WORKSHEET - 3.1

Student Name: Udit Gupta   **UID: 21BCS9091**

Branch: CSE   **Section/Group: 720 - B**

Semester: 4TH   **Date of Performance: 28/Apr/2023**

Subject Name: PROGRAMMING IN PYTHON   Subject Code: 21CSP-259

1. **Aim**: Programs to demonstrate searching and sorting using lists in python.

   - Python Program to implement Linear and Binary Search.
   - Python Program to implement Bubble Sort.
   - Python Program to implement Selection Sort.

2. **Source Code:**

(a) WAP to implement Binary and Linear Search

```python
# Function for linearSearch!
def linearSearch(array, key):
   # Iterating through the list to find key
   for i in range(len(array)):
     if(array[i] == key):
         return i
   return -1


# Function for BinarySearch!
def binarySearch(array, key):
   start = 0              # Index pointing at first index of list
   end = len(array)-1     # Index pointing at last index of list
   mid = (start + end)//2    # Calculating middle index of list

   while(start <= end):
     # Checking basic conditions of binary search
     if(array[mid] == key):
        return mid
     elif(array[mid] > key):
        end = mid - 1
     else:
        start = mid + 1
        mid = (start + end)//2

   return -1
```

```
# ----Main---

print("This program is programmed by Udit Gupta, 21BCS9091")
array = []
num = int(input("Enter number of elements in list: "))

i = 0
while(i < num):
    # Using try and except block to avoid errors
    try:
        x = int(input("Enter number: "))
        array.append(x)
        i = i + 1
    except:
        print("Enter Integer only!")
        continue

key = int(input("Enter the element you want to search: "))
print("Your element is found at: ", linearSearch(array, key),"th index" )
array.sort()        # Sorting the array before passing it to BinarySearch function
print("Your element is found at: ", binarySearch(array, key),"th index" )
```

- **Output**

```
This program is programmed by Udit Gupta, 21BCS9091
Enter number of elements in list: 5
Enter number: String
Enter Integer only!
Enter number: 4
Enter number: 2
Enter number: 6
Enter number: 1
Enter number: 0
Enter the element you want to search: 0
Your element is found at:  4 th index
Your element is found at:  0 th index
```

(b) WAP to implement Selection Sort

```
array = []
print("This program is programmed by Udit Gupta, 21BCS9091")
n = int(input("Enter number of elements: "))
i = 0

#Using while loop for getting inputs from user
while(i < n):
```

```python
    try:
        x = int(input("Enter elements: ")
        array.append(x)
        i = i + 1
    except:
        print("Enter integers only!")

# Using loop to perform selection sort to sort the array
for i in range(0, len(array)):
    min = i
    for j in range(i+1, len(array)):
        if(array[j] < array[min]):
            min = j
    (array[i], array[min]) =(array[min], array[i])
print(array)
```

- **Output**

```
 ↳   This program is programmed by Udit Gupta, 21BCS9091
     Enter number of elements: 5
     Enter elements: 10
     Enter elements: 8
     Enter elements: 11
     Enter elements: 2
     Enter elements: dgd
     Enter integers only!
     Enter elements: 0
     [0, 2, 8, 10, 11]
```

(c) WAP to implement Bubble Sort

```python
array = []
print("This program is programmed by Udit Gupta, 21BCS9091")
n = int(input("Enter number of elements: "))
i = 0

#Using while loop for getting inputs from user
while(i < n):
    try:
        x = int(input("Enter elements: "))
        array.append(x)
        i = i + 1
    except:
        print("Enter integers only!")

swapped = False
# Using loop to perform selection sort to sort the array
```

```
for i in range(len(array)-1):
    for j in range(0, len(array) - i - 1):
        if(array[j] > array[j+1]):
            swapped = True
            (array[j], array[j+1]) = (array[j+1], array[j])

print(array)
```

- **Output**

```
    This program is programmed by Udit Gupta, 21BCS9091
    Enter number of elements: 5
    Enter elements: ABX
    Enter integers only!
    Enter elements: 5
    Enter elements: 4
    Enter elements: 3
    Enter elements: 2
    Enter elements: 1
    [1, 2, 3, 4, 5]
```