

Miscellaneous Interview Questions

- 1) Create N threads and use them to print first M numbers in the following manner:
(Morgan Stanley)

Let's say $N = 3$ and $M = 7$

Thread 1 - Print 1

Thread 2 - Print 2

Thread 3 - Print 3

Thread 1 - Print 4

Thread 2 - Print 5

Thread 3 - Print 6

Thread 1 - Print 7

<https://wilddiary.com/printing-numbers-in-sequence-from-alternating-threads/>

- 2) Print the diameter of a Binary Tree (Morgan Stanley)

<https://www.geeksforgeeks.org/diameter-of-a-binary-tree/>

<https://www.geeksforgeeks.org/print-longest-leaf-leaf-path-binary-tree/>

- 3) Design a Card Game

- Black Jack
- Poker

(Service Now)

- 4) Design an Elevator System

- 5) Design an Airline System

- 6) Design a Restaurant

- 7) Design a Bakery

- 8) Design a Parking lot

ServiceNow Interview:

Written

1. Given a number, return excel column string for the number. (26 to Z , 27 AA)

1st Round

1. JVM pros and cons , why java has both compiler and interpreter.
2. SOLID programming principles
3. Design an examination portal (HLD). I want to get top 'n' students. How?
<https://www.coursera.org/learn/object-oriented-design>

2nd Round

1. Producer/Consumer, shared resources.
2. Print all possible type of parentheses for a given n. For n = 2 answer is (()) , ()()
3. DP minimum number of jumps required for going from 0 to n.
<https://www.geeksforgeeks.org/minimum-number-jumps-reach-endset-2on-solution/>

3rd Round (Managerial)

1. Sum of any number of numbers in array is equal to n. (subset sum problem)
<https://www.geeksforgeeks.org/subset-sum-problem-dp-25/>
2. Design WhatsApp Messenger
3. 25 horses puzzle

4th Round (HR)

5th Round

(ServiceNow US)

Question 1 :

4	8	100	-1000
70	-10	2000	70
-5	-21	-6	8
10000	-20	15	21

You have a set of integers which can be positive or negative (for example the one represented by this matrix).

Starting position is the top left (4 in this example) and end position is the bottom right (21 in this example).

You need to make your way from start to end while adding the numbers as you go.

From your current position, you can go to any of the 8 surrounding positions (up, down, left, right, diagonal) provided you have not counted it before.

The goal is to find the largest possible sum from start to end.

1. Design an generic algorithm to solve the problem. (Not just this specific example)
2. Code the solution to the problem.

Question 2:

Design blackjack game.

Goldman Sachs Interview:

Round 1 (HackerRank):

1. Find the number of subarrays with sum exactly equal to a given number.
<https://www.geeksforgeeks.org/number-subarrays-sum-exactly-equal-k/>
2. Given array of numbers (profits/losses on that day). Find the smallest non-negative number that needs to be added to first number such that sum all numbers from 0..i (i = 0 to n-1) become positive. The connection between numbers from 0..i and i+1 number is, result obtained after summing up all numbers from 0..i is added to i+1 number.

Eg-1:

Array: 4, 3, -2, 7

Ans: You can add 0 to 1st number such that you are profitable on every table like,
4, 7, 5, 12

Eg-2:

Array: 5, -9, 8, -15, 2

Ans: To make profitable on every day, you need to add 11 to 1st number such that numbers now become,
16, 7, 15, 0, 2

Eg-2:

Array: 5, -9, 100, -15, 2

Ans: To make profitable on every day, you need to add 4 to 1st number such that numbers now become,
9, 0, 100, 85, 87

Round 2 (CoderPad):

1. Given array of non-negative integers and a targetSum. Find that subarray with least length from this array such that sum of all numbers is at least targetSum.
2. Rain water trapped problem.
<https://www.interviewbit.com/problems/rain-water-trapped/>
3. How does HashTable work in Java (implementation details, buckets, collisions).
4. Difference between ArrayList vs LinkedList.

Round 3 (Telephonic Screening):

1. Difference between Abstract Class and Interface. When do you use which one?
2. Given an Employee Class. Make it immutable.
<https://www.journaldev.com/129/how-to-create-immutable-class-in-java>
3. Explain the Exception Hierarchy in Java.
<https://www.programcreek.com/2009/02/diagram-for-hierarchy-of-exception-classes/>
<https://www.programcreek.com/2013/10/top-10-questions-about-java-exceptions/>
4. Can we write our own Exceptions? Why do you want to write a Custom Exception? Have you ever written a Custom Exception?
<https://www.mkyong.com/java/java-custom-exception-examples/>
<https://stackify.com/java-custom-exceptions/>
5. Difference between REST and SOAP. Which one do you prefer? Why?
<https://www.guru99.com/comparison-between-web-services.html>
<https://stackify.com/soap-vs-rest/>
<https://stackoverflow.com/questions/19884295/soap-vs-rest-differences>
6. What are the tools you used for performance analysis at your workplace?
(Jdev CPU Profiler + JRockit)

COVIAM Technologies Interview:

Round 1 (Skype Screening):

About me and project related questions

Round 2 (codeshare.io):

1. Given a two dimensional array of size mxn
2. Array will contain only 1's and 0's
3. In a row if one zero occurs, the rest of elements in that row are zeros
4. Find the row which has maximum number of 1's with minimum time complexity

Example:

```
1 1 0 0
1 0 0 0
1 1 1 0 <- This row is the answer in this question
1 1 1 0
1 1 1 0
0 0 0 0
1 1 1 0
1 0 0 0
```

```
public int rowNumber(int[][] arr) {
    if((arr == null) || (arr.length == 0)) {
        return -1;
    }
    int rows = arr.length, cols = arr[0].length;
    int ans = 0, j = 0;
    for(int i = rows-1; i >= 0; i--) {
        if(arr[i][j] == 0) {
            Continue;
        } else {
            while(j < cols) {
                if(arr[i][j] == 1) {
                    ans = i;
                } else {
```

```

        j--;
        break;
    }
    j++;
}
}
}
return ans;
}

```

2. Reverse the second half of single Linked List

Example:

Input: 1->2->3->4->5->6

Output: 1->2->3->6->5->4

Input: 1->2->3->4->5

Output: 1->2->3->5->4

```

public void reverse(ListNode l) {
    if((l == null) || (l.next == null)) {
        return;
    }
    ListNode prev = null, curr = l;
    while(curr != null) {
        ListNode next = curr.next;
        curr.next = prev;
        prev = curr;
        curr = next;
    }
    return prev;
}

public void reverseSecondHalf(ListNode l) {
    if((l == null) || (l.next == null) || (l.next.next == null)) {
        return;
    }
    ListNode slow = l, fast = l;
    while((fast != null) && (fast.next != null)) {
        slow = slow.next;
        fast = fast.next.next;
    }
    if((fast != null) && (fast.next == null)) {
        slow = slow.next;
    }
}

```

```
    }  
    reverse(slow);  
}
```

Graph Algorithms:

- 1) DFS
- 2) BFS
- 3) Topological Sorting
- 4) Minimum Spanning Tree
- 5) Shortest Path