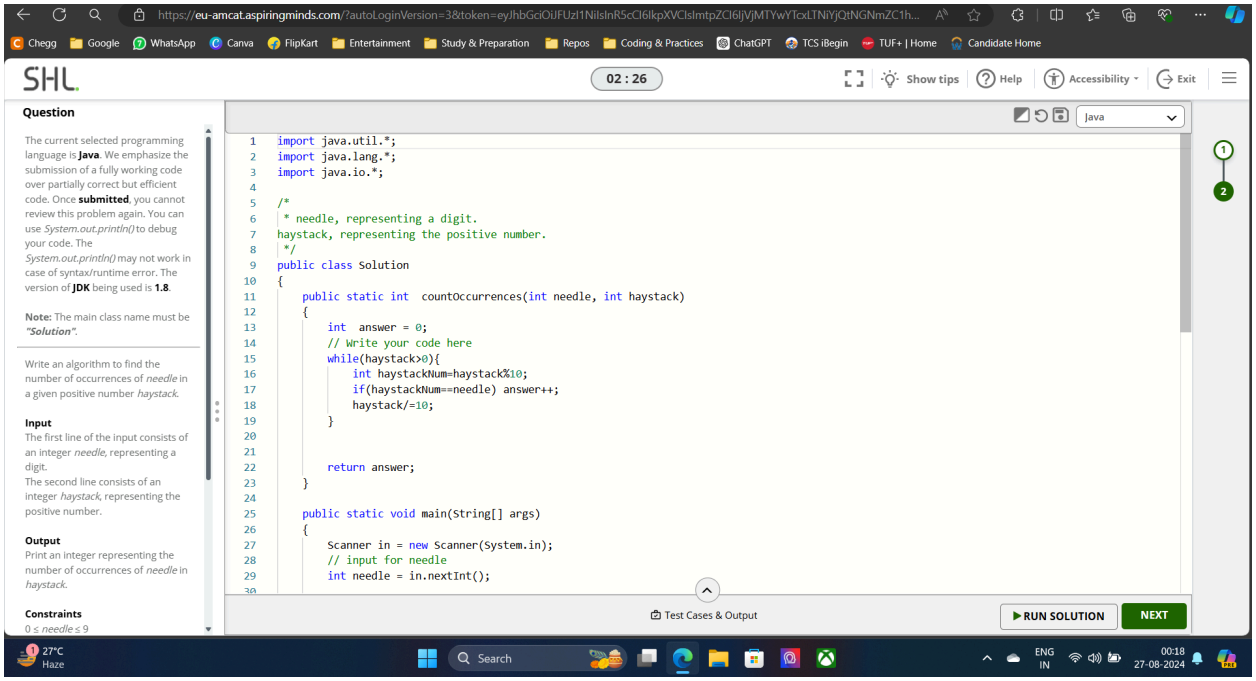
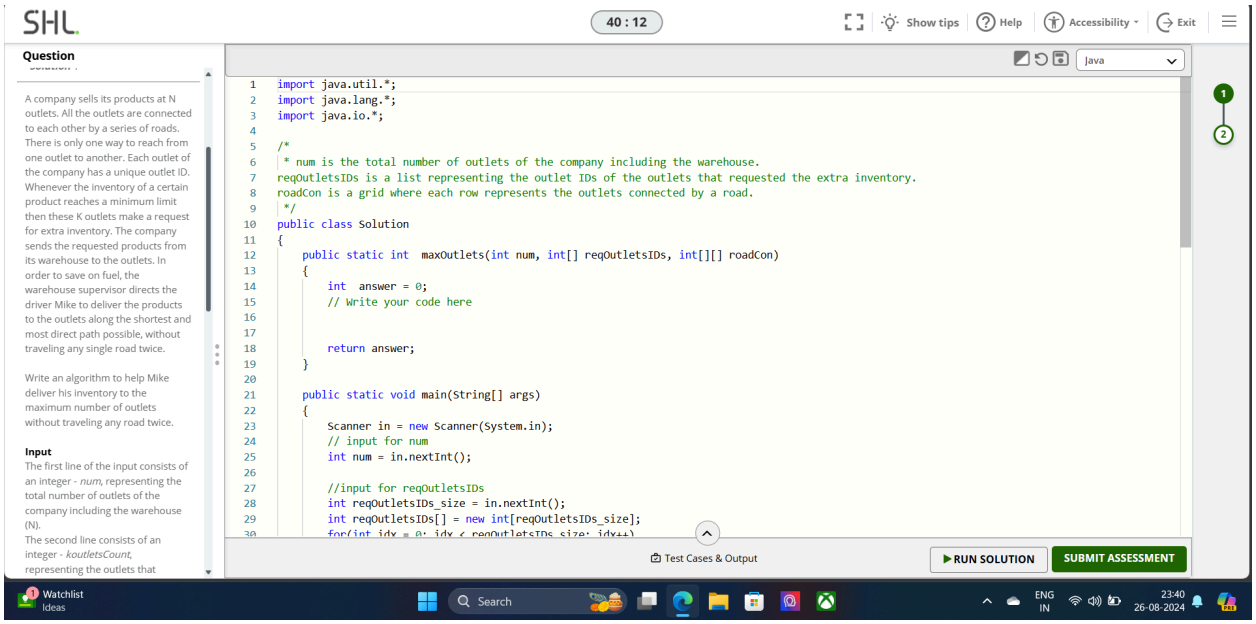
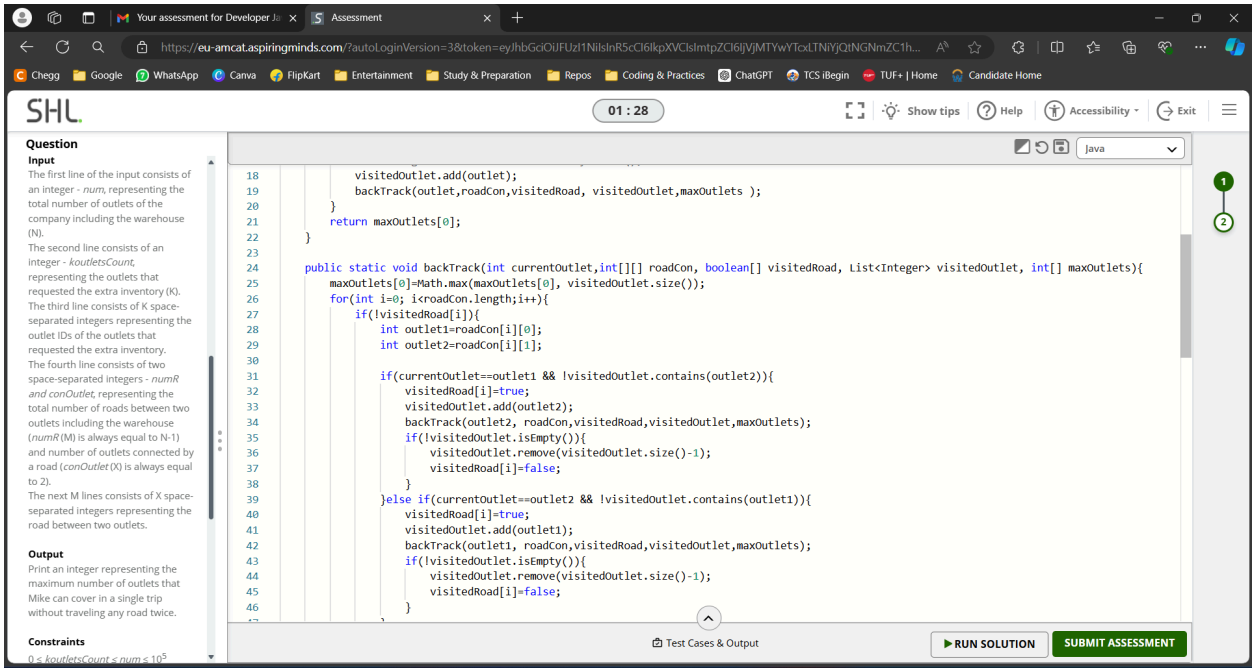
SAP Labs Coding Assessment

Question1:



Question 2:




Your assessment for Developer Ja
Assessment
https://eu-amcat.aspiringminds.com/?autoLoginVersion=3&token=eyJhbGciOiJFUzI1NiIsInR5cCI6IkpXVCIsImtpZCI6IjVjMTYwYTcxLTNiYjQtNGNmZC1h...
Chegg
Google
WhatsApp
Canva
FlipKart
Entertainment
Study & Preparation
Repos
Coding & Practices
ChatGPT
TCS iBegin
TUF+ | Home
Candidate Home
SHL
01 : 28
Show tips
Help
Accessibility
Exit
Question
Input
The first line of the input consists of an integer - num, representing the total number of outlets of the company including the warehouse (N).
The second line consists of an integer - koutletsCount, representing the outlets that requested the extra inventory (K).
The third line consists of K space-separated integers representing the outlet IDs of the outlets that requested the extra inventory.
The fourth line consists of two space-separated integers - numR and conOutlet, representing the total number of roads between two outlets including the warehouse (numR (M) is always equal to N-1) and number of outlets connected by a road (conOutlet (X) is always equal to 2).
The next M lines consists of X space-separated integers representing the road between two outlets.
Output
Print an integer representing the maximum number of outlets that Mike can cover in a single trip without traveling any road twice.
Constraints
0 ≤ koutletsCount ≤ num ≤ 10^5
Java
visitedOutlet.add(outlet);
backTrack(outlet,roadCon,visitedRoad, visitedOutlet,maxOutlets);
}
return maxOutlets[0];
}
public static void backTrack(int currentOutlet,int[][] roadCon, boolean[] visitedRoad, List<Integer> visitedOutlet, int[] maxOutlets){
maxOutlets[0]=Math.max(maxOutlets[0], visitedOutlet.size());
for(int i=0; i<roadCon.length;i++){
if(!visitedRoad[i]){
int outlet1=roadCon[i][0];
int outlet2=roadCon[i][1];
if(currentOutlet==outlet1 && !visitedOutlet.contains(outlet2)){
visitedRoad[i]=true;
visitedOutlet.add(outlet2);
backTrack(outlet2, roadCon,visitedRoad,visitedOutlet,maxOutlets);
if(!visitedOutlet.isEmpty()){
visitedOutlet.remove(visitedOutlet.size()-1);
visitedRoad[i]=false;
}
}else if(currentOutlet==outlet2 && !visitedOutlet.contains(outlet1)){
visitedRoad[i]=true;
visitedOutlet.add(outlet1);
backTrack(outlet1, roadCon,visitedRoad,visitedOutlet,maxOutlets);
if(!visitedOutlet.isEmpty()){
visitedOutlet.remove(visitedOutlet.size()-1);
visitedRoad[i]=false;
}
1
2
Test Cases & Output
RUN SOLUTION
SUBMIT ASSESSMENT