Accenture Sections	Information	Questions and Time
Cognitive Ability	• English Ability • Critical Thinking and Problem Solving • Abstract Reasoning	50 Ques in 50 mins
Technical Assessment	• Common Application and MS Office • Pseudo Code • Fundamental of Networking, Security and Cloud	40 Ques in 40 mins
Coding Round	• C • C++ • Dot Net • JAVA • Python	2 Ques in 45 mins

DEBUG WITH SHUBHAM

Accenture Technical Assessment Detailed Overview

07-Oct-2024 Coding Interview Questions


https://www.youtube.com/@DebugWithShubham


https://www.linkedin.com/in/debugwithshubham/


https://www.instagram.com/debugwithshubham/


https://topmate.io/debugwithshubham


https://t.me/debugwithshubham

2. Hands On Programming

Revisit Later

Question 1

How to Attempt?

You are developing a feature for a mobile game that tracks player performance. The game records the results of consecutive rounds as a binary array **A**, where 1 indicates a win and 0 indicates a loss. The game developers want to analyze how well players are doing by determining the longest streak of consecutive wins. Your task is to find and return an integer value representing the count of the maximum number of consecutive 1s (wins) in the array.

Input Specification:

input1 : A binary integer array A representing the game records.
input2 : An integer value representing the number of games played.

Output Specification:

Return an integer value representing the count of the maximum number of consecutive 1s (wins) in the array.

Example 1:

input1 : {1,1,0,1,1,1}
input2 : 6

Output : 3

Explanation:
Here, the number of games played is 6 and the game records are {1,1,0,1,1,1}. The longest streak of consecutive wins (1s) is 3. Therefore, **3** is returned as the output.

Example 2:

input1 : {1,0,1,1,0,1}
input2 : 6

Output : 2

Explanation:
Here, the number of games played is 6 and the game records are {1,0,1,1,0,1}. The longest streak of consecutive wins (1s) is 2. Therefore, **2** is returned as the output.

Question-1

You will get all the solution(C++,PYTHON,JAVA) on github Question name is Longest Streak of Consecutive Wins

main.py

```python
def max_consecutive_ones_sliding_window(A, n):
    max_count = 0
    current_count = 0
    for i in range(n):
        if A[i] == 1:
            current_count += 1
        else:
            current_count = 0
        max_count = max(max_count, current_count)
    return max_count

# Example
A = [1, 1, 0, 1, 1, 1,0,1,1,0,1,1,1,1]
n = 14
print(max_consecutive_ones_sliding_window(A, n))

```

Java

```java
public class MaxConsecutiveOnes {
    public static int maxConsecutiveOnes(int[] A, int n) {
        int maxCount = 0;
        int currentCount = 0;

        for (int i = 0; i < n; i++) {
            if (A[i] == 1) {
                currentCount++;
            } else {
                currentCount = 0;
            }
            maxCount = Math.max(maxCount, currentCount);
        }

        return maxCount;
    }

    public static void main(String[] args) {
        int[] A = {1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1};
        int n = 14;
        System.out.println(maxConsecutiveOnes(A, n));  // Output: 4
    }
}
```

main.cpp

```cpp
#include <iostream>
#include <vector>
#include <algorithm>
using namespace std;

int maxConsecutiveOnes(vector<int>& A, int n) {
    int maxCount = 0;
    int currentCount = 0;
    for (int i = 0; i < n; i++) {
        if (A[i] == 1) {
            currentCount++;
        } else {
            currentCount = 0;
        }
        maxCount = max(maxCount, currentCount);
    }
    return maxCount;
}
int main() {
    vector<int> A = {1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1};
    int n = A.size();
    cout << maxConsecutiveOnes(A, n) << endl;

    return 0;
}

```

2. Hands On Programming

Revisit Later

Question-2

How to Attempt?

Mike has written an English paragraph **P** on the white board, all in upper case. The paragraph contains vowels and consonants. His goal is to find that vowel which has occurred maximum times. He is writing the vowels as "E:19" which means E has occurred maximum times in paragraph P and the count is 19, separated with a ':'. Your task is to find and return a string representing that vowel and its respective occurrence. If no vowel is found, then return "No vowels found". If there is a conflict between two vowels with same occurrence, then return the vowel that arrives first in lexicographic order.

Input Specification:

input1 : A string value P representing the paragraph written on the white board.

Output Specification:

Return a string representing the vowel with the maximum occurrence along with its respective occurrence. If no vowel is found, then return "No vowels found". If there is a conflict between two vowels with same occurrence, then return the vowel that arrives first in lexicographic order.

xample 1:

input1 : THIS IS NOT AN INTEGER VALUE

2. Hands On Programming

Output Specification:

Return a string representing the vowel with the maximum occurrence along with its respective occurrence. If no vowel is found, then return "No vowels found". If there is a conflict between two vowels with same occurrence, then return the vowel that arrives first in lexicographic order.

Example 1:

input1 : THIS IS NOT AN INTEGER VALUE

Output : E:3

Explanation:

Here, paragraph P is "THIS IS NOT AN INTEGER VALUE". Below are the counts of ea vowel:

- A:2
- E:3
- I:3
- O:1
- U:1

We can see that the alphabet E and I occurred 3 times, which is maximum. Since E comes before I in lexicographic order, 'E' will be given the preference. Therefore, **E:** returned as the output.

- I:3
- O:1
- U:1

We can see that the alphabet E and I occurred 3 times, which is maxim comes before I in lexicographic order, 'E' will be given the preference. returned as the output.

Example 2:

input1 : RHYTHM

Output : No vowels found

Explanation:

Here, paragraph P is "RHYTHM". Below are the counts of each vowel:

- A:0
- E:0
- I:0
- O:0
- U:0

Since there is no vowel in the paragraph, **No vowels found** is returned

You will get all the solution(C++,PYTHON,JAVA) on github Question name is Maximum Vowel Frequency in a Paragraph

main.py

```python
def max_vowel_occurrence(P):
    vowels = "AEIOU"
    count = {v: 0 for v in vowels}
    for char in P:
        if char in count:
            count[char] += 1
    max_vowel = None
    max_count = 0
    for vowel in vowels:
        if count[vowel] > max_count:
            max_vowel = vowel
            max_count = count[vowel]
    if max_vowel:
        return f"{max_vowel}:{max_count}"
    else:
        return "No vowels found"
P = "THIS IS NOT AN INTEGER VALUE"
print(max_vowel_occurrence(P))
```

Main.java

```java
import java.util.HashMap;
import java.util.Map;

public class MaxVowelOccurrence {
    public static String maxVowelOccurrence(String P) {
        String vowels = "AEIOU";
        Map<Character, Integer> count = new HashMap<>();
        for (char v : vowels.toCharArray()) {
            count.put(v, 0);
        for (char ch : P.toCharArray()) {
            if (count.containsKey(ch)) {
                count.put(ch, count.get(ch) + 1);
            }
        }
        char maxVowel = 0;
        int maxCount = 0;
        for (char v : vowels.toCharArray()) {
            if (count.get(v) > maxCount) {
                maxVowel = v;
                maxCount = count.get(v);
            }
        }
        if (maxVowel != 0) {
            return maxVowel + ":" + maxCount;
        } else {
            return "No vowels found";
        }
    }
    public static void main(String[] args) {
        String P = "THIS IS NOT AN INTEGER VALUE";
        System.out.println(maxVowelOccurrence(P));  // Output: E:3
    }
}
```

main.cpp

```cpp
#include <unordered_map>
#include <string>
using namespace std;

string maxVowelOccurrence(const string& P) {
    string vowels = "AEIOU";
    unordered_map<char, int> count;
    for (char v : vowels) {
        count[v] = 0;
    for (char ch : P) {
        if (count.find(ch) != count.end()) {
            count[ch]++;
        }
    }
    char maxVowel = '\0';
    int maxCount = 0;

    for (char v : vowels) {
        if (count[v] > maxCount) {
            maxVowel = v;
            maxCount = count[v];
        }
    }
    if (maxVowel != '\0') {
        return string(1, maxVowel) + ":" + to_string(maxCount);
    } else {
        return "No vowels found";
    }
}
int main() {
    string P = "THIS IS NOT AN INTEGER VALUE";
    cout << maxVowelOccurrence(P) << endl;  // Output: E:3
    return 0;
}
```