for a class project. Write a Python program that satisfies the above requirements using the built-in fnctions print(), len(), max(), min() sorted(), reversed(), and range().

Algorithm:

1. Start the program.
2. Print a welcome message: outputs a simple greeting.
3. Determine and print the number of students: Uses len() to find the number of elements in the students_names list.
4. Print the type of lists: Uses type() to show the types of the student_names and student_grade lists.
5. Find and prints highest and lowest grades: Uses max() and min() to determine the highest and lowest value in student_grades.
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 converts it to a list.
8. Generate and print a range of grade indices: Uses range() to create a list of indices from 1

to the number of students

9. Stop.

Program:

```
def analyze_student_grade():
# Sample data
student_names = ["Alice", "Bob", "Charlie", "Daina"]
student_grade = [85,92,78,90]
# 1. Print a welcome message
print("Welcome to the Student Grade 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 grades list.
print("\nType of student_name list: ", type(student_
     names))
print("Type of student_grade list: ", type(student_grades))
# 4. Find and print the highest and lowest grade.
highest_grade=max(student_grade]
lowest_grade = min(student_grade)
print("\nHighest grade: ", highest_grade)
print("Lowest grade:", lowest_grade)
# 5. Print the list of grade storted in ascending
     order.
sorted_grade= sorted(student_grades)
print("\nSorted grades:", sorted_grade)
#6. Print the list of grade in reverse order.
reversed_grade = list (reversed(sorted_grade))
print("Reversed grades:", reversed_grades)
```

Output:-

```
Welcome to the students Grades Analyzer!
Number of students: 4
Type of students_name list: <class 'list'>
Type of student_grade list: <class 'list'>
Highest grade: 92
Lowest grade: 78
sorted grades: [78, 85, 90, 92]
Reversed grades: [92, 90, 85, 78]
Grade indicate from 1 to number of student: [1,2,3,4]
```

```
#7. Generate and print a range of grade
indicates from 1 to the number of students
grade_indices = list(range(1, num_students + 1))
print("\n Grade indices from 1 to number of
        students:", grade_indicos)

# Run the analysis
analyze_student_grades()
```

7.2. You are tasked with creating a small calculator application to help users perform basic arithmetic operations and greet them with a personalized message: Your application should perform the following tasks: addition, subtraction, multiplication, division.

Algorithm:

1. Start the program.
2. User Input for Numbers: The program prompts the user to enter two numbers.
3. User Inputs for operation: The program prompts the user to choose an arithmetic operation (addition, subtraction, multiplication, division)
4. Perform Operation: Based on the user's choice, the perform the chosen arithmetic operation usin the defined fonctions.
5. Display Result: The program display the result of the operation.
6. stop.

Program:

```
def add (a,b)
"""" Retorn the som of two numbers""""

return a+b.
def subtract (a, b):
"""" Retorn the difference between two numbers.""""
return a-b.
def multiply (a,b):
"""" Retorn the product of two numbers""""
return a*b
def divide (a, b):
```

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: 2.0

Greeting:
Hello, Alice! welcome to the program.
```

```
"""" Return the quotient of two numbers: Handles
     division by zero""""
if b! =0
     return a/b
else:
    return "Error: Division by zero"
def greet(name):
"""" Return a greeting message for the user""""
return f"Hello, {name}! welcome to the program.
def main():
# Demonstrating the use of user-defined function.
# Arithmetic operations.
num 1 =10
num 2=5
print("Arithmetic Operations:")
print(f"Sum of {num1} and {num2}:", add(num1, num2))
print(f"Difference between {num1} and {num2}:",
        subtract(num1, num2)).
print(f"Product of {num1} and {num2}: " multipy
        (num1, num2))
print(f"Quotient of {num1} and {num2}:", divide(num1,
        num2)
# Greeting the user
user_name = "Alice"
print("\n Greeting:")
print(greet(user_name))

# Run the main function.
if_name_=="_main_".
  main()
```

Result:-

Thus, the python program using 'Functions' concepts was successfully executed and the outputs was verified..