

Here we have to develop a simple railway ticket booking application.

Develop a Railway Reservation Application contains

1.AC coach

2.Non AC coach

3. Seater

each should contain 60 seats and 10 waiting list max allowed rest request should be cancelled.

you should have

1.Ticket Booking

2.Ticket Cancellation

3.Status Checking

-----  
Here they asked us to create a "Railway reservation system" and gave us 4 modules.  
The modules were:

1. Booking
2. Availability checking
3. Cancellation
4. Prepare chart

We were asked to create the modules for representing each data first and to continue with the implementation phase.

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.

-----

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.

-There 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 |
|---|---|---|---|---|----|-----|

These were just sample inputs. It should work for any input that they give. Those who finished both the modules within 3 hours and if it worked for all the inputs they give, those candidates were given extra modules to work with.

-----  
Real time programming and analysis:

Note: Showing output does matter need to show the output as soon as possible. And also need to solve the constraints very fast, since you know what you have done in your program. After finishing the program always explain the logic behind it and the constraints about the processing and how you solved those constraints to the technical people.

1) To form a structure which has few elements:

```
struct product {
    char productname[20];
    int product_price;
    int product_id;
}
```

Get the product name, price and id and display the product name and price in descending of the price.

2) For the same above structure, now add another structure which is the category. That category will have products in it.

```
Struct category
{
    char category_name[20];
    int cat_id;
}
```

According the category get the product name, product price and id, then display all the products category wise in descending order.

3) For the same structure which as category and product, get the category id from the user in the product structure and save to the category list. Then display them all in category wise.

4) A sheet full of data will be given with inventory stock list, which as different categories and different products as input with category capacity and product availability in the structure. Now we need to add a new category or new product with capacity and availability. Need to check whether the product availability is exceeding the category capacity, if yes the output rack is full or else tell how much free space is available and add the product to list.

5) Constraints in the above in question will be given, need to solve all the

constraints, so that the Technical HR gets satisfied.

After these rounds, if they get satisfied, they will call you for Technical HR, followed by General HR. If you solved every single question and you were really fast in problem solving, then HR interview will be easy. Or else HR interview will be very tough especially the Technical HR (The Technical HR round as lots of logical questions)

-----  
Scenario: text editor

Only 40 characters per line and words should be wrapped if they brake

Also perform insert delete operations

Q1: Given a matrix dimension find the shortest path between two points.

Q2: shortest path between two points and a monster is present. So we have to avoid the monster and take a shortest path or die.

Q3: print the maze step by step.

Q4: same as Q2 but trigger is present so we can take the trigger to shoot the monster anywhere in the maze.

Q5: this is the tricky part. There are many holes in between. we should avoid them and take the shortest path.

-----  
We were asked to design a code (its almost a game without any graphics)  
the question shall be divided into modules and each shall be verified for 7 to 8 test cases. when first module is done only then u can get the second and important thing is all the modules are inter related.

our question was a dungeon game.

An adventurer, A monster, A trigger, A treasure, Pits these are the components.  
The size and location shall be given in run time. Adventurer must reach treasure fast than monster else he dies (Hint: use absolute of distance)

around 5 to 6 modules were present and 3 hour time.

-----  
Third round was followed immediately after completion of second round. 50% candidates were selected for this round which consists of advance programming question. One full fledged project consist of 7 modules which has to be done on their laptops. Test duration was of 3 hours. Test was on 1 Feb 2016.

Description:

There was one question for each candidate.

1. Develop a railway project which consist of 7 requirements modules. The modules includes booking of railway tickets, cancellation, route option, fare calculations, allotting/dealloting berth, coach position.  
Some other questions given to students were online billing and shopping. The instructor may ask to change the requirements run time, you should be ready for it.

-----  
This round contained of advance programing but the logics were easy but they gave more importance to the modularity, Object oriented approach and things like that. They asked to create a simple console based application for e commerce. Two user's admin and customer with different functionalities were given. The Customer has to

buy items then if discount is applicable for that product he can redeem it and he can also view the Order history. The Admin can add new items and so on.

-----  
It was application development kind of programming round where we were asked to develop any application which supports some list of use cases. We were asked banking application with 12 use cases. In this round, the design which we create and the object oriented concepts which we use were monitored. Preferably we were asked to use an object oriented language to code, but those were not good in C++ or java, were also allowed to use procedural language like C.

-----  
About 50 - 60 were shortlisted .. This round was also a programming round.. Questions were based of matrix transformation. Each question was an extension of the previous question in some way.. There were 5 questions .. I solved only 2 ☹ But got selected for the next round , I think it was based on my performance on all previous rounds..

-----  
We were asked to develop a mini-project 'Invoice Management' with the following modules :

1. Add a customer
2. Add an invoice
3. Add items to an invoice
4. List all customers
5. List all invoices
6. List all invoices of a customer
7. Display the full details of an invoice

-----  
We were asked to design an application program for n\*n tic-tac-toe game. Here, you are expected to code with proper standards and in a most optimized way. And, in this round you need to find all the edge cases and corner cases yourself. The interviewers won't help you if you miss anything. So, make sure you covered all the cases before showing output to the interviewers.

Implementation of Tic Tac Toe Game

Only 2 people cleared this round.

-----  
This was an application development round. The application was TOLL PAYMENT PROCESSING .

They insisted us to do it in a object oriented language. First they asked the design( what are all the classes and objects & what data structure do you use).

Application description:

There are 'n' number of points in a highway out of which some points collect toll. Each toll has its own charging scheme according to the vehicles and whether or not they are a VIP user.  
If they are VIP user, 20% discount apply.  
If the vehicle passes 3 toll gates, it has to pay in all the 3 toll gates according to the scheme of respective tolls.  
There were 4 modules.

1. Given the details of vehicle type, start and destination.....display the total toll paid during the journey and print the amount after applying the discount.

2. Display the details of all the tolls....like what are all the vehicles(vehicle number) passed that respective toll and the amount each vehicle paid.....and the total amount charged in that toll.

3. Display the details of all the vehicles .....like what are all the journeys did it take...the start and destination of the same.....tolls it passed during that journey....amount paid in that journey....and the total amount paid by that vehicle.

4. Assume the highway as a circular path.....we have to find the short route and identify the tolls between that route and calculate the amount.

After this round , only 7 students survived for the next round.

-----  
This was also coding round. They asked us to develop a train reservation system. Here also each question was continuation to the previous one(better to write code as function).The task was to write a program to book a ticket, cancel, add to waitlist.... Some constraints were also given.

-----  
This was also coding round. They asked us to develop a train reservation system. Here also each question was continuation to the previous one(better to write code as function).The task was to write a program to book a ticket, cancel, add to waitlist.... Some constraints were also given.

-----  
This is an advanced programming round. The task is to develop an Invoice Management. The main focus in this round is in designing part. We should mainly focus on the data base part. How we are going to design the tables to store data and how we are going manage the data plays an important role. It is better to refresh DBMS concepts before attending the ZOHO placements. Techstud videos on DBMS in youtube helped me a lot to understand basic concepts of DBMS. Particularly, in this round think in the aspect of ER models (Tables, attributes, primary and foreign key etc).

After designing part they gave us two queries. Here, the coding part should be done in a very optimal way. Based on the query, from the data we stored, the result should be printed. I did in C language but better to implement in any OOPS language.

-----  
1. Given a MxN matrix filled with '-' and you need to drop the balloons in the desired columns starting from the bottom. You need to print the matrix when a new balloon is dropped.  
You need to continue getting inputs until the box is full or until the user chooses to stop.

TEST CASE :

Enter the matrix size(m\*n) : 3 3  
Enter the column number : 2

```

Enter the color of the balloon : R
Contents of the matrix :
- - -
- - -
- R -
Do you wish to continue(Y/N) : Y
Enter the column number : 2
Enter the color of the balloon : B
Contents of the matrix :
- - -
- B -
- R -
Do you wish to continue(Y/N) : Y
Enter the column number : 1
Enter the color of the balloon : R
Contents of the matrix :
- - -
- B -
R R -
Do you wish to continue(Y/N) : Y
Enter the column number : 2
Enter the color of the balloon : R
Contents of the matrix :
- R -
- B -
R R -
Do you wish to continue(Y/N) : N
Program Stopped

```

2. Extended version of the previous problem. Now you need to quit when a row become filled completely.

TEST CASE :

```

Enter the matrix size(m*n) : 3 3
Enter the column number : 2
Enter the color of the balloon : R
Contents of the matrix :
- - -
- - -
- R -
Do you wish to continue(Y/N) : Y
Enter the column number : 2
Enter the color of the balloon : B
Contents of the matrix :
- - -
- B -
- R -
Do you wish to continue(Y/N) : Y
Enter the column number : 2
Enter the color of the balloon : R
Contents of the matrix :
- R -
- B -
- R -

```

Column is filled completely. Program is terminated.

3. Extended version of the previous problem. Now you need to drop balloon in the first free cell from left if the specified column is filled in every row.



TEST CASE :

Enter the matrix size(m\*n) : 3 3  
Enter the column number : 2  
Enter the color of the balloon : R  
Contents of the matrix :

- - -

- - -

- R -

Do you wish to continue(Y/N) : Y

Enter the column number : 2

Enter the color of the balloon : B

Contents of the matrix :

- - -

- - -

B R -

Do you wish to continue(Y/N) : Y

Enter the column number : 2

Enter the color of the balloon : R

Contents of the matrix :

- - -

- - -

B R R

Do you wish to continue(Y/N) : Y

Enter the column number : 2

Enter the color of the balloon : R

Contents of the matrix :

- - -

- R -

B R R

Do you wish to continue(Y/N) : Y

Enter the column number : 2

Enter the color of the balloon : B

Contents of the matrix :

- - -

B R -

B R R

Do you wish to continue(Y/N) : N

Program terminated.

4. Extended version of the previous problem. If any column has three continuous balloons of same colors then we need to burst them.

TEST CASE :

Enter the matrix size(m\*n) : 3 3  
Enter the column number : 2  
Enter the color of the balloon : R  
Contents of the matrix :

- - -

- - -

- R -

Do you wish to continue(Y/N) : Y

Enter the column number : 2

Enter the color of the balloon : R

Contents of the matrix :

- - -

- - -

R R -

Do you wish to continue(Y/N) : Y

```

Enter the column number      : 2
Enter the color of the balloon : R
Contents of the matrix      :
- - -
- - -
R R R
Do you wish to continue(Y/N) : Y
Enter the column number      : 2
Enter the color of the balloon : R
Contents of the matrix      :
- - -
- R -
R R R
Do you wish to continue(Y/N) : Y
Enter the column number      : 2
Enter the color of the balloon : B
Contents of the matrix      :
- - -
R R -
R R R
Do you wish to continue(Y/N) : Y
Enter the column number      : 2
Enter the color of the balloon : R
Contents of the matrix      :
- - -
R R R
R R R
Do you wish to continue(Y/N) : Y
Enter the column number      : 2
Enter the color of the balloon : R
Contents of the matrix      :
- - -
R - R
R - R
Do you wish to continue(Y/N) : N
Program Terminated.

```

5. Extended version of the previous problem. Now you need to burst the three continuous colors in the same row.

```

-----
In these rounds we were asked to implement the logic for 2 games
- Minesweeper
- Breakout a.k.a. Arkanoid a.k.a. Brick-Breaker (you'll find it online)

```

The game was split into various stages and we were asked to implement the logic stage by stage.

```

-----
One coding question is asked. The question is to implement the mail server. They
checked for data structures used to solve the problem, authentication method and
some basic things like logout, send, and login modules

```

```

-----
Lift system
There were 8 modules
1. Display the position of Lift

Lift   : L1 L2 L3 L4 L5
Floor: 0  0  0  0  0
2. Assign Lift to the users

```

Input : 2 5

Output : L1 is assigned  
Lift : L1 L2 L3 L4 L5  
Floor: 5 0 0 0 0  
3. Assign nearest lift by comparing their current positions  
Assume,

Lift : L1 L2 L3 L4 L5  
Floor: 5 2 7 9 0  
Input : 4 10  
Output :  
L1 is assigned  
Lift : L1 L2 L3 L4 L5  
Floor: 10 2 7 9 0  
Explanation : L1 is near to 4 floor

4. If two lifts are nearest to the user's source floor, then assign the lift with same direction of user's requirement.  
Example: if user request to move from 4 to 2, and if L3 is in 5th floor & L5 is in 3rd floor, then we should assign L3 because user requested for downward motion so L3 will move down from 5th floor

5. Restrict L1 & L2 for 0-5th floor, L3 & L4 for 6-10th floor, L5 for 0-10th  
Initially all lifts are at 0th floor.

6. Assign lift with least number of stops

Example:

If L3 is in 9th floor

And L5 is at 8th floor

If user wants to move from 8 to 0

We should assign L3 because L3 will stop at 8,7,6 and then 0 NumberOfStops = 3, but L5 will stop at 8,7,6,5,4,3,2,1,0 and NumberOfStops = 8 so we should assign L3

7. Assign capacity (Number of people capable to travel) to all lift and assign according to the capacity

8. If any lift is under maintenance then their current position should be marked as "-1" and that lift should not be assigned at any cost.

-----  
I was asked to come to next round on the end of round 2 itself. Yet another full day of coding. In this round I need to implement a game in C, that's well known as 'Tetris blocks'. It involves full of matrix operations. I completed nearly around 75% of the game. After this I was asked to implement the Binary Search tree-insert, search, delete.

-----  
Those who have solved 4 or above problems has shortlisted for 3rd round. It was a gaming round.

Size of the array is given where w is wall, g ground, o ball, numbers are bricks.

I/p size 7

Number of bricks 6

Position (2, 2)(2, 3)(2, 4)(3, 2)(3, 3)(3, 4)

Ball:5(life)

w w w w w w w

```

w                                w
w      1  1  1      w
w      1  1  1      w
w                                w
w                                w
w  g  g  o  g  g  w

```

There are three commands St, lt, rt straight, left, right respectively.

If it is st the ball moves straight .if there any brick(1) along the way it hit it .then brick disappear.ball back to original position.if there is no brick .it come to initial position.

I/p st

O/p

```

w w w w w w w
w                                w
w      1  1  1      w
w      1      1      w
w                                w
w                                w
w  g  g  o  g  g  w

```

Ball count:5

I/p lt

O/p

```

w w w w w w w
w                                w
w      1  1  1      w
w                                1      w
w                                w
w                                w
w  g  o  g  g  g  w

```

Ball count:4

(Lt : ball moves diagonally left there is no brick on the way so it hit the wall

reflect back horizontally there is a brick(3, 2) after hitting it it moves downwards ball position changed.hence ball count get reduced. On moving downwards is there any brick that also disappear.)

Same for rt but moves diagonally right.

This is first module.In second module each brick has value for each hit value get reduced.it disappear only when it become zero.

There are about 7 modules.

Those who have clear at least first module.selected for fourth round.

-----  
Given an employee data base.

Name, Age, Designation, Department Of ten people.

and Five tasks were given such as

1. Print all employee details.
2. Searching employee details
3. Employees under the given manger name of the department
4. reporting to tree of the given employee name

-----  
This round is mainly based on data structure and oops concepts.  
No inbuilt collections are allowed. You need to implement on your own.

Needs to discuss your approach before start solving the problem.

Design a system with following functionalities,

SET a variable  
GET a variable  
UNSET a variable  
COUNT NUMBERS OF VARIABLE with given value  
BEGIN – Begins a new transaction  
ROLLBACK – Roll back all the commands in the open transaction  
COMMIT – Commit the transaction  
EXAMPLE 1:

```
SET a 20
GET a 20
SET b 30
GET b 30
SET a 10
GET a 10
UPDATE c 40 No variable named "c"
SET c 30
COUNT 30 2
COUNT 40 null
UNSET a
GET a null
```

EXAMPLE 2:

```
GET a null
SET a 30
GET a 30
```

EXAMPLE 3:

```
SET a 30
BEGIN
GET a 30
SET a 40
GET a 40
SET b 40
GET b 40
ROLLBACK
GET b null
GET a 30
```

EXAMPLE 4:

```
BEGIN
SET a 40
SET b 40
SET c 50
COUNT 40 2
BEGIN
COUNT 40 null
COMMIT
COUNT 40 2
BEGIN
SET c 10
GET c 10
ROLLBACK
GET c 50
```

-----  
One application will be given. Time: 2h30m

Write an application for booking railway ticket reservation system. The application should have four functionalities.

Book  
Cancel  
Print booked tickets (details with summary)  
Print available tickets (details with summary)  
Conditions for booking:

There are a total of 63 berths for 63 confirmed tickets, 9 berths for 18 RAC tickets and 10 tickets in waiting-list. If the waiting-list ticket count goes above 10, print as 'No tickets available'. The following passenger details should be obtained from the user.

Name  
Age  
Gender  
Berth Preference

The tickets should not be allocated for children below age 5. But, their details should be stored. Lower berth should be allocated for persons whose age is above 60 and ladies with children if available. Side-lower berths should be allocated for

RAC passengers.

Conditions for cancelling:

Whenever a ticket is cancelled, a ticket from RAC should be confirmed and a waiting-list ticket should move to RAC.

Conditions for printing booked tickets:

Print all the tickets that are filled along with the passenger details and at the end, print the total number of tickets that are filled.

Conditions for printing available tickets:

Print all the tickets that are unoccupied and at the end, print the total number of tickets that are unoccupied.

-----  
Many students will be able to solve this round. So try to perform best in round 2.

You will be asked to explain before starting. Try to figure out the boundary testcases. They asked us to implement a gift card system with a existing bank.

The bank has many customers. Each customer can create many gift card. Balance should be detected from bank balance when a gift card is created. A gift card has a id, pin. You can Top-up any gift card, close gift card and buy item using gift card (they are called transactions), etc.

Write clean code and use object oriented language.

Boundary case are :

Customer does not exist  
Money is negative  
=-----

This time we were asked to develop the good old BOMBERMAN game. There were six modules, each of which is an improvement of the previous module. We were evaluated at the completion on each module. The each module were given with very clear explanation and sample test cases. The HR's were friendly and cleared the doubts of the students. Around twenty were selected for the next round.  
=-----

we were told to develop an application on a given environment (Chess tournament). we were given 3 hours to write the code. They mainly focused on data structures used, how neatly we were writing the code and how well we are able to fix the bugs in it.  
=-----

The third round consisted of module coding. As told before, this was the first round on the second day.

The module given to me was Flight Reservation system. There were several modules where we had to implement functionalities. Some are mentioned below:

Get input from two flights - 101 and 102 with different passengers for each flight. After booking seats, display the seat number to the passenger.

The meal has an additional charge which has to be included

Surge pricing was that every ticket (business/ economy) was incremented by a specific price after each successful booking.

A flight summary has to be displayed.

Cancellation of flights should be done. The refund should be calculated properly.

I used structures and file manipulation for completing the module round.  
-----

The bank has initially three customers. There were around eight modules .

Account Login  
Purchase  
Account Login

Giving customer id and password . Password should be encrypted and stored

Encryption is like A-> B, B->C

a-> b, b->c, c->d

0->1, 1->2

On successful login, it should print the account details

1. Create Gift Card
2. TopUp
3. Transaction History
4. Block
5. Logout

1. Create Gift Card

Gift card with 5 digit card no and 4 digit pin number will be generated

2. TopUp

For topup, amount need to be reduced from main account balance and added to gift card

3. Transaction History

Should print all the transaction details of a particular gift card

4. Block

If the card is blocked, shouldn't be available for topUp, Purchase. the amount in gift card should be transferred to main account

5. Logging Out

After log out, should go to main module,

2. Purchase

Login to the gift card

Purchase Amount



Then print Available balance

#### 7.Redeem points :

For Every 100 rupee purchase, 1 reward point is added .For 10 reward points, 10 will be added to main account

#### 8.Doing for Multiple gift cards

Around 5 to 6 of us who completed 6 to 7 modules were shortlisted to next round

-----  
This is called as complex coding round in which are given with a train ticket reservation problem in which extra feature enhancement is given one after another.

first simple reservation for 50 seats from A to F.

After stations are added in middle like B, C, D, E. Where people can board any station and get down at any station assuming train is going from A to F.

Then Waiting list are to be added after 50 seats reserved and if any reserved seat is cancelled we have to check the waiting list and need to confirm the seat if possible.

After which we need to print the details of the PNR details.

-----  
Third round was application development round. Question which I got was "Event Management". There were around 20 events which are about to happen. Manage all the events according to the given conditions.

List of events are given below none of them has any specific order to be followed.Conditions

are :1. Event should start from 9.00 AM

2. No overlapping or any time gap between two events.

3. The last event should complete by 4 or to maximum by 5. There is a compulsory event

called "Networking hands-on" which has to be started not earlier than 4.00 PM nor later

than 5.00 PM. The remaining events should be scheduled following day until all the events are covered.

4. Every event has specific duration mentioned along with it and some events contain a

keyword "lightning" which means that particular event's duration is 5 minutes.

5. There should be no event scheduled between 12 to 1 PM and kept for lunch.

Input:

Welcome event 30 mins

C programming 45 mins

Working with Java Beans 30 mins

Ruby on Rails programming 60 mins

Introduction to Groovy 60 mins

Rails Debugging 45 mins

Tips and tricks in C 30 mins

Back-end development in MySQL 50 mins

Sit down and Take notes lightning

Clojure Introduction 45 mins

Team Management Concepts 30 mins

Introduction to Java Frameworks lightning

Working with Angular JS 45 mins

Ruby on Rails programming web development concepts 60 mins

Introduction to Kotlin Java 60 mins

Debugging and Testing products 60 mins

Documenting a software 40 mins

Server side development 60 mins

Output:

Schedule for Day 1

09:00 AM Welcome event 30 mins  
09:30 AM C programming 45 mins  
10:15 AM Ruby on Rails programming 60 mins  
11:15 AM Rails Debugging 45 mins  
12:00 PM LUNCH  
01:00 PM Working with Java Beans 30 mins  
01:30 PM Introduction to Groovy 60 mins  
02:30 PM Tips and tricks in C 30 mins  
03:00 PM Back-end development in MySQL 50 mins  
03:50 PM Sit down and Take notes lightning  
03:55 PM Clojure Introduction 45 mins  
04:40 PM Networking Hands-on

Schedule for Day 2

09:30 AM Team Management Concepts 30 mins  
09:30 AM Introduction to Java Frameworks lightning  
09:35 AM Working with Angular JS 45 mins  
10:20 AM Ruby on Rails programming web development concepts 60 mins  
11:20 AM Documenting a software 40 mins  
12:00 PM LUNCH  
01:00 PM Introduction to Kotlin Java 60 mins  
02:00 PM Debugging and Testing products 60 mins  
03:00 PM Server side development 60 mins  
04:00 PM Networking Hands-on

-----  
Third Round(Advanced coding round to be done in C/ C++/ Java) : This happened the same day after lunch. This round consisted of one question and duration was 3 hours. The question was about Railway Reservation System which consisted of two things which are booking and cancellation and there were many constraints which you need to study very carefully. The question was a bit tough to explain as it is very length and composed of many constraints. But as it was tough none of us were able to complete the question in the given time and we were asked to leave and none of us cleared the round.

I was very much disappointed as I couldn't solve the question, but I was a bit satisfied that I have tried my best. Guess what I got one more mail from Zoho after 5 days on November 4, 2019. I was asked to undergo Incubation in Zoho for performance check and it started on November 6, 2019 and I was put in a team known as Manage Engine Mobile which is a mobile application development team and I was asked to learn Kotlin and I've been assigned to a mentor. Mentor was very much helpful and the team was very good and I was learning there nearly a month. After that a technical interview was conducted by the manager of the same team and he checked what I've learned during incubation and then a formal HR interview was conducted and finally by god's grace I was given the offer for the role of android developer. They checked only whether I was open to learning or not during incubation.

Zoho Corporation is a leading product based company which is serving as an operating system for business and it has a wide variety of products which are competing with google in the market. I would suggest you all to learn C programming in depth and know the OOPS concepts in C++ and then go for learning Data Structures and algorithms and all you need to do is bloody coding. GeeksforGeeks helped me a lot for learning variety of things related to programming and it is an ocean for learning technical things and thanks to Interview Preparation course by Ravindra Babu Ravula which helped me a lot for learning programming, data structures and

algorithms.

-----  
Question was to build an Employee record management app with 5 tasks to be executed to qualify the round.

we can hard code the employee records in any data structure of our choice.

Columns of the table are: EmpName, Age, Designation, Department, ReportingTo

Task 1: Display all the Employee records in a table format.

Task 2: Search records based on user give criteria.(this task is divided into two sub tasks)

a. display option according to the column type 1.String - Equals, Notequals, Contains, Notcontains, startswith, endswith 2.Integer- <, >, =, !=, between.

b. apply the same for multiple conditions using 'AND' for each condition. (similar to 'where' class in SQL using multiple columns).

Task 3: print reporting tree of the given employee.

Task 4: print the employees reporting to the given manager.

Task 5: print summary of Department, Designation, ReportingTo.(menu driven)

-----  
Create an engine that can process the user query. The main focus is not the logic but System Design.

How the created query engine scales out perfectly even upon adding new features later?

How do we create a system that can handle the following inputs and process the input query?

Question:

Given a table containing a set of 10 employees with respective fields:

| ID | Name | Age | Designation | Department | Reporting To |
|----|------|-----|-------------|------------|--------------|
|----|------|-----|-------------|------------|--------------|

Show all employee data

Process the query:

Get input from the user until presses exit.

Get field value to compare, comparison operator as input

If the field value is age (int data type), supported comparators: >, <, !=, ==

If the field value is of string data type, supported comparators: 'startswith', 'contains', 'endswith', 'notcontains', 'equals' and 'notequals'.

Use 'AND' in default for queries with multiple checks.

Eg: age > 30 and age < 50 and department contains finance and reporting to A

Show the reporting to hierarchy for the given employee name: J -> I -> F -> D -> C -> B-> A

Show the employees reporting to the given manager.

Show summary of Department, Designation, ReportingTo.

The entire application is to have a menu and the user could be able to select from the menu.

-----  
It was System Design round for 2 hours time duration. Design an In-Memory file management system.

Module1: Create a new directory and file in all Levels.

Module2: List all directories and files.

Module3: Update file content.

Module4: Update directory and file names.  
Module5: Delete directory and file.  
Module6: Restore deleted directories and files.

-----  
The question was to develop a spreadsheet with the following functionalities. First we were asked to explain our method of approach towards the problem only then we were allowed to code.

Print all the cells in the spreadsheet  
Display the result in the cell by evaluating the expression (without precedence and order of evaluation)  
Display the result in the cell by evaluation the expression (using precedence and order of evaluation)  
To check if there is any error in the given expression  
To check if there is any circular dependency between the given cells  
Change the contents in the given range of cells if a certain condition is satisfied

-----  
You are probably expecting an OOP-based application like Railway reservation, Traffic control, Lift control, etc, but instead, I got something new.

An algebraic problem, just one question for 3-4 hours.

Write a program to multiply two or more algebraic expressions and print the result.

Examples:

Input:  $(2x+y)*(3x-5y)$   
Output:  $6x^2-7xy-5y^2$   
Input:  $(2xy+4x^2y)*(2x^2y+6xy)$   
Output:  $28x^3y^2+8x^4y^2+12x^2y^2$   
Input:  $(2x^2y+3xy^2z-xz^3)*(5xyz+3y^2z-2z)$   
Output:  $10x^3y^2z+6x^2y^3z-4x^2yz+15x^2y^3z^2+9xy^4z^2-6xy^2z^2-5x^2yz^4-3xy^2z^4+2xz^4$   
-----