Lab-2 - Assignment

1. Implement a simple Guess the Number game in Python. In the game, the user has to guess a randomly generated number. Use branching, looping, and flow control statements to manage the game's flow. Some salient features of the game implementation is as follows:
    a) Use the random module to generate a random number between a specified range. This will be the number the user needs to guess.
    b) Use a while loop to repeatedly prompt the user for their guess until they correctly guess the number.
    c) Use if..else statements to check if the user's guess is correct, too high, or too low. Provide appropriate feedback.
    d) Introduce flow control statements like continue to skip certain parts of the loop or break to exit the loop when the correct guess is made.

Code:

```
import random

a = (int)(input("Enter the first number: "))
b = (int)(input("Enter the second number: "))

r = random.randint(a, b)

while(True):
  n = (int)(input ("Guess a number in the range "))
  if (n > r):
    print("Guess is too high")
  elif(n < r):
    print("Guess is too low")
  else:
    print("Correct guess")
    break
```

2. Implement a user-defined module yoursearch using Python for the following two search algorithms
    a) Linear Search
    b) Binary Search

Further, implement another user-defined module yoursort using Python for the following three sort algorithms

    c) Bubble Sort
    d) Insertion Sort

e) Selection Sort

Finally, implement a driver program where you import both the user-defined modules. Take input from user for key elements and use inbuilt functions from random module for creating the list of elements.

Code

yoursearch.py

```
def linear_search(arr, key):
    for i in range(len(arr)):
        if arr[i] == key:
            return i
    return -1

def binary_search(arr, key):
    low, high = 0, len(arr) - 1
    while low <= high:
        mid = (low + high) / 2
        if arr[mid] == key:
            return mid
        elif arr[mid] < key:
            low = mid + 1
        else:
            high = mid - 1
    return -1
```

yoursort.py

```
def bubble_sort(arr):
    n = len(arr)
    for i in range(n):
        for j in range(0, n-i-1):
            if arr[j] > arr[j+1]:
                arr[j], arr[j+1] = arr[j+1], arr[j]

def insertion_sort(arr):
    for i in range(1, len(arr)):
        key = arr[i]
        j = i - 1
        while j >= 0 and key < arr[j]:
            arr[j + 1] = arr[j]
            j -= 1
        arr[j + 1] = key

def selection_sort(arr):
```

```
    n = len(arr)
    for i in range(n):
        min_idx = i
        for j in range(i+1, n):
            if arr[j] < arr[min_idx]:
                min_idx = j
        arr[i], arr[min_idx] = arr[min_idx], arr[i]
```

main.py

```
import yoursearch
import yoursort
import random

r = (int)(input("Enter the range of the array :" ))
arr=[]
for i in range(r):
    arr.append(random.randint(1, 100))
print("The Original Array is ", arr)

print("1. Bubble Sort")
print("2. Insertion Sort")
print("3. Selection Sort")
print("4. Linear Search")
print("5. Binary Search")

ch = (int)(input("Enter your choice : "))

if (ch == 1):
    yoursort.bubble_sort(arr)
    print("The Sorted Array is ", arr)
elif (ch == 2):
    yoursort.insertion_sort(arr)
    print("The Sorted Array is ", arr)
elif (ch == 3):
    yoursort.selection_sort(arr)
    print("The Sorted Array is ", arr)
elif (ch == 4):
    key = (int)(input("Enter the number to be searched : "))
    res = yoursearch.linear_search(arr, key)
    if (res == -1):
        print("The number is not found")
    else:
        print("The number is found at index ", res)
elif (ch == 5):
    key = (int)(input("Enter the number to be searched : "))
```

```
    res = yoursearch.binary_search(arr, key)
    if (res == -1):
        print("The number is not found")
    else:
        print("The number is found at index ", res)
else:
    print("Invalid Choice")
```