Round 1 :
Q1) Finding longest range of continuous 0s in a binary array.
Q2) Implementation of LRU cache
Q3) return true/false given string s(binary string) such that max size of substring containing only 1 is greater than and max size of substring containing only zero containing only zero.
Q4) Left View of Binary Tree
Q5) unordered_map implementation details. Eg: When does it have O(n) complexity.
Q6) Implement hash function for a class. (Just syntax) Solution: Implement hash<Class>() and operator==()
Q7) Where are local,global, dynamic memory variables stored
8) implementation of getmin,top,pop function using a stack (I was allowed to use array only)
9) implementation of vector
10)LRU Cache Implementation, Concept of Hashing(Chaining Technique, How order and unordered map implements, Discussion on hash functions)
Q13)Given a vector. Find the number of distinct elements for every window of size k.
Q12) Find the output of the C code
Q13)implementing upper bound in vector for a given value x
Q14)create a employe registration system involving keys,designation and other parameters.(More concerned with time complexity.)

```
#include <stdio.h>
struct node{
   int a;
   int b;
};
```

Q15)

```
void main(){
   char str[20] = {0};
   struct node* tmp = (struct node*) str;
   printf( "original ptr 0x%x\n", str);
   printf( "original tmp 0x%x\n", tmp);
   printf( "tmp 0x%x str 0x%x\n", tmp + 1, str + 1);
}
```

Round 2 :
Q1) Oops (was asked to show inheritance) Q2) Implementation of malloc() and free() q3) Questions related to memory hierarchy (where are local global variables, objects etc stored in memory)
Q4) Given a very big array find the largest k elements w/o sorting (heap soln)
Q5) Implement a LRU Cache with show cache() and update cache() functions.

Yash Sharma

Round 1:

I was asked to code in my local editor. The whole exact txt file is in the doc below with my comments added afterwards for questions.

Arista Internship '21 Interview Questions

Round 2:

Q1: Design a To-Do app.
Q2: Design a system for elevator keeping in mind different variables to use, functions to make.