1. I’m interested in Dev team, for me developing something is more challenging than testing an existing thing.
2. I don’t have favorite programming language, but now I’m using Java and it’s comfortable for me.
3. Linked list is a data structure. It consists of nodes which contain data and pointer to next node. It is used when constant time insertion or deletion in any position is needed and if random access is not mandatory.
4. It’s not their fault I stepped back. So I would serve them as I did before, I will ask for help from my colleague if possible.
5. I would test if it closes safely, if it works fine with different types of papers, is it comfortable for hand and does it have enough writing capacity.
6. I would test if money transactions are done properly.
7. Using sum of all elements. Subtract sum of all elements from sum of numbers 1 +2+…+1million which is (n\*(n+1)/2), where n is one million, the answer will be missing number.
8. I would compute the angel from 12:00 to hour-hand, every hour is 360/12 = 30 degree, let’s assume it d1 degree, then do the some for minute-hand every 5 minutes is 30 degree, let’s assume we get d2 degree, then compute x = abs(d1-d2). The answer would be min(x, 360-x)
9. If x&(x-1) ==0 is power of yes, and no otherwise
10. 1.put all elements to set

2.put from set to list

1. Count how many times each character occurs in string, using bucket array, or hash table, then make sure each element occurs <=1 times
2. For(int i=0; i<s.size()/2; ++i) //in c++  
    swap(s[i], s[n-i-1]);
3. Iterate string and if element is not the last and equals it’s next element count++,  
   else answer += s[i] + to\_string(count)
4. indexOfEnd = size-1; for(i=[0,size-1]) if(arr[i]=1) then swap(arr[i],arr[indexOfEnd])
6. Suppose the minimum number of drops in worst case is x, if we drop a lightbulb some floor higher than x, say x+1, if it breaks we have no other option than to start dropping second lightbulb from 1st floor up to x, that means we have done x+1 drops, which is > x, but x in worst case we should do <= x drops, so we can never drop 1st lightbulb higher than x, throwing somewhere lower than x is not optimal, so we will drop first lightbulb throw at x.  
   if the lightbulb does not crash we can forget about floors <=x, and we come to same problem but as we have already done one drop we should solve it in at most x-1 drops.  
   so first we throw from x than floors higher x-1 then x-2 ..  
   so we get x + x-1 + x-2 … +1 = 100, if we solve this will get x=14
7. Place it in the corners
8. 2 weights (3:3 //3=>1:1//1)
9. I would use backtracking
10. I would find LCS(longest common subsequence) of s, and reverse(s), using dynamic programming