Computational Thinking Using Python – CSE1500

Lab sheet – 2.3

Student Activity: In the previous lab sheet, Linear and Binary search were implemented using Lists. The same problems can also be solved using arrays (as follows). Only the array declaration part (highlighted in yellow colour) is different; the rest of the logic is the same

Problem:

A teacher has a record of student roll numbers who submitted an assignment. The teacher wants to **check whether a particular student has submitted the assignment** or not.

Write a Python program that:

1. Accepts the total number of students and their roll numbers.
2. Accepts the roll number to search for.
3. Uses **linear search** to determine whether the roll number is present.
4. Displays the result: "Student has submitted the assignment" or "Student has not submitted the assignment."

```
# Program to check if a student has submitted the assignment using Linear Search
# Read total number of students

from array import *
n = int(input("Enter number of students: "))
# create an empty integer array
roll_numbers= array('i', [])

# Read roll numbers of students
print("Enter roll numbers of students:")
for i in range(n):
    roll = int(input("enter roll number  "))
    roll_numbers.append(roll)

# Read roll number to search
search_roll = int(input("Enter roll number to search: "))

# Initialize a flag to indicate if roll number is found
found = False

# Linear Search
for roll in roll_numbers:
    if roll == search_roll:
```

```python
        found = True
        break

# Display result
if found:
    print("Student has submitted the assignment.")
else:
    print("Student has not submitted the assignment.")
```

====================

Problem:

A teacher has a **sorted list of student roll numbers** who have cleared an exam.
The teacher wants to **quickly check whether a particular student has cleared the exam**.

Since the list is sorted, using **binary search** will be faster than checking one by one.

Write a Python program that:

1. Accepts the total number of students and their **sorted roll numbers**.
2. Accepts the roll number to search for.
3. Uses **binary search** to check whether the roll number is present.
4. Displays the result: "Student has cleared the exam" or "Student has not cleared the exam."

```python
# Program to check if a student has cleared the exam using Binary Search
# Read total number of students

from array import *

n = int(input("Enter number of students: "))

# create an empty integer array
roll_numbers= array('i', [])

# Read sorted roll numbers from user
print("Enter roll numbers of students:")
for i in range(n):
    roll = int(input(f"Student {i+1}: "))
    roll_numbers.append(roll)

# sort the array using sorted function
roll_numbers = array('i', sorted(roll_numbers))

# Read roll number to search
search_roll = int(input("Enter roll number to search: "))
```

```python
# Initialize low and high pointers
low = 0
high = n - 1

# Binary Search
found = False
while low <= high:
    mid = (low + high) // 2
    if roll_numbers[mid] == search_roll:
        found = True
        break
    elif search_roll < roll_numbers[mid]:
        high = mid - 1
    else:
        low = mid + 1

# Display sorted array
print("given array in sorted order ")
print(roll_numbers)

# Display result
if found:
    print("Student has cleared the exam.")
else:
    print("Student has not cleared the exam.")
```

=====================