Top Deloitte Coding Questions with Answers

Deloitte Coding Exam Questions: Overview

No. of Questions For The Coding Round	2 Live Coding Questions
Type of Test	Online
Time Duration	90 Minutes
Negative Marking	No Negative Marking

Top 10 Deloitte Coding Questions with Answers

Here, we've listed some commonly asked Deloitte coding questions with answers:

Q.1 How can a number be expressed as a sum of two prime numbers? Write a Program

Answer:

Sample Input: 34

C++ Code:

```
#include <iostream>
using namespace std;

bool isPrime(int n) {
    if (n <= 1) return false;
    for (int i = 2; i * i <= n; i++) {
        if (n % i == 0) return false;
    }
    return true;
}

void findPrimeSum(int num) {
    bool found = false;
```

```
    for (int i = 2; i <= num / 2; i++) {
        if (isPrime(i) && isPrime(num - i)) {
            cout << num << " can be expressed as the sum of " << i << " and " << num-i << "." <<
endl;
            found = true;
            break; // Uncomment this to find all pairs
        }
    }
    if (!found)
        cout << num << " cannot be expressed as the sum of two prime numbers." << endl;
}

int main() {
    int num = 34;
    findPrimeSum(num);
    return 0;
}
```

Java Code:

```
public class PrimeSum {

    public static boolean isPrime(int n) {
        if (n <= 1) return false;
        for (int i = 2; i <= Math.sqrt(n); i++) {
            if (n % i == 0) return false;
        }
        return true;
    }

    public static void findPrimeSum(int num) {
        boolean found = false;
        for (int i = 2; i <= num / 2; i++) {
            if (isPrime(i) && isPrime(num - i)) {
                System.out.println(num + " can be expressed as the sum of " + i + " and " + (num-i)
+ ".");
                found = true;
                break; // Uncomment this to find all pairs
            }
        }
        if (!found)
            System.out.println(num + " cannot be expressed as the sum of two prime numbers.");
    }

    public static void main(String[] args) {
        int num = 34;
        findPrimeSum(num);
    }
}
```

Python Code:

```
def is_prime(n):
    if n <= 1:
        return False
    for i in range(2, int(n**0.5) + 1):
        if n % i == 0:
            return False
    return True

def find_prime_sum(n):
    for i in range(2, n//2 + 1):
        if is_prime(i) and is_prime(n - i):
            return f"{n} can be expressed as the sum of {i} and {n - i}."
    return f"{n} cannot be expressed as the sum of two prime numbers."

# Sample input
n = 34
find_prime_sum(n)
```

Output:

'34 can be expressed as the sum of 3 and 31.'

Q.2 How to Check Whether a Number is a Perfect Number or not? Write a Program

Answer:

Sample Input: 28

C++ Code:

```
#include <iostream>
using namespace std;

bool isPerfect(int n) {
    int sum = 1;
    for (int i = 2; i * i <= n; i++) {
        if (n % i == 0) {
            sum += i;
            if (i != n / i) sum += n / i;
        }
    }
    return sum == n && n != 1;
```

```
}

int main() {
    int n = 28; // Sample input
    if (isPerfect(n))
        cout << n << " is a perfect number." << endl;
    else
        cout << n << " is not a perfect number." << endl;
    return 0;
}
```

Java Code:

```
public class PerfectNumber {

    static boolean isPerfect(int n) {
        int sum = 1;
        for (int i = 2; i * i <= n; i++) {
            if (n % i == 0) {
                sum += i;
                if (i != n / i) sum += n / i;
            }
        }
        return sum == n && n != 1;
    }

    public static void main(String[] args) {
        int n = 28; // Sample input
        if (isPerfect(n))
            System.out.println(n + " is a perfect number.");
        else
            System.out.println(n + " is not a perfect number.");
    }
}
```

Python Code:

```
def is_perfect(n):
    sum = 1
    for i in range(2, int(n**0.5) + 1):
        if n % i == 0:
            sum += i
            if i != n // i:
                sum += n // i
    return sum == n and n != 1

# Sample input
n = 28
if is_perfect(n):
    result = f"{n} is a perfect number."
```

```
else:
    result = f"{n} is not a perfect number."

result
```

Output:

28 is a perfect number because its divisors are 1, 2, 4, 7, and 14, and the sum of these divisors is 28.

Q.3 Problem Statement:

There is a list of decimal numbers, as well as an infection that has a specific amount of spikes. Each spike represents the amount of binary digits that a virus will eat from the right-hand part of a decimal number. Write a program that will determine the final status of each number following the virus that has consumed the specified amount of binary digits from each one of them.

Answer:

Sample Input:

6
10 20 30 40 50 60
3

C++ Code:

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

int main() {
    int N, n;
    cin >> N;
    vector<int> v(N);
    for (int i = 0; i < N; i++)
        cin >> v[i];
    cin >> n;
    for (auto i : v)
        cout << (i >> n) << " ";
    return 0;
}
```

Java Code:

```
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int N = sc.nextInt();
        int[] arr = new int[N];
        for (int i = 0; i < N; i++) {
            arr[i] = sc.nextInt();
        }
        int n = sc.nextInt();
        for (int i : arr) {
            System.out.print((i >> n) + " ");
        }
    }
}
```

Python Code:

```
N = int(input())
a = list(map(int, input().split()))
n = int(input())
result = " ".join(str(i >> n) for i in a)
print(result)
```

Output:

‘1 2 3 5 6 7’

Q.4 Can You Write a program that takes a string containing multiple words as input? Sort these words in ascending order alphabetically. Then, print all the words that start with a vowel, along with their positions in the sorted list.

Answer:

Sample Input: apple banana cherry orange mango

C++ Code:

```
#include <iostream>
```

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

bool isVowel(char c) {
    return c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u' ||
           c == 'A' || c == 'E' || c == 'I' || c == 'O' || c == 'U';
}

int main() {
    string input;
    getline(cin, input);

    vector<string> words;
    string word = "";
    for (char c : input) {
        if (c == ' ') {
            words.push_back(word);
            word = "";
        } else {
            word += c;
        }
    }
    words.push_back(word);

    sort(words.begin(), words.end());

    for (int i = 0; i < words.size(); i++) {
        if (isVowel(words[i][0])) {
            cout << i + 1 << ": " << words[i] << endl;
        }
    }

    return 0;
}
```

Java Code:

```
import java.util.Arrays;

public class Main {
    public static void main(String[] args) {
        String input = "apple banana cherry orange mango";

        String[] words = input.split(" ");
        Arrays.sort(words);

        for (int i = 0; i < words.length; i++) {
            if (isVowel(words[i].charAt(0))) {
                System.out.println((i + 1) + ": " + words[i]);
```

```
        }
      }
   }

   public static boolean isVowel(char c) {
      return c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u' ||
            c == 'A' || c == 'E' || c == 'I' || c == 'O' || c == 'U';
   }
}
```

Python Code:

```
def is_vowel(c):
   return c in 'aeiouAEIOU'

input_str = "apple banana cherry orange mango"

words = input_str.split()
words.sort()

for i, word in enumerate(words, 1):
   if is_vowel(word[0]):
      print(f"{i}: {word}")
```

Output:

1: apple
2: orange

Also Read: [20 IBM Coding Questions with Answers]

Q.5 Problem Statement:

After JEE Mains, students have been admitted to an engineering college, forming a class of 'n' students. The Head of the Department (HOD) is tasked with selecting a class monitor. However, the HOD meets the students one by one and has a peculiar way of choosing. Each time the HOD meets a student, if the student's rank is lower than that of the previous student met, the HOD cuts the previous student's name and writes the new student's name along with their rank in a register. Given the ranks the HOD receives each time they meet a student, predict how many names are cut from the list.

Constraints:

Number of visits <= 10^9
Ranks <= 10000

Input Format:

The first line contains the number of visits 'N'.
The second line contains 'N' space-separated ranks the HOD receives each time.
Output Format:

Output Format:

Single N space separated integers denoting the final situation with the array v.

Answer:

Sample Input:

7
5 8 6 3 9 4 2

C++ Code:

```
#include <iostream>
using namespace std;

int main() {
    int N;
    cin >> N;

    int prevRank, currRank, cuts = 0;
    cin >> prevRank;

    for (int i = 1; i < N; i++) {
        cin >> currRank;
        if (currRank < prevRank) {
            cuts++;
        }
        prevRank = currRank;
    }

    cout << cuts << endl;

    return 0;
}
```

Java Code:

```
import java.util.Scanner;
```

```
public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int N = sc.nextInt();

        int prevRank = sc.nextInt();
        int cuts = 0;

        for (int i = 1; i < N; i++) {
            int currRank = sc.nextInt();
            if (currRank < prevRank) {
                cuts++;
            }
            prevRank = currRank;
        }

        System.out.println(cuts);
    }
}
```

Python Code:

```
N = int(input())
ranks = list(map(int, input().split()))

cuts = 0
prev_rank = ranks[0]

for rank in ranks[1:]:
    if rank < prev_rank:
        cuts += 1
    prev_rank = rank

print(cuts)
```

Output: 3

Q.6 Problem Statement:

Rahul is known for copying in exams from his adjacent students, but he's smart about it. Instead of directly copying the words, he changes the positions of letters while keeping the letters constant. As the examiner, you need to determine if Rahul has copied a certain word from the adjacent student who is giving the same exam. You should provide Rahul with the appropriate markings based on your findings.

Note: Uppercase and lowercase letters are considered the same.

Input Format:

The first line contains the word of the adjacent student.
The second line contains Rahul's words.

Output Format:

0 if Rahul did not copy.
1 if Rahul copied.
Constraints:

1 <= Length of string <= 10^6

Answer:

Sample Input:

HELLO
EHLLO

C++ Code:

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

int main() {
    string adjacentWord, rahulWord;
    cin >> adjacentWord >> rahulWord;

    sort(adjacentWord.begin(), adjacentWord.end());
    sort(rahulWord.begin(), rahulWord.end());

    if (adjacentWord == rahulWord) {
        cout << 1 << endl;
    } else {
        cout << 0 << endl;
    }

    return 0;
}
```

Java Code:

```
import java.util.Arrays;
import java.util.Scanner;
```

```
public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String adjacentWord = sc.next();
        String rahulWord = sc.next();

        char[] adjacentArray = adjacentWord.toCharArray();
        char[] rahulArray = rahulWord.toCharArray();

        Arrays.sort(adjacentArray);
        Arrays.sort(rahulArray);

        if (Arrays.equals(adjacentArray, rahulArray)) {
            System.out.println(1);
        } else {
            System.out.println(0);
        }
    }
}
```

Python Code:

```
adjacent_word = input()
rahul_word = input()

if sorted(adjacent_word.lower()) == sorted(rahul_word.lower()):
    print(1)
else:
    print(0)
```

Output: 1

Q.7 Problem Statement:

Anirudh is attending an astronomy lecture, but he's struggling with understanding the material. His professor, who is very strict, asks students to write a program to print a trapezium pattern using stars (*) and dots (.) as shown below. Since Anirudh is not proficient in astronomy, can you help him?

Answer:

Sample Input:

N = 5

C++ Code:

```
#include <iostream>
using namespace std;

int main() {
    int N;
    cin >> N;

    for (int i = 0; i < N; i++) {
        for (int j = 0; j < N * 2; j++) {
            if (j < N - i || j >= N + i + 1) {
                cout << ".";
            } else {
                cout << "*";
            }
        }
        cout << endl;
    }

    return 0;
}
```

Java Code:

```
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int N = sc.nextInt();

        for (int i = 0; i < N; i++) {
            for (int j = 0; j < N * 2; j++) {
                if (j < N - i || j >= N + i + 1) {
                    System.out.print(".");
                } else {
                    System.out.print("*");
                }
            }
            System.out.println();
        }
    }
}
```

Python Code:

```
N = int(input())

for i in range(N):
```

```
    for j in range(N * 2):
        if j < N - i or j >= N + i + 1:
            print(".", end="")
        else:
            print("*", end="")
    print()
```

Output:

```
**..**
*....*
......
*....*
**..**
```

Q.8 Problem Statement:

You are given an array, and you need to choose a contiguous subarray of length 'k'. Then, find the minimum value within that subarray and return the maximum of those minimum values.

Answer:

Sample Input:

- Length of the subarray: 2
- Size of the array: 6
- Array elements: [3, 1, 4, 6, 2, 5]

Explanation:

The subarrays of size 2 are: [3, 1], [1, 4], [4, 6], [6, 2], and [2, 5].

The minimum values within these subarrays are: 1, 1, 4, 2, and 2 respectively.
The maximum of these minimum values is 4.

C++ Code:

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

int main() {
    int k, n;
```

```
    cin >> k >> n;
    vector<int> arr(n);
    for (int i = 0; i < n; ++i) {
        cin >> arr[i];
    }

    int max_min = INT_MIN;
    for (int i = 0; i <= n - k; ++i) {
        int min_val = *min_element(arr.begin() + i, arr.begin() + i + k);
        max_min = max(max_min, min_val);
    }

    cout << max_min << endl;

    return 0;
}
```

Java Code:

```
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int k = sc.nextInt();
        int n = sc.nextInt();
        int[] arr = new int[n];
        for (int i = 0; i < n; ++i) {
            arr[i] = sc.nextInt();
        }

        int maxMin = Integer.MIN_VALUE;
        for (int i = 0; i <= n - k; ++i) {
            int minVal = Integer.MAX_VALUE;
            for (int j = i; j < i + k; ++j) {
                minVal = Math.min(minVal, arr[j]);
            }
            maxMin = Math.max(maxMin, minVal);
        }

        System.out.println(maxMin);
    }
}
```

Python Code:

```
k, n = map(int, input().split())
arr = list(map(int, input().split()))

max_min = float('-inf')
```

```
for i in range(n - k + 1):
    min_val = min(arr[i:i + k])
    max_min = max(max_min, min_val)

print(max_min)
```

Output: 4

Q.9 Problem Statement:

A password manager wants to create new passwords using two strings given by the user, then combines them to create a harder-to-guess combination. Given two strings, the task is to interleave the characters of the strings to create a new string. Begin with an empty string and alternately append a character from string 'a' and from string 'b'. If one of the strings is exhausted before the other, append the remaining letters from the other string all at once. The resulting string is the new password.

Example:
If a = 'hackerrank' and b = 'mountain', the resulting password is 'hmaocuknetrariannk'.

Function Description:
Complete the function newPassword which takes two strings 'a' and 'b' as input and returns the new password as a string.

Parameters:

Str a: String 'a'
Str b: String 'b'

Returns:

Str: New password using two strings

Answer:

Sample Input:

a = "open"
b = "source"

C++ Code:

```
#include <iostream>
```

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

string newPassword(string a, string b) {
    string result = "";
    int i = 0, j = 0;

    while (i < a.length() && j < b.length()) {
        result += a[i++];
        result += b[j++];
    }

    while (i < a.length()) {
        result += a[i++];
    }

    while (j < b.length()) {
        result += b[j++];
    }

    return result;
}

int main() {
    string a, b;
    cin >> a >> b;
    cout << newPassword(a, b) << endl;
    return 0;
}
```

Java Code:

```
import java.util.Scanner;

public class Main {
    public static String newPassword(String a, String b) {
        StringBuilder result = new StringBuilder();
        int i = 0, j = 0;

        while (i < a.length() && j < b.length()) {
            result.append(a.charAt(i++));
            result.append(b.charAt(j++));
        }

        while (i < a.length()) {
            result.append(a.charAt(i++));
        }

        while (j < b.length()) {
            result.append(b.charAt(j++));
```

```
        }

        return result.toString();
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String a = sc.next();
        String b = sc.next();
        System.out.println(newPassword(a, b));
    }
}
```

Python Code:

```
def newPassword(a, b):
    result = ""
    i, j = 0, 0

    while i < len(a) and j < len(b):
        result += a[i] + b[j]
        i += 1
        j += 1

    while i < len(a):
        result += a[i]
        i += 1

    while j < len(b):
        result += b[j]
        j += 1

    return result

a = input()
b = input()
print(newPassword(a, b))
```

Output: ‘ospuoerce’

Lean More: [Top Infosys Coding Questions with Answers For Specialist Programmer]

Q.10 Problem Statement:

Anish, known for his laziness, is tasked with writing the name of the winner in a game where two people participate. However, instead of writing the winner's name directly, he simply writes the longest common subsequence (LCS) of the two names. This allows him to minimize the number of changes or avoid using the backspace key while editing the name. Given two names, your

task is to predict what Anish will write on his computer before the start of the game. If there are multiple longest subsequences possible, choose the one with the smallest lexicographic value.

Input Format:

Two lines containing two strings representing the names (all letters are in capital case).

Output Format:

A single line containing the lexicographically smallest possible longest common subsequence.

Answer:

Sample Input:

GEEK
EKEG

C++ Code:

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

int main() {
    string s1, s2;
    cin >> s1 >> s2;

    vector<vector<int>> dp(s1.size() + 1, vector<int>(s2.size() + 1, 0));

    for (int i = 1; i <= s1.size(); ++i) {
        for (int j = 1; j <= s2.size(); ++j) {
            if (s1[i - 1] == s2[j - 1]) {
                dp[i][j] = dp[i - 1][j - 1] + 1;
            } else {
                dp[i][j] = max(dp[i - 1][j], dp[i][j - 1]);
            }
        }
    }

    string lcs = "";
    int i = s1.size(), j = s2.size();
    while (i > 0 && j > 0) {
        if (s1[i - 1] == s2[j - 1]) {
            lcs = s1[i - 1] + lcs;
            i--;
```

```
            j--;
        } else if (dp[i - 1][j] > dp[i][j - 1]) {
            i--;
        } else {
            j--;
        }
    }

    cout << lcs << endl;

    return 0;
}
```

Java Code:

```
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String s1 = sc.next();
        String s2 = sc.next();

        int[][] dp = new int[s1.length() + 1][s2.length() + 1];

        for (int i = 1; i <= s1.length(); ++i) {
            for (int j = 1; j <= s2.length(); ++j) {
                if (s1.charAt(i - 1) == s2.charAt(j - 1)) {
                    dp[i][j] = dp[i - 1][j - 1] + 1;
                } else {
                    dp[i][j] = Math.max(dp[i - 1][j], dp[i][j - 1]);
                }
            }
        }

        StringBuilder lcs = new StringBuilder();
        int i = s1.length(), j = s2.length();
        while (i > 0 && j > 0) {
            if (s1.charAt(i - 1) == s2.charAt(j - 1)) {
                lcs.insert(0, s1.charAt(i - 1));
                i--;
                j--;
            } else if (dp[i - 1][j] > dp[i][j - 1]) {
                i--;
            } else {
                j--;
            }
        }

        System.out.println(lcs.toString());
```

```
    }
}
```

Python Code:

```
def lcs(s1, s2):
    m, n = len(s1), len(s2)
    dp = [[0] * (n + 1) for _ in range(m + 1)]

    for i in range(1, m + 1):
        for j in range(1, n + 1):
            if s1[i - 1] == s2[j - 1]:
                dp[i][j] = dp[i - 1][j - 1] + 1
            else:
                dp[i][j] = max(dp[i - 1][j], dp[i][j - 1])

    lcs = ""
    i, j = m, n
    while i > 0 and j > 0:
        if s1[i - 1] == s2[j - 1]:
            lcs = s1[i - 1] + lcs
            i -= 1
            j -= 1
        elif dp[i - 1][j] > dp[i][j - 1]:
            i -= 1
        else:
            j -= 1

    return lcs

s1 = input()
s2 = input()
print(lcs(s1, s2))
```

Output: ‘EG’

Follow Us on Facebook for the Latest Updates.