Experiment 5(A)

Student Name: Tanmaya Kumar Pani **UID: 22BCS12986**
Branch: BE-CSE **Section/Group: IOT-613B**
Semester: 5 **Date of Performance: 03/9/24**
Subject Name: Advanced Programming Lab-1 **Subject Code: 22CSP-314**

1. Title: Pangrams

2. Objective:

A *pangram* is a string that contains every letter of the alphabet. Given a sentence determine whether it is a pangram in the English alphabet. Ignore case. Return either pangram or not pangram as appropriate.

3. Algorithm:

a) **Initialize**:
- Create a boolean array letters[26] and initialize all its elements to false. This array will be used to track the presence of each letter of the alphabet.

b) **Convert to Lowercase**:
- Convert the entire input string to lowercase to ensure the check is case-insensitive.

c) **Iterate Through the String**:
- For each character c in the string:
  - **Check if c is a letter**:
    - If c is between 'a' and 'z' (i.e., a lowercase letter):
      - Mark the corresponding index in the letters array as true. The index is calculated as c - 'a', which maps 'a' to index 0, 'b' to index 1, and so on.

d) **Check the Boolean Array**:
- Iterate through the letters array:
  - If any entry in the letters array is false, return "not pangram" immediately. This indicates that at least one letter is missing from the input string.

e) **Return Result**:
- If all entries in the letters array are true, return "pangram". This indicates that the string contains every letter of the alphabet at least once.

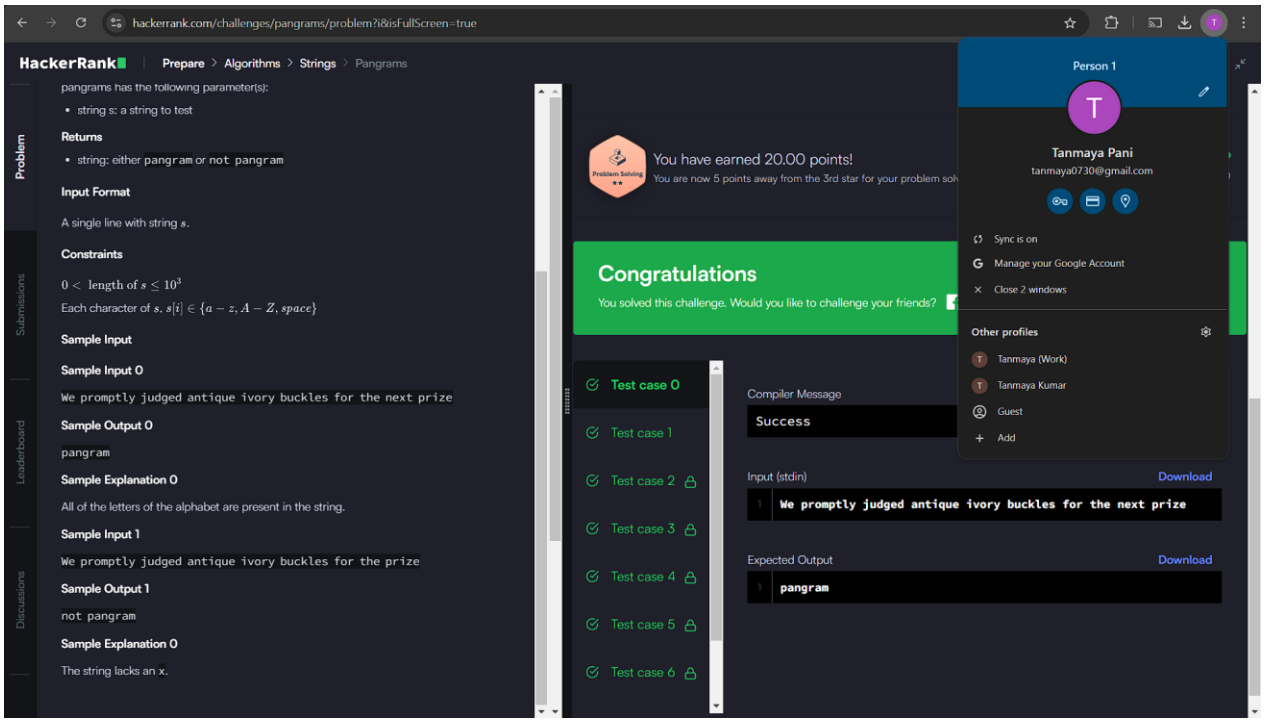
4. Implementation/Code:

```
string pangrams(string s) {

    transform(s.begin(), s.end(), s.begin(), ::tolower);
    bool letters[26]={false};
    for(char ch : s){
            if(ch>='a' && ch<='z'){
                letters[ch - 'a'] = true;
            }
    }
    for(bool letterpresent : letters){
        if(!letterpresent){
            return "not pangram";
        }
    }
    return "pangram";

}
```

5. Output:



6. Learning Outcomes:

• **Understanding Pangrams**: You learned how to identify a pangram by checking if a string contains all the letters of the English alphabet.
• **String Manipulation**: You practiced converting strings to lowercase and iterating through them character by character.
• **Boolean Arrays**: You used a boolean array to efficiently track the presence of each letter in the alphabet.
• **Conditional Logic**: You applied conditional checks to determine if all required elements (letters) were present.
• **Algorithm Design**: You developed an algorithmic approach to solving a problem by breaking it down into logical steps.

7. Time Complexity: O(n)

8. Space Complexity: O(n)