Round 1: Aptitude & Technical (Online Assessment)

Around 25 questions covering Quants, Reasoning, and Predict the output (C) were asked in this round. Predict the output of code questions were not of type MCQs. Most of the questions were based on nested loops, typecasting, and recursion. The duration of this round was 60 mins.

Round 2: Programming (F2F)

It was a live coding round, the interviewer gave me 4 coding questions. The duration of this round was around 100 mins. First, the interviewer gave me a Pattern-based program and asked me to share the screen and code. Upon completing it, they shared a link consisting of 3 programming questions and asked me to explain the approach and the complexities before writing the code for each of the questions.

1. Given a String with numbers and operators. Perform the operations on the numbers in their respective order.
Input : "12345*+-+"

Output: 6

Explanation: [1*2+3-4+5 = 6]

I explained my approach and gave a working solution in Python.

2. Program to check whether the given matrix is an upper triangular or lower triangular.
The constraint was each element should be visited only once.

Input:
4 6 1 4
0 3 5 9
0 0 6 2
0 0 0 8

Output: Upper Triangular Matrix

Input:
1 0
1 1
Output: Lower Triangular Matrix
3. Find the longest increasing sub-sequence in Array.
Input : {1,5,3,7}

Output: {1,5,7} or {1,3,7}

Input : {10,22,9,33,21,50,41}

Output: {10,22,33,50}

I solved all the problems correctly. The interviewer asked me to code the 2nd and 3rd questions in java. Upon completing this round, I was called for the next programming round which was scheduled after 2 hours on the same day.

Round 3: Advanced Programming (F2F) ~ 2 hrs.

This round was the most challenging and exciting of all, as it involved both Object design and data structure knowledge. Given family and extended family information, I was asked to construct a family tree from that information and need to find the list of eligible people one can marry. The conditions were as follows:

- For a male, a match will be his father's sister's daughter or his mother's brothers' daughter.
- For a female, a match will be her father's sister's son or her mother's brother's son.
- Names are unique.

Input format:

<person's name>, <gender>, <father's name>, <mother's name>

Input:
John, Male, Brad, Lisa
Emma, Female, Brad, Lisa
Alex, Male, John, Jenny
Emily, Female, Steve, Emma
Rachel, Female, Steve, Emma

Person name: Alex
Output: Emily, Rachel
I presented the UML diagram for the question and explained my approach. I was able to give a working solution in Java and the interviewer was satisfied with my solution.

Round 4: Technical Interview (1 hr)

It was a technical interview and discussion round, they tested both theoretical and practical knowledge. The questions were from Projects, Data structures, and some general questions about my goals. The interviewer also asked about the 2nd and 3rd round programming questions and asked me to optimize the previous round question. He gave me some hints, and I could give the optimized solution for the problem.

Round 5: HR Interview (30 mins)

It was a general HR round. They asked few questions about:

- Family background
- Hobbies
- Challenges faced while developing the project
- Why Zoho and why not other companies?
- Why should we hire you?

1) Alternate sorting: Given an array of integers, rearrange the array in such a way that the first element is first maximum and second element is first minimum.

Eg.) Input : {1, 2, 3, 4, 5, 6, 7}

Output : {7, 1, 6, 2, 5, 3, 4}

2) Remove unbalanced parentheses in a given expression.

Eg.) Input : ((abc)((de))

Output : ((abc)(de))

Input : (((ab)

Output : (ab)

3) Form a number system with only 3 and 4. Find the nth number of the number system.
Eg.) The numbers are: 3, 4, 33, 34, 43, 44, 333, 334, 343, 344, 433, 434, 443, 444, 3333, 3334, 3343, 3344, 3433, 3434, 3443, 3444 ….

4) Check whether a given mathematical expression is valid.

Eg.) Input : (a+b)(a*b)

Output : Valid

Input : (ab)(ab+)

Output : Invalid

Input : ((a+b)

Output : Invalid

I don’t remember the 5th question.

3) Third Round : Advanced Coding
A matrix game was given with 5 rules. We were asked to implement each of the rules separately.

```
R3 | -   -   - |
R2 | -   -   - |
R1 | -   -   - |
     C1  C2  C3
```

Each of the 9 cells can either be empty or filled with an atom. R3, R2, R1 are the rays that originate from the left. C1, C2, C3 are the rays that originate from the bottom of the box.

Input : Position of the atoms and the rays that gets originated from the outside of the box.

```
Eg.) 3
     3 1
     2 2
     1 3
     3
     R3 C1 C3
```

Output : Print the box.

Rule 1:
A ray that has an atom in its path should print ‘H’ (Hit) If it does not have any atoms in its path, the ray should pass to the other side.

```
     C1     C3
R3 | -   -   - | R3
H  | -   X   - |
R1 | -   -   - | R1
     C1  H   C3
```

Rule 2 & 3:
A ray that has an atom in its diagonal adjacent position should refract.

```
H  | -   -   - |
H  | X   -   - |
R  | -   X   - |
     R   H   R
```

Input rays: R1, R2, C3

```
H  | -   X   - |
R2 | -   -   - | C3
   | -   -   - |
     R2     C3
```

Rule 4:
A ray that has atoms in both of the diagonal adjacent positions should reflect back.

Input ray: C2

```
  | -   -   - |
  | X   -   X |
  | -   -   - |
       R
```

Input ray: R2

```
   | -   X   - |
 R  | -   -   - |
    | -   X   - |
```

Rule 5:
The deflection of rays should happen in the order of the input rays.

Input Rays: R3, R2, C1, C3

```
  H | -   X   - |
 R2 | -   -   - | C3
    | -   -   - |
      R2      C3
```

The final task was to implement these rules for dynamic matrix size.

Input : no of rows, no of columns

```
 Eg.) 4 4 (row & column)
      2 (No of atoms)
      4 4 (Position of atom)
      2 2 (Position of atom)
      2 (No of rays)
      R4 C2 (Ray number)
```

```
 H  | -   -   -   X |
    | -   -   -   - |
    | -   X   -   - |
    | -   -   -   - |
          H
```

The final task was very confusing and it had to handle all the cases. There are chances for a ray to end at the starting position if the number of rows and columns are more than 5.

4) Fourth Round : Technical Interview
Basic questions from hashing, searching, sorting, JVM, OS, Threads. In-depth questions from the projects that I mentioned in my resume. So don’t just add projects that you are not thorough enough to answer all questions.
And a simple puzzle : (x-a)(x-b)(x-c)….(x-z) = ?

1. Write a program to give the following output for the given input
Eg 1: Input: a1b10

Output: abbbbbbbbbb

Eg: 2: Input: b3c6d15

Output: bbbccccccddddddddddddddd

The number varies from 1 to 99.

2. Write a program to sort the elements in odd positions in descending order and elements in ascending order
Eg 1: Input: 13,2 4,15,12,10,5

Output: 13,2,12,10,5,15,4

Eg 2: Input: 1,2,3,4,5,6,7,8,9

Output: 9,2,7,4,5,6,3,8,1

3. Write a program to print the following output for the given input. You can assume the string is of odd length

Eg 1: Input: 12345

Output:

```
1     5
  2  4
    3
  2 4
1    5
```

Eg 2: Input: geeksforgeeks

Output:

```
g                 s
  e             k
    e         e
      k     e
        s  g
          f  r
            o
          f r
    s    g
   k      e
  e        e
 e          k
g            s
```

4. Find if a String2 is substring of String1. If it is, return the index of the first occurrence. else return -1.
Eg 1:Input:

String 1: test123string

String 2: 123

Output: 4

Eg 2: Input:

String 1: testing12

String 2: 1234

Output: -1

5. Given two sorted arrays, merge them such that the elements are not repeated
Eg 1: Input:

Array 1: 2,4,5,6,7,9,10,13

Array 2: 2,3,4,5,6,7,8,9,11,15

Output:

Merged array: 2,3,4,5,6,7,8,9,10,11,13,15

6. Using Recursion reverse the string such as
Eg 1: Input: one two three

Output: three two one

Eg 2: Input: I love india

Output: india love I

19 cleared this round and they were called for the next round. The next round took place on the next day

ROUND 3: COMPLEX CODING(3 hours)
1) Design a Call taxi booking application
-There are n number of taxi's. For simplicity, assume 4. But it should work for any number of taxi's.

-The are 6 points(A,B,C,D,E,F)
-All the points are in a straight line, and each point is 15kms away from the adjacent points.
-It takes 60 mins to travel from one point to another
-Each taxi charges Rs.100 minimum for the first 5 kilometers and Rs.10 for the subsequent kilometers.
-For simplicity, time can be entered as absolute time. Eg: 9hrs, 15hrs etc.
-All taxi's are initially stationed at A.
-When a customer books a Taxi, a free taxi at that point is allocated
-If no free taxi is available at that point, a free taxi at the nearest point is allocated.
-If two taxi's are free at the same point, one with lower earning is allocated
-Note that the taxi only charges the customer from the pickup point to the drop point. Not the distance it travels from an adjacent point to pickup the customer.
-If no taxi is free at that time, booking is rejected

Design modules for

1) Call taxi booking

Input 1:

Customer ID: 1

Pickup Point: A

Drop Point: B

Pickup Time: 9

Output 1:

Taxi can be allotted.

Taxi-1 is allotted

Input 2:

Customer ID: 2

Pickup Point: B

Drop Point: D

Pickup Time: 9

Output 1:

Taxi can be allotted.

Taxi-2 is allotted

(Note: Since Taxi-1 would have completed its journey when second booking is done, so Taxi-2 from nearest point A which is free is allocated)

Input 3:

Customer ID: 3

Pickup Point: B

Drop Point: C

Pickup Time: 12

Output 1:

Taxi can be allotted.

Taxi-1 is allotted

2) Display the Taxi details

```
Taxi No:    Total Earnings:
BookingID    CustomerID    From    To    PickupTime    DropTime    Amount
```

Output:

```
Taxi-1    Total Earnings: Rs. 400

1    1    A    B    9    10    200
3    3    B    C    12    13    200

Taxi-2 Total Earnings: Rs. 350
2    2    B    D    9    11    350
```