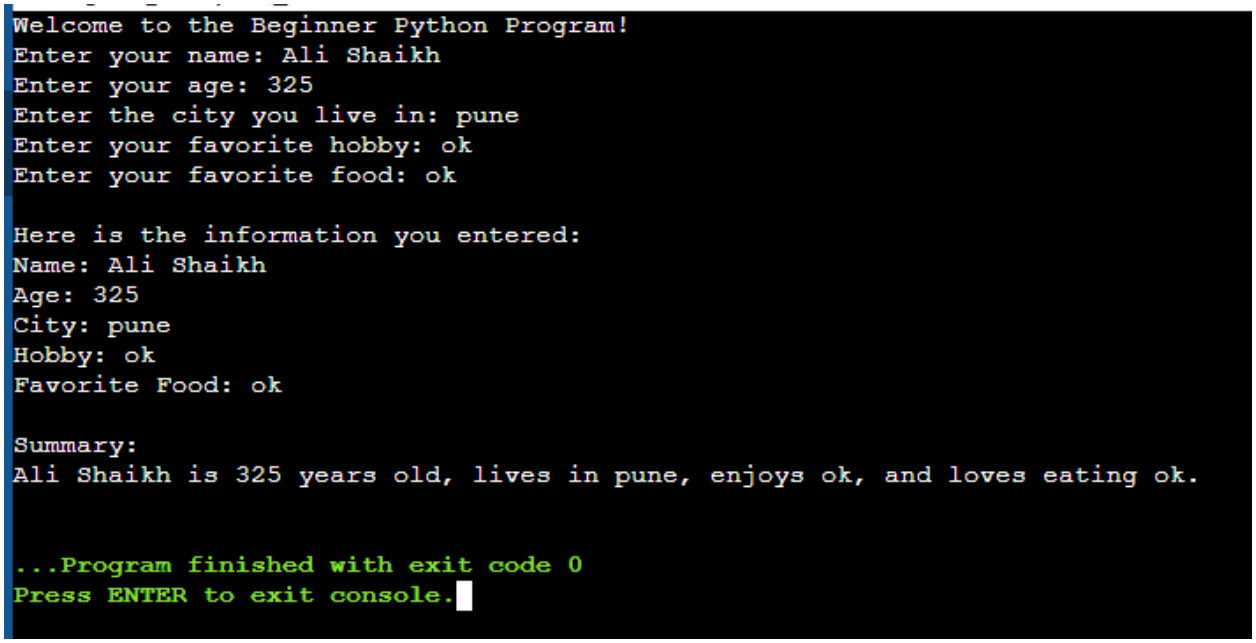
Python
RollNo: SCOD09

1. Write a program by using basic concepts like input, output, variable, keywords, and identifiers.
   Code:

```
print("Welcome to the Beginner Python Program!")

name = input("Enter your name: ")
age = input("Enter your age: ")
city = input("Enter the city you live in: ")
hobby = input("Enter your favorite hobby: ")
food = input("Enter your favorite food: ")

print("\nHere is the information you entered:")
print("Name:", name)
print("Age:", age)
print("City:", city)
print("Hobby:", hobby)
print("Favorite Food:", food)

print("\nSummary:")
print(name, "is", age, "years old, lives in", city + ", enjoys", hobby + ", and loves eating", food + ".")
```

   Output:-



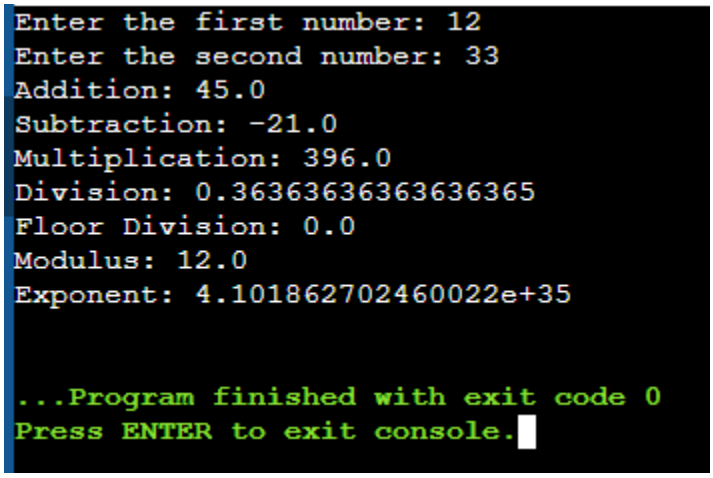
Python
RollNo: SCOD09

2. Write a program to demonstrate the of arithmetic operators

Code:-

```
num1 = float(input("Enter the first number: "))
num2 = float(input("Enter the second number: "))

addition = num1 + num2
subtraction = num1 - num2
multiplication = num1 * num2
division = num1 / num2
floor_division = num1 // num2
modulus = num1 % num2
exponent = num1 ** num2

print("Addition:", addition)
print("Subtraction:", subtraction)
print("Multiplication:", multiplication)
print("Division:", division)
print("Floor Division:", floor_division)
print("Modulus:", modulus)
print("Exponent:", exponent)
```

Output:-



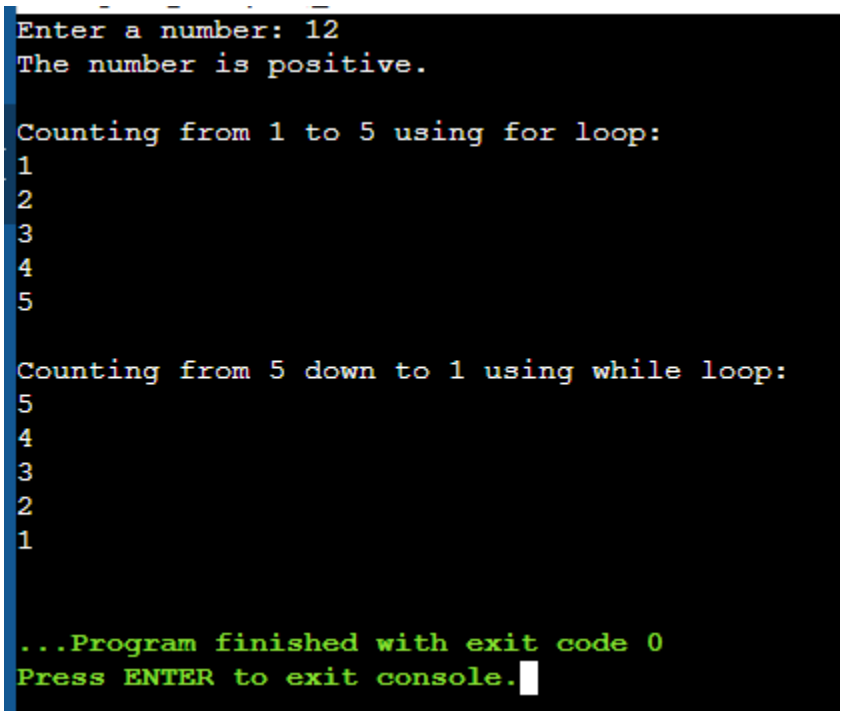
Python
RollNo: SCOD09

3. Write a program to demonstrate working of control structures.

Code:-

```
number = int(input("Enter a number: "))
if number > 0:
   print("The number is positive.")
elif number < 0:
   print("The number is negative.")
else:
   print("The number is zero.")

print("\nCounting from 1 to 5 using for loop:")
for i in range(1, 6):
   print(i)

print("\nCounting from 5 down to 1 using while loop:")
count = 5
while count > 0:
   print(count)
   count -= 1
```

Output:-



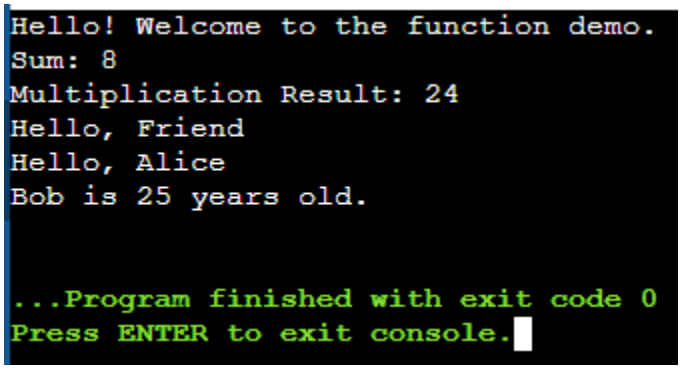
Python
RollNo: SCOD09

4. Write a program to demonstrate working of different types of functions

Code:-

```
# Function with no parameters and no return value
def greet():
    print("Hello! Welcome to the function demo.")

# Function with parameters but no return value
def add_numbers(a, b):
    print("Sum:", a + b)

# Function with parameters and a return value
def multiply_numbers(a, b):
    return a * b

# Function with default parameter
def greet_person(name="Friend"):
    print("Hello,", name)

# Function with keyword arguments
def describe_person(name, age):
    print(name, "is", age, "years old.")

# Calling the functions
greet()
add_numbers(5, 3)
result = multiply_numbers(4, 6)
print("Multiplication Result:", result)
greet_person()
greet_person("Alice")
describe_person(age=25, name="Bob")
```

Output:



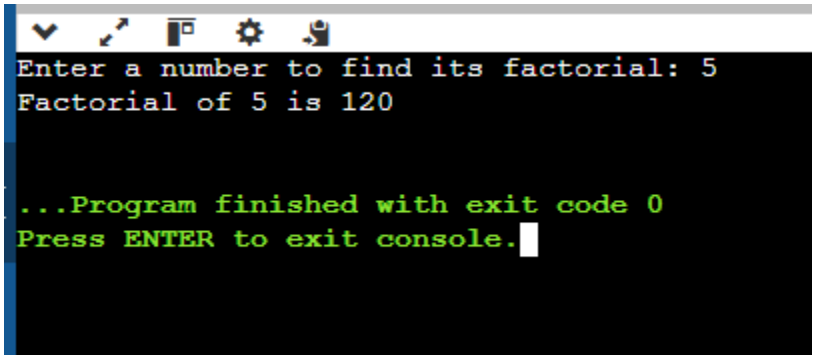
Python
RollNo: SCOD09

5. Write a program to demonstrate working of recursion.

Code:-

```
def factorial(n):
    if n == 0 or n == 1:
        return 1
    else:
        return n * factorial(n - 1)

num = int(input("Enter a number to find its factorial: "))
result = factorial(num)
print("Factorial of", num, "is", result)
```

Output:-



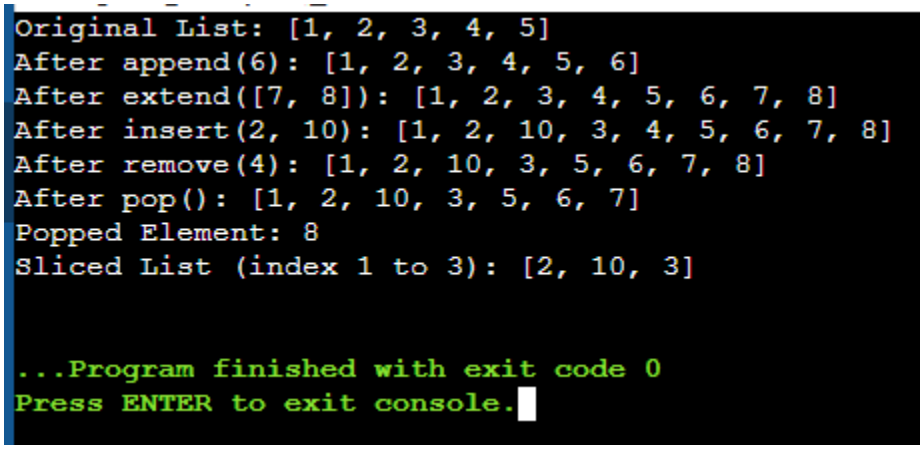
Python
RollNo: SCOD09

6. Write a program to perform different operations on list: i. Append ii. Extend iii. Insert iv. Remove v. Pop vi. Slice

Code:-

```
# Create a list
my_list = [1, 2, 3, 4, 5]
print("Original List:", my_list)

# i. Append - add an element at the end
my_list.append(6)
print("After append(6):", my_list)

# ii. Extend - add multiple elements at the end
my_list.extend([7, 8])
print("After extend([7, 8]):", my_list)

# iii. Insert - insert an element at a specific position
my_list.insert(2, 10)  # insert 10 at index 2
print("After insert(2, 10):", my_list)

# iv. Remove - remove a specific element
my_list.remove(4)
print("After remove(4):", my_list)

popped_element = my_list.pop()
print("After pop():", my_list)
print("Popped Element:", popped_element)

sliced_list = my_list[1:4]  # elements from index 1 to 3
print("Sliced List (index 1 to 3):", sliced_list)
```

Output:-



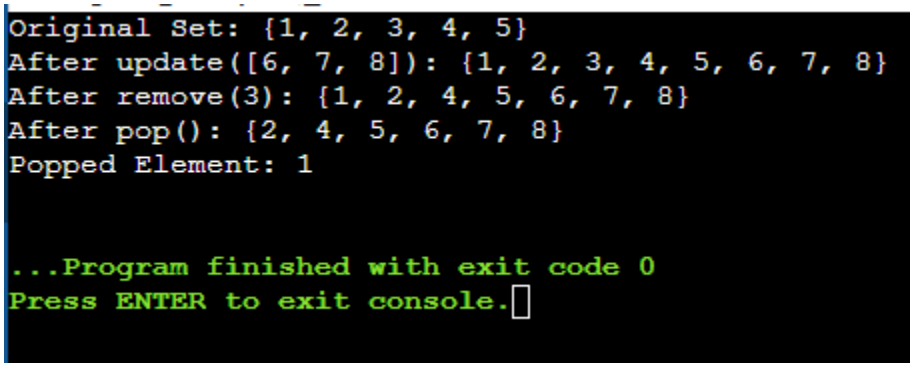
Python
RollNo: SCOD09

7. Write a program to perform different operations on set: i. Update ii. Remove the element iii. Clear iv. Pop

Code:-

```
# Create a set
my_set = {1, 2, 3, 4, 5}
print("Original Set:", my_set)

# i. Update - add multiple elements to the set
my_set.update([6, 7, 8])
print("After update([6, 7, 8]):", my_set)

# ii. Remove - remove a specific element
my_set.remove(3)
print("After remove(3):", my_set)

# iv. Pop - remove and return an arbitrary element
popped_element = my_set.pop()
print("After pop():", my_set)
print("Popped Element:", popped_element)
```

Output:-



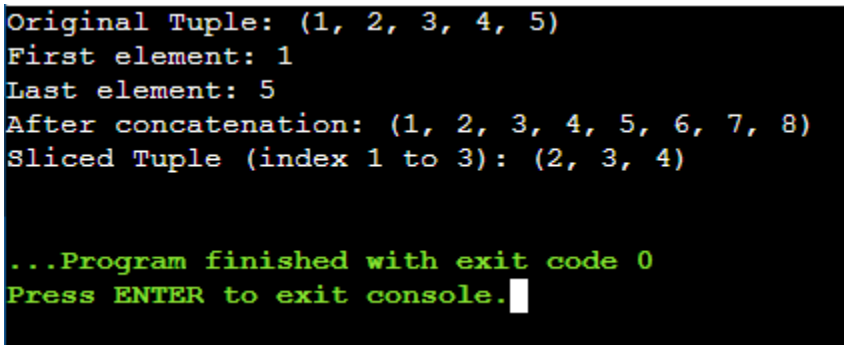
Python
RollNo: SCOD09

8. Write a program to perform different operations on tuple: i. Accessing ii. Concatenation iii. Slicing iv. Deleting

Code:-

```
# Create a tuple
my_tuple = (1, 2, 3, 4, 5)
print("Original Tuple:", my_tuple)

# i. Accessing - access elements using index
print("First element:", my_tuple[0])
print("Last element:", my_tuple[-1])

# ii. Concatenation - combine two tuples
new_tuple = my_tuple + (6, 7, 8)
print("After concatenation:", new_tuple)

# iii. Slicing - get a part of the tuple
sliced_tuple = my_tuple[1:4]  # elements from index 1 to 3
print("Sliced Tuple (index 1 to 3):", sliced_tuple)

# iv. Deleting - delete the entire tuple
del my_tuple
# print(my_tuple)  # This will cause an error because the tuple is deleted
```

Output:-



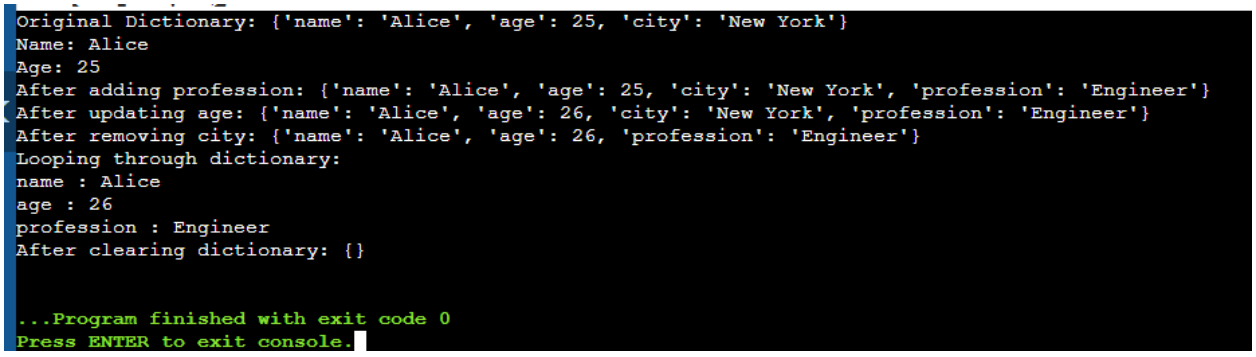
Python
RollNo: SCOD09

9. Write a program to demonstrate the working of dictionary.

Code:-

```
# Create a dictionary
my_dict = {
   "name": "Alice",
   "age": 25,
   "city": "New York"
}
print("Original Dictionary:", my_dict)

# Accessing elements
print("Name:", my_dict["name"])
print("Age:", my_dict.get("age"))

# Adding new element
my_dict["profession"] = "Engineer"
print("After adding profession:", my_dict)

my_dict["age"] = 26
print("After updating age:", my_dict)

my_dict.pop("city")
print("After removing city:", my_dict)

print("Looping through dictionary:")
for key, value in my_dict.items():
   print(key, ":", value)

my_dict.clear()
print("After clearing dictionary:", my_dict)
```

Output:-



Python
RollNo: SCOD09

10. Write a program demonstrates various operations on strings: i. Slicing ii. Concatenating iii. Finding the length of the string iv. Converting the string to uppercase and lowercase v. Replacing a substring with another substring vi. Splitting the string into a list of substrings.

Code:-

```
my_string = "Hello, Python Programming!"
print("Original String:", my_string)

sliced_string = my_string[7:13]  # characters from index 7 to 12
print("Sliced String:", sliced_string)

new_string = my_string + " Let's learn."
print("After Concatenation:", new_string)

length = len(my_string)
print("Length of the string:", length)

print("Uppercase:", my_string.upper())
print("Lowercase:", my_string.lower())

replaced_string = my_string.replace("Python", "Java")
print("After Replacing:", replaced_string)

split_list = my_string.split()
print("Split into list:", split_list)
```

Output:-

Python
RollNo: SCOD09

11. Write a program to demonstrate different operations on file: i. Create ii. Open iii. Read iv. Write v. Update vi. Delete

Code:-

```
import os
file = open("example.txt", "w")
file.write("This is the first line.\n")
file.write("This is the second line.\n")
file.close()
print("File created and written successfully.")
file = open("example.txt", "r")
content = file.read()
file.close()
print("\nContent of the file:")
print(content)
file = open("example.txt", "w")
file.write("This is new content, overwriting old content.\n")
file.close()
print("\nFile overwritten successfully.")
file = open("example.txt", "a")
file.write("Appending this line to the file.\n")
file.close()
print("Content appended successfully.")
file = open("example.txt", "r")
content = file.read()
file.close()
print("\nUpdated Content of the file:")
print(content)
os.remove("example.txt")
print("File deleted successfully.")
```

Output:-