Computational Thinking Using Python – CSE1500

Lab sheet – 2.2

Algorithm: Find Maximum Element in an Array

```
Step 1: Start
Step 2: Read the size of the array n
Step 3: Create an empty array arr
Step 4: Repeat steps (5) and (6) for each element i = 1 to n
        Step 5: Read element arr[i]
        Step 6: Store the element in the array
Step 7: Assume the first element as the maximum
        max_element = arr[0]
Step 8: Repeat for each element from index 1 to n-1
        If arr[i] > max_element then
            max_element = arr[i]
Step 9: Display max_element as the largest element in the array
Step 10: Stop
```

----

1) Problem Statement:

A teacher wants to analyze the performance of students in a class test.
The teacher enters the marks of all students, and the program should find and display the **highest mark** in the class.

```python
# Program to find the maximum marks scored by a student
# without using any built-in functions

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

# create an empty list to store marks
marks = []

# read marks from user
print("Enter marks of each student:")
for i in range(n):
   m = int(input(f"Student {i+1}: "))
   marks.append(m)

# assume first student's mark as maximum
max_mark = marks[0]

# compare each student's mark
```

```
for i in range(1, n):
    if marks[i] > max_mark:
        max_mark = marks[i]

# display result
print("\nHighest mark in the class:", max_mark)
```
===============

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.”

Method 1:

```
# Program to check if a student has submitted the assignment using Linear Search
# Read total number of students
n = int(input("Enter number of students: "))
# Create an empty list to store roll numbers
roll_numbers = []

# 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:
        found = True
        break

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

=========

Method 2:

```
# Read total number of students
n = int(input("Enter number of students: "))

# Create an empty list to store roll numbers
roll_numbers = []

# 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: "))

# Linear Search using for-else
for roll in roll_numbers:
    if roll == search_roll:
        print("Student has submitted the assignment.")
        break
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.”

```
# Program to check if a student has cleared the exam using Binary Search

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

# Create a list to store sorted roll numbers
roll_numbers = []

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

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

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

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

A teacher wants to analyze a sentence to count vowels and consonants for pronunciation practice. Task: Count and display the number of vowels and consonants in the given string. Hint: Use membership operators (in), and string methods like lower().

```
# Program to count vowels and consonants in a given sentence
# read input from user
sentence = input("Enter a sentence: ")
# convert sentence to lowercase for easy comparison
sentence = sentence.lower()
# define vowels
vowels = "aeiou"
# initialize counters
vowel_count = 0
consonant_count = 0
# loop through each character in the sentence
for ch in sentence:
    if ch.isalpha():  # check if character is a letter
        if ch in vowels:
            vowel_count += 1
        else:
            consonant_count += 1
# display results
print("Number of vowels:", vowel_count)
print("Number of consonants:", consonant_count)
```

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