Top Infosys Coding Questions and Answers (2024)

Q.1 - How do you swap two arrays in C++?

Answer:

Sample Input

A[ ] = {21, 22, 23, 24}
B[ ] = {25, 26, 27, 28}

Sample Output

A[ ] = {25, 26, 27, 28}
B[ ] = {21, 22, 23, 24}

C++ Code:

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

void swapArrays(int* a, int* b, int size) {
    for (int i = 0; i < size; i++) {
        int temp = a[i];
        a[i] = b[i];
        b[i] = temp;
    }
}

int main() {
    int a[] = {21, 22, 23, 24};
    int b[] = {25, 26, 27, 28};
    int n = sizeof(a)/sizeof(a[0]);

    swapArrays(a, b, n);

    cout << "a[] = ";
    for (int i = 0; i < n; i++)
        cout << a[i] << ", ";
    cout << "\nb[] = ";
    for (int i = 0; i < n; i++)
```

```
        cout << b[i] << ", ";
    return 0;
}
```

Output

A[ ] = {25, 26, 27, 28}
B[ ] = {21, 22, 23, 24}

Q.2 - Write a program to arrange the given numbers to form the biggest number

Answer:

Sample Input

{5, 67, 2, 88, 9, 76, 52, 4}

C++

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

// Custom comparator to decide which concatenation forms a larger number
struct Comparator {
    bool operator()(int a, int b) {
        string ab = to_string(a) + to_string(b);
        string ba = to_string(b) + to_string(a);
        return ab > ba;
    }
};

int main() {
    vector<int> nums = {5, 67, 2, 88, 9, 76, 52, 4};
    sort(nums.begin(), nums.end(), Comparator());

    for (int num : nums) {
        cout << num;
    }
    cout << endl;
    return 0;
}
```

Java

```
import java.util.*;

public class LargestNumberSimple {

    public static void main(String[] args) {
        // Input array of integers
        String[] numbers = {"5", "67", "2", "88", "9", "76", "52", "4"};

        // Sort the array using a custom Comparator
        Arrays.sort(numbers, new Comparator<String>() {
            public int compare(String a, String b) {
                // Compare two concatenations to decide which should come first
                return (b + a).compareTo(a + b);
            }
        });

        // Check for a case where the largest number is 0
        if (numbers[0].equals("0")) {
            System.out.println("0");
            return;
        }

        // Concatenate the sorted numbers
        String largestNumber = "";
        for (String num : numbers) {
            largestNumber += num;
        }

        // Print the largest number
        System.out.println(largestNumber);
    }
}
```

Python

```
def largestNumber(arr):
    # Convert integers to strings to enable custom sorting
    arr = sorted(map(str, arr), key=lambda x: x*10, reverse=True)
    # Join and return the sorted array, handling leading zeros
    result = ''.join(arr)
    return result if result[0] != '0' else '0'

# New input array
arr = [5, 67, 2, 88, 9, 76, 52, 4]
print(largestNumber(arr))
```

Output

988766752452

Q.3 - Find the smallest and largest number in an Array

Answer:

Sample Input

[3, 1, 56, 34, 12, 9, 98, 23, 4]

C++ Code:

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

int main() {
    vector<int> arr = {3, 1, 56, 34, 12, 9, 98, 23, 4};
    int minVal = arr[0];
    int maxVal = arr[0];

    for(int i = 1; i < arr.size(); i++) {
        if(arr[i] > maxVal) maxVal = arr[i];
        if(arr[i] < minVal) minVal = arr[i];
    }

    cout << "Smallest Number: " << minVal << endl;
    cout << "Largest Number: " << maxVal << endl;

    return 0;
}
```

Java Code:

```
public class Main {
    public static void main(String[] args) {
        int[] arr = {3, 1, 56, 34, 12, 9, 98, 23, 4};
        int minVal = arr[0];
        int maxVal = arr[0];

        for(int i = 1; i < arr.length; i++) {
            if(arr[i] > maxVal) maxVal = arr[i];
            if(arr[i] < minVal) minVal = arr[i];
        }

        System.out.println("Smallest Number: " + minVal);
        System.out.println("Largest Number: " + maxVal);
    }
}
```

Python Code:

```
arr = [3, 1, 56, 34, 12, 9, 98, 23, 4]
minVal = min(arr)
maxVal = max(arr)

print(f"Smallest Number: {minVal}")
print(f"Largest Number: {maxVal}")
```

Output

Smallest Number: 1
Largest Number: 98

Q.4 - Find the next permutation of the given string in C++

Answer:

Sample Input

s=”dcd”

C++ Code:

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

void swap(char* a, char* b) {
    if (*a == *b)
        return;
    *a ^= *b;
    *b ^= *a;
    *a ^= *b;
}

void reverseString(string& s, int start, int end) {
    while (start < end) {
        swap(s[start], s[end]);
        start++;
        end--;
    }
}

bool nextPermutation(string& s) {
    int n = s.length();
```

```
    int i = n - 2;
    while (i >= 0 && s[i] >= s[i + 1])
        i--;
    if (i < 0)
        return false;
    int j = n - 1;
    while (s[j] <= s[i])
        j--;
    swap(s[i], s[j]);
    reverseString(s, i + 1, n - 1);
    return true;
}

int main() {
    string s = "dcd"; // Sample Input
    bool val = nextPermutation(s);
    if (!val)
        cout << "No next permutation possible" << endl;
    else
        cout << "Next permutation: " << s << endl;
    return 0;
}
```

Output

Next permutation: ddc

Q.5 How do you rotate a matrix by 90 degrees? Write a program.

Answer:

Sample Input:

1 2 3
4 5 6
7 8 9

C++ Code:

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

void rotateMatrix(vector<vector<int>>& matrix) {
    int n = matrix.size();
    // Transpose the matrix
```

```
    for (int i = 0; i < n; ++i) {
        for (int j = i + 1; j < n; ++j) {
            swap(matrix[i][j], matrix[j][i]);
        }
    }
    // Reverse each row
    for (int i = 0; i < n; ++i) {
        int left = 0, right = n - 1;
        while (left < right) {
            swap(matrix[i][left], matrix[i][right]);
            left++;
            right--;
        }
    }
}

int main() {
    // Input matrix
    vector<vector<int>> matrix = {{1, 2, 3},
                                  {4, 5, 6},
                                  {7, 8, 9}};

    // Print original matrix
    cout << "Original Matrix:" << endl;
    for (const auto& row : matrix) {
        for (int val : row) {
            cout << val << " ";
        }
        cout << endl;
    }

    // Rotate matrix
    rotateMatrix(matrix);

    // Print rotated matrix
    cout << "Rotated Matrix:" << endl;
    for (const auto& row : matrix) {
        for (int val : row) {
            cout << val << " ";
        }
        cout << endl;
    }

    return 0;
}
```

Java Code:

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

```
public class RotateMatrix {
    public static void rotateMatrix(int[][] matrix) {
        int n = matrix.length;
        // Transpose the matrix
        for (int i = 0; i < n; ++i) {
            for (int j = i + 1; j < n; ++j) {
                int temp = matrix[i][j];
                matrix[i][j] = matrix[j][i];
                matrix[j][i] = temp;
            }
        }
        // Reverse each row
        for (int i = 0; i < n; ++i) {
            int left = 0, right = n - 1;
            while (left < right) {
                int temp = matrix[i][left];
                matrix[i][left] = matrix[i][right];
                matrix[i][right] = temp;
                left++;
                right--;
            }
        }
    }

    public static void main(String[] args) {
        // Input matrix
        int[][] matrix = {{1, 2, 3},
                          {4, 5, 6},
                          {7, 8, 9}};

        // Print original matrix
        System.out.println("Original Matrix:");
        for (int[] row : matrix) {
            System.out.println(Arrays.toString(row));
        }

        // Rotate matrix
        rotateMatrix(matrix);

        // Print rotated matrix
        System.out.println("Rotated Matrix:");
        for (int[] row : matrix) {
            System.out.println(Arrays.toString(row));
        }
    }
}
```

Python Code:

```
def rotate_matrix(matrix):
```

```
    n = len(matrix)
    # Transpose the matrix
    for i in range(n):
        for j in range(i + 1, n):
            matrix[i][j], matrix[j][i] = matrix[j][i], matrix[i][j]
    # Reverse each row
    for i in range(n):
        matrix[i] = matrix[i][::-1]

# Input matrix
matrix = [[1, 2, 3],
          [4, 5, 6],
          [7, 8, 9]]

# Print original matrix
print("Original Matrix:")
for row in matrix:
    print(row)

# Rotate matrix
rotate_matrix(matrix)

# Print rotated matrix
print("Rotated Matrix:")
for row in matrix:
    print(row)
```

Output:

Rotated Matrix:

7 4 1
8 5 2
9 6 3

Also Read: [20 IBM Coding Questions with Answers (Assessment by Expert)]

Q.6 How do you find the missing characters to make a string pangram? Write a program

Answer:

Sample Input: "A quick movement of the enemy will jeopardize six gunboats"

C++ Code:

```
#include <iostream>
#include <string>
#include <unordered_set>
#include <algorithm>

std::string findMissingCharacters(const std::string& s) {
    std::unordered_set<char> alphabets;
    for (char c = 'a'; c <= 'z'; ++c) {
        alphabets.insert(c);
    }

    for (char c : s) {
        alphabets.erase(tolower(c));
    }

    std::string missing;
    for (char c : alphabets) {
        missing.push_back(c);
    }

    std::sort(missing.begin(), missing.end());
    return missing;
}

int main() {
    std::string str = "A quick movement of the enemy will jeopardize six gunboats";
    std::cout << "Missing characters: " << findMissingCharacters(str) << std::endl;
    return 0;
}
```

Java Code:

```
import java.util.HashSet;
import java.util.Set;

public class PangramChecker {
    public static String findMissingCharacters(String s) {
        Set<Character> alphabets = new HashSet<>();
        for (char c = 'a'; c <= 'z'; c++) {
            alphabets.add(c);
        }

        for (char c : s.toLowerCase().toCharArray()) {
            alphabets.remove(c);
        }

        StringBuilder missing = new StringBuilder();
        for (char c : alphabets) {
```

```
            missing.append(c);
        }

        return missing.toString();
    }

    public static void main(String[] args) {
        String str = "A quick movement of the enemy will jeopardize six gunboats";
        System.out.println("Missing characters: " + findMissingCharacters(str));
    }
}
```

Python Code:

```
def find_missing_characters_for_pangram(s):
    alphabets = set('abcdefghijklmnopqrstuvwxyz')
    for char in s.lower():
        alphabets.discard(char)
    return ''.join(sorted(alphabets))

# Example usage
string = "A quick movement of the enemy will jeopardize six gunboats"
missing_characters = find_missing_characters_for_pangram(string)
print("Missing characters:", missing_characters)
```

Output

"Missing characters: flr"

Q.7 How do you find the number of unique characters in a given string? Write a program

Answer:

Sample Input: "Hello, World!"

C++ Code:

```
#include <iostream>
#include <unordered_set>

int countUniqueCharacters(const std::string& str) {
    std::unordered_set<char> uniqueChars;
    for (char c : str) {
        uniqueChars.insert(c);
    }
```

```
    return uniqueChars.size();
}

int main() {
    std::string input = "Hello, World!";
    std::cout << "Number of unique characters: " << countUniqueCharacters(input) <<
std::endl;
    return 0;
}
```

Java Code:

```
import java.util.HashSet;
import java.util.Set;

public class UniqueCharacterCounter {
    public static int countUniqueCharacters(String str) {
        Set<Character> uniqueChars = new HashSet<>();
        for (char c : str.toCharArray()) {
            uniqueChars.add(c);
        }
        return uniqueChars.size();
    }

    public static void main(String[] args) {
        String input = "Hello, World!";
        System.out.println("Number of unique characters: " + countUniqueCharacters(input));
    }
}
```

Python Code:

```
def count_unique_characters(s):
    return len(set(s))

# Example usage
input_string = "Hello, World!"
unique_character_count = count_unique_characters(input_string)
print("Number of unique characters:", unique_character_count)
```

Output

Number of unique characters is 10.

Q.8 Write a Program for the Subtraction of Two Matrices

Answer:

Sample Input:

a)
1 2
3 4

b)
4 3
2 1

C++ Code:

```
#include <iostream>
#include <vector>

std::vector<std::vector<int>> subtractMatrices(const std::vector<std::vector<int>>& A, const
std::vector<std::vector<int>>& B) {
    std::vector<std::vector<int>> result(A.size(), std::vector<int>(A[0].size()));
    for (size_t i = 0; i < A.size(); ++i) {
        for (size_t j = 0; j < A[0].size(); ++j) {
            result[i][j] = A[i][j] - B[i][j];
        }
    }
    return result;
}

int main() {
    std::vector<std::vector<int>> A = {{1, 2}, {3, 4}};
    std::vector<std::vector<int>> B = {{4, 3}, {2, 1}};
    std::vector<std::vector<int>> result = subtractMatrices(A, B);

    for (const auto& row : result) {
        for (int val : row) {
            std::cout << val << " ";
        }
        std::cout << std::endl;
    }
    return 0;
}
```

Java Code:

```
public class MatrixSubtraction {
    public static int[][] subtractMatrices(int[][] A, int[][] B) {
        int[][] result = new int[A.length][A[0].length];
        for (int i = 0; i < A.length; i++) {
            for (int j = 0; j < A[0].length; j++) {
                result[i][j] = A[i][j] - B[i][j];
            }
```

```
        }
        return result;
    }

    public static void main(String[] args) {
        int[][] A = {{1, 2}, {3, 4}};
        int[][] B = {{4, 3}, {2, 1}};
        int[][] result = subtractMatrices(A, B);

        for (int[] row : result) {
            for (int val : row) {
                System.out.print(val + " ");
            }
            System.out.println();
        }
    }
}
```

Python Code:

```
def subtract_matrices(A, B):
    return [[A[i][j] - B[i][j] for j in range(len(A[0]))] for i in range(len(A))]

# Example usage
A = [[1, 2], [3, 4]]
B = [[4, 3], [2, 1]]
result = subtract_matrices(A, B)

for row in result:
    print(" ".join(map(str, row)))
```

Output

```
-3 -1
 1  3
```

Q.9 How do you multiply two matrices and show results through another matrix? Write a program

Answer:

A:
1 2
3 4

B:
5 6

C++ Code:

```
#include <iostream>
#include <vector>

std::vector<std::vector<int>> multiplyMatrices(const std::vector<std::vector<int>>& A, const
std::vector<std::vector<int>>& B) {
    size_t rows = A.size();
    size_t cols = B[0].size();
    size_t common = B.size();
    std::vector<std::vector<int>> result(rows, std::vector<int>(cols, 0));

    for (size_t i = 0; i < rows; ++i) {
        for (size_t j = 0; j < cols; ++j) {
            for (size_t k = 0; k < common; ++k) {
                result[i][j] += A[i][k] * B[k][j];
            }
        }
    }
    return result;
}

int main() {
    std::vector<std::vector<int>> A = {{1, 2}, {3, 4}};
    std::vector<std::vector<int>> B = {{5, 6}, {7, 8}};
    std::vector<std::vector<int>> result = multiplyMatrices(A, B);

    for (const auto& row : result) {
        for (int val : row) {
            std::cout << val << " ";
        }
        std::cout << std::endl;
    }
    return 0;
}
```

Java Code:

```
public class MatrixMultiplication {
    public static int[][] multiplyMatrices(int[][] A, int[][] B) {
        int rows = A.length;
        int cols = B[0].length;
        int common = B.length;
        int[][] result = new int[rows][cols];

        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < cols; j++) {
                for (int k = 0; k < common; k++) {
```

```
                result[i][j] += A[i][k] * B[k][j];
            }
        }
    }
    return result;
}

public static void main(String[] args) {
    int[][] A = {{1, 2}, {3, 4}};
    int[][] B = {{5, 6}, {7, 8}};
    int[][] result = multiplyMatrices(A, B);

    for (int[] row : result) {
        for (int val : row) {
            System.out.print(val + " ");
        }
        System.out.println();
    }
}
}
```

Python Code:

```
def multiply_matrices(A, B):
    rows_A, cols_A = len(A), len(A[0])
    rows_B, cols_B = len(B), len(B[0])
    result = [[0 for _ in range(cols_B)] for _ in range(rows_A)]

    for i in range(rows_A):
        for j in range(cols_B):
            for k in range(cols_A):
                result[i][j] += A[i][k] * B[k][j]
    return result

# Example usage
A = [[1, 2], [3, 4]]
B = [[5, 6], [7, 8]]
result = multiply_matrices(A, B)

for row in result:
    print(" ".join(map(str, row)))
```

Output

The output of multiplying the given matrices A and B is:

```
19 22
43 50
```

Q.10 How do you convert decimal numbers to binary numbers? Write a Program

Answer:

Sample Input: 29

C++ Code

```
#include <iostream>
#include <string>
#include <algorithm>

std::string decimalToBinary(int n) {
    std::string binary = "";
    while (n > 0) {
        binary += std::to_string(n % 2);
        n /= 2;
    }
    std::reverse(binary.begin(), binary.end());
    return binary;
}

int main() {
    int decimal = 29;
    std::cout << "Binary of " << decimal << " is " << decimalToBinary(decimal) << std::endl;
    return 0;
}
```

Java Code:

```
public class DecimalToBinary {
    public static String decimalToBinary(int n) {
        StringBuilder binary = new StringBuilder();
        while (n > 0) {
            binary.insert(0, n % 2);
            n /= 2;
        }
        return binary.toString();
    }

    public static void main(String[] args) {
        int decimal = 29;
        System.out.println("Binary of " + decimal + " is " + decimalToBinary(decimal));
    }
}
```

Python Code:

```
def decimal_to_binary(n):
    binary = ""
    while n > 0:
        binary = str(n % 2) + binary
        n //= 2
    return binary

# Example usage
decimal = 29
binary = decimal_to_binary(decimal)
print(f"Binary of {decimal} is {binary}")
```

Output

The binary representation of the decimal number 29 is '11101'.

Follow Us on Facebook for Latest Updates.