1. Write a Python program for sequential search.

```
def linear_Search(list1, n, key):

# Searching list1 sequentially
for i in range(0, n):
        if (list1[i] == key):
                return i
return -1


list1 = [1 ,3, 5, 4, 7, 9]
key = 7

n = len(list1)
res = linear_Search(list1, n, key)
if(res == -1):
print("Element not found")
else:
print("Element found at index: ", res)
```

2. Create a simple calculator program in python.

```
# Program make a simple calculator

def add(x, y):
return x + y

# This function subtracts two numbers
def subtract(x, y):
return x - y

# This function multiplies two numbers
def multiply(x, y):
return x * y

# This function divides two numbers
def divide(x, y):
return x / y


print("Select operation.")
print("1.Add")
```

```
print("2.Subtract")
print("3.Multiply")
print("4.Divide")

while True:
# take input from the user
choice = input("Enter choice(1/2/3/4): ")

# check if choice is one of the four options
if choice in ('1', '2', '3', '4'):
num1 = float(input("Enter first number: "))
num2 = float(input("Enter second number: "))

if choice == '1':
        print(num1, "+", num2, "=", add(num1, num2))

elif choice == '2':
        print(num1, "-", num2, "=", subtract(num1, num2))

elif choice == '3':
        print(num1, "*", num2, "=", multiply(num1, num2))
```

```
elif choice == '4':
        print(num1, "/", num2, "=", divide(num1, num2))

# check if user wants another calculation
# break the while loop if answer is no
next_calculation = input("Let's do next calculation? (yes/no): ")
if next_calculation == "no":
break

else:
print("Invalid Input")
```

3. How to create a string in Python?

```
# create string type variables

name = "Python"
print(name)

message = "I love Python."
print(message)
```

4. Create a program in Python for Implement Selection Sort.

```
def selection_sort(array):
```

```
length = len(array)

for i in range(length-1):
        minIndex = i

        for j in range(i+1, length):
                if array[j]<array[minIndex]:
                minIndex = j

        array[i], array[minIndex] = array[minIndex], array[i]


return array
array = [21,6,9,33,3]

print("The sorted array is: ", selection_sort(array))
```

5. Create a program in Python for Python program to implement a stack.

```
class Stack_struct:
def __init__(self):
        self.items = []

def check_empty(self):
        return self.items == []

def add_elements(self, my_data):
        self.items.append(my_data)

```

```
def delete_elements(self):
    return self.items.pop()


my_instance = Stack_struct()
while True:
print('Push <value>')
print('Pop')
print('Quit')
my_input = input('What operation would you like to perform ? ').split()

my_op = my_input[0].strip().lower()
if my_op == 'push':
    my_instance.add_elements(int(my_input[1]))
elif my_op == 'pop':
    if my_instance.check_empty():
    print('The stack is empty')
    else:
    print('The deleted value is : ', my_instance.delete_elements())
elif my_op == 'Quit':
    break
```

6. Create a program in Python for Demonstrate usage of basic regular expression(any 5).

a. import re

b. pattern = '^a...s$'
c. test_string = 'abyss'

d. result = re.match(pattern, test_string)

e. if result:
    i. print("Search successful.")
f. else:
    i. print("Search unsuccessful.")

7. Write a function that computes and returns the size/length of a list in python.

```
ListName = [ "Hello", "Edureka", 1,2,3 ]
print ("The list is : " + str(ListName))
counter = 0
for i in ListName:
counter = counter + 1
print ("Length of list using naive method is : " + str(counter))
```

8. Create a program in Python for Demonstrate use of Dictionaries (Insert and Delete the dictionary value).

```
# Changing and adding Dictionary Elements
my_dict = {'name': 'Jack', 'age': 26}

# update value
my_dict['age'] = 27

#Output: {'age': 27, 'name': 'Jack'}
print(my_dict)

# add item
my_dict['address'] = 'Downtown'

# Output: {'address': 'Downtown', 'age': 27, 'name': 'Jack'}
print(my_dict)
```

9. Create a program in Python for Reading and Writing a CSV file.

```
import csv

with open('employee_birthday.txt') as csv_file:
    csv_reader = csv.reader(csv_file, delimiter=',')
    line_count = 0
    for row in csv_reader:
        if line_count == 0:
            print(f'Column names are {", ".join(row)}')
            line_count += 1
        else:
            print(f'\t{row[0]} works in the {row[1]}
            department, and was born in {row[2]}.')
            line_count += 1
    print(f'Processed {line_count} lines.')
```

10. Write a program for connectivity between python and mysql and insert some records.

```
import mysql.connector
db_connection = mysql.connector.connect(
host= "localhost",
user= "root",
passwd= "root"
)
# creating database_cursor to perform SQL operation
db_cursor = db_connection.cursor()
# executing cursor with execute method and pass SQL query
db_cursor.execute("CREATE DATABASE my_first_db")
# get list of all databases
db_cursor.execute("SHOW DATABASES")
#print all databases
```

11. for db in db_cursor:
    a. print(db)