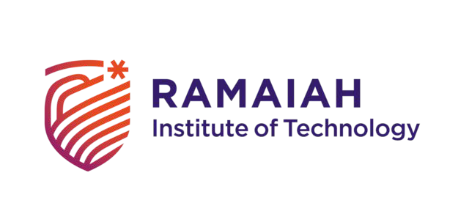
RAMAIAH INSTITUTE OF TECHNOLOGY

MSRIT NAGAR, BENGALURU,560054

Report on

LEET CODE SOLUTIONS USING C

DATASTRUCTURES LAB ( IS36 )

Submitted by

Candidate Name: -

U Sanjay (1MS22IS146)

Under the Guidance of

Faculty Incharge

Mr. Shivananda S

Assistant Professor

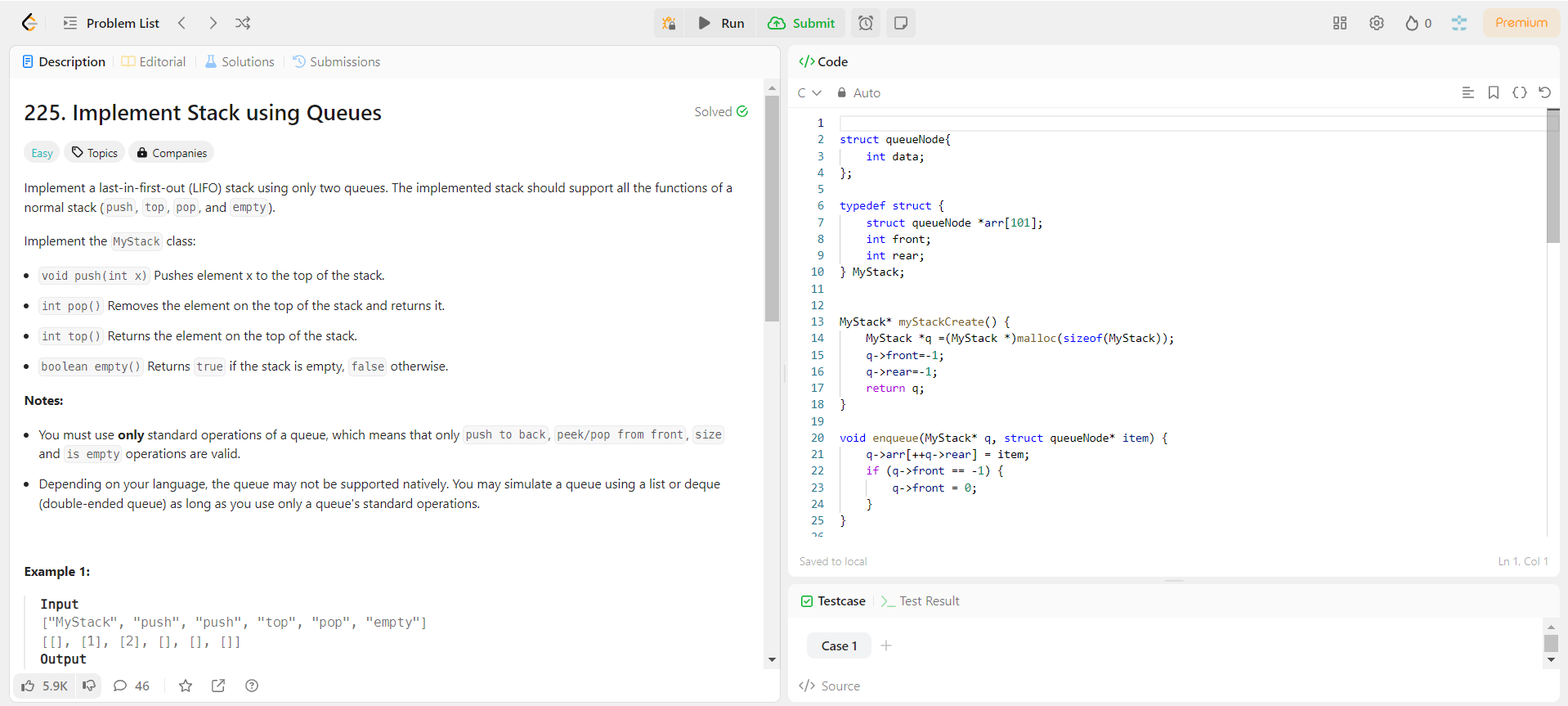
Dept of ISE

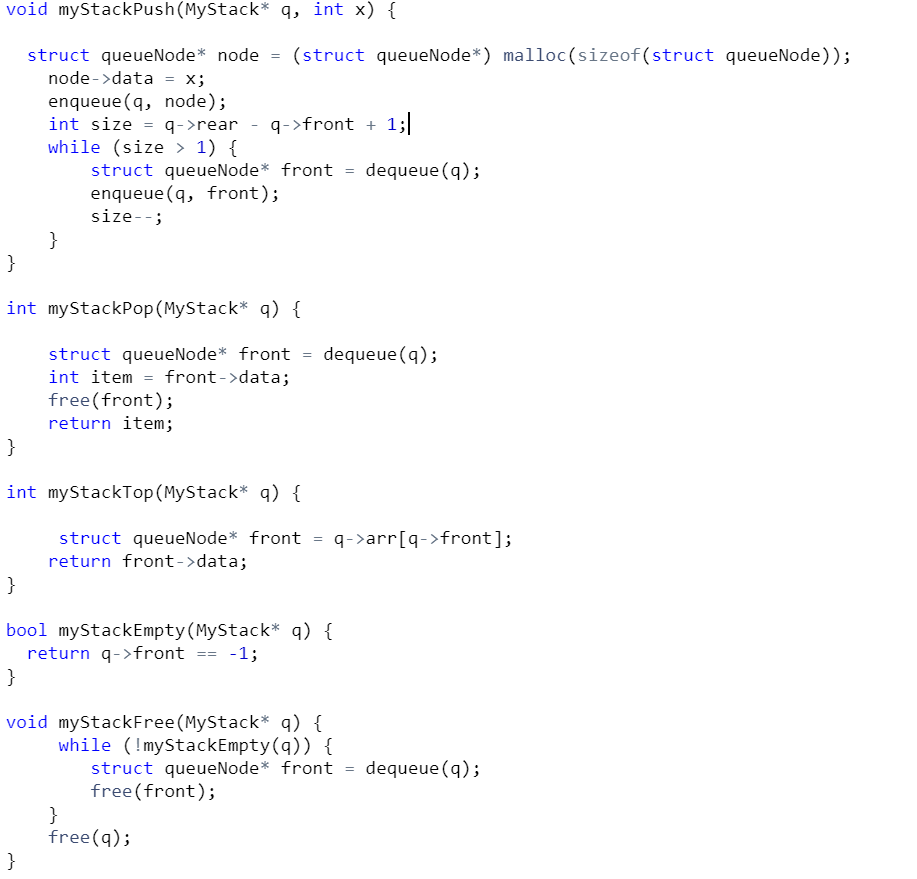
Department of Information Science and Engineering

Ramaiah Institute of Technology

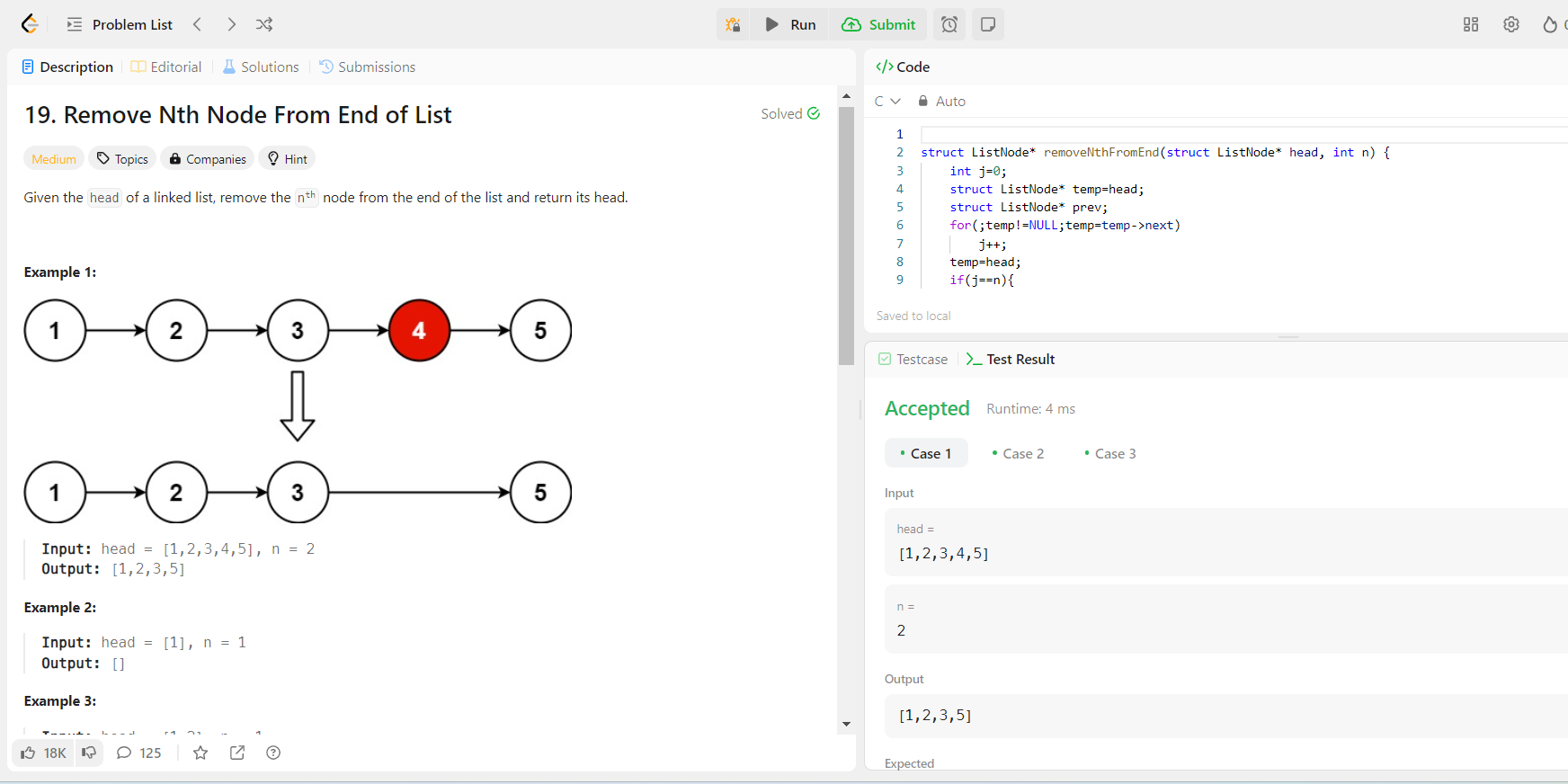
2023 – 2024

1.Implement Stack Using Queue



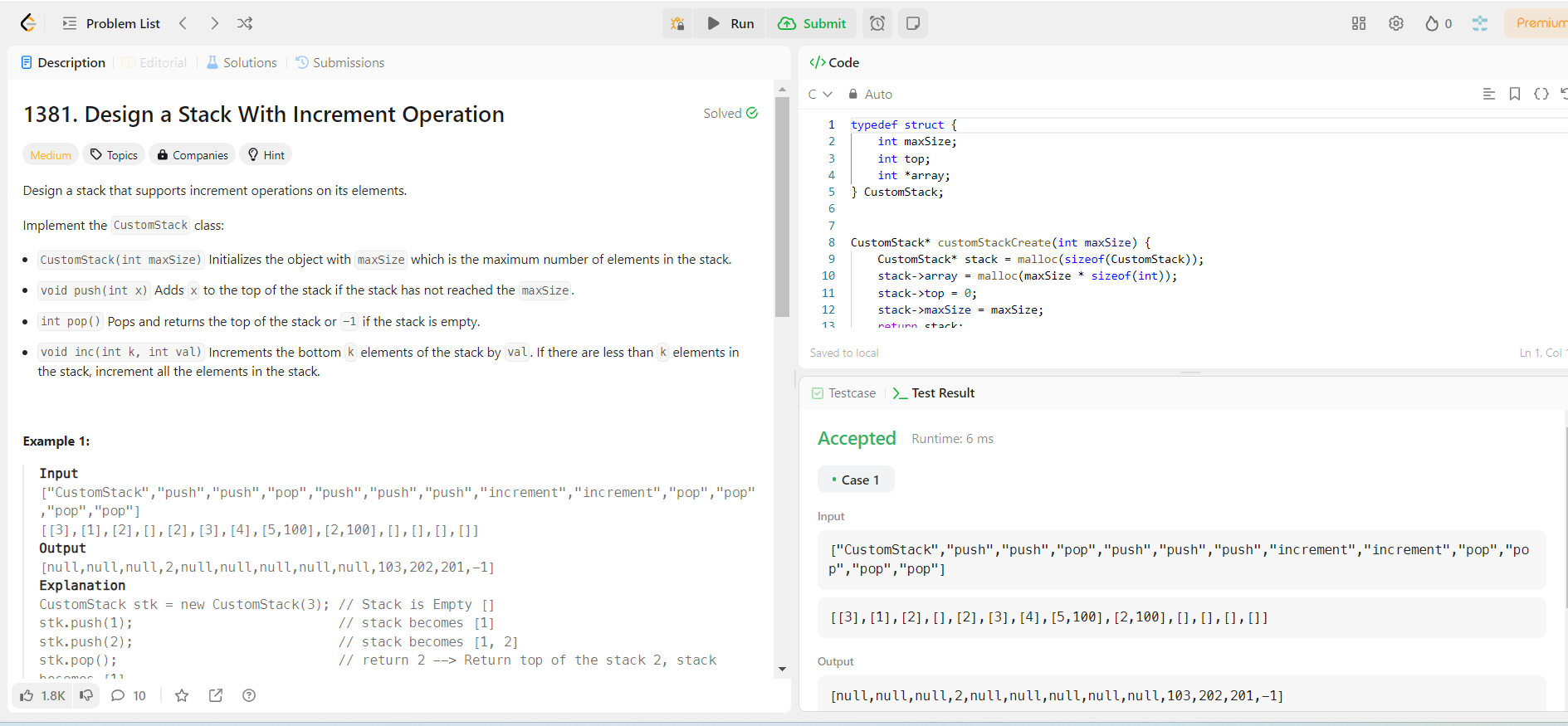
 

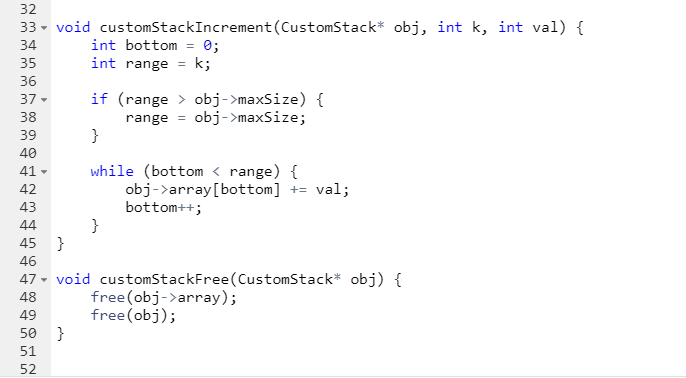
2. Remove nth Node from End of List



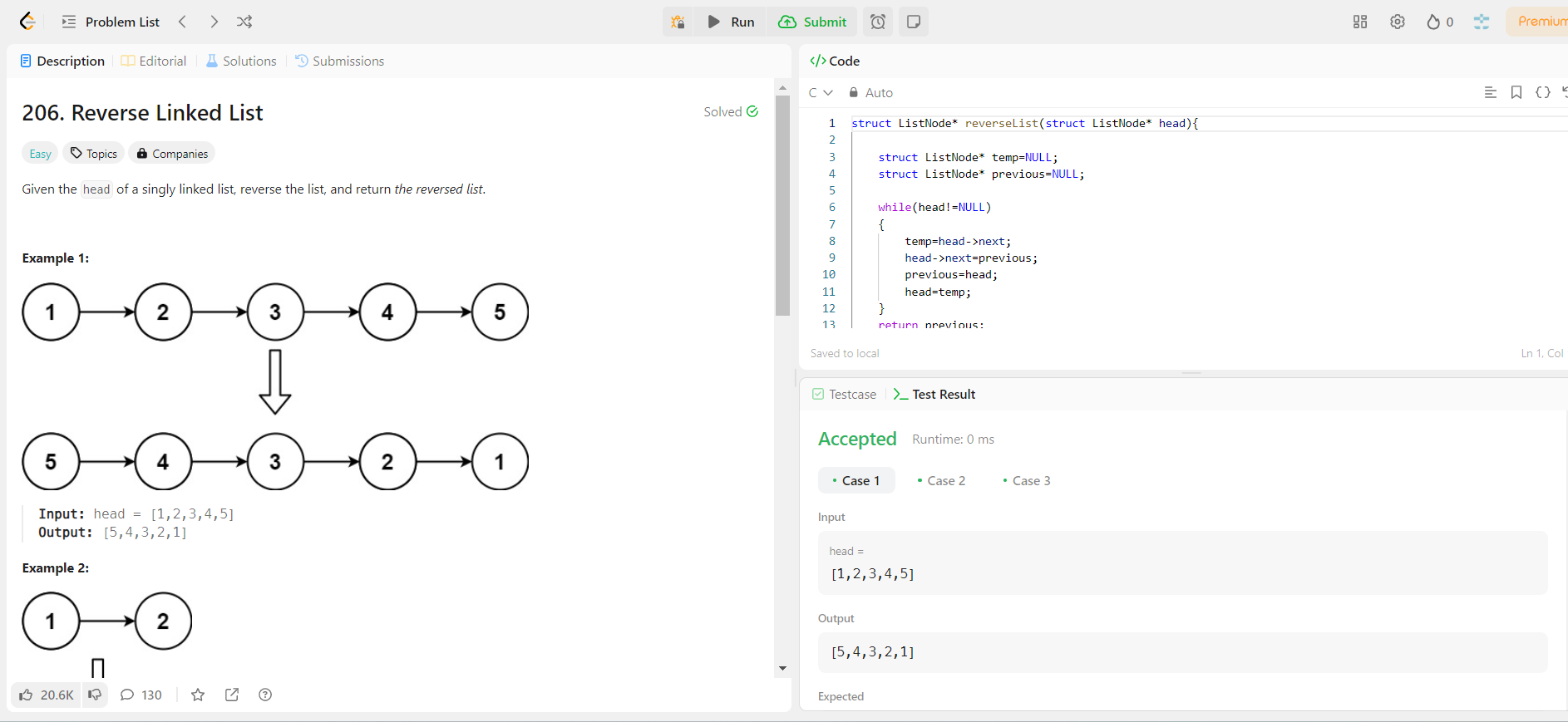


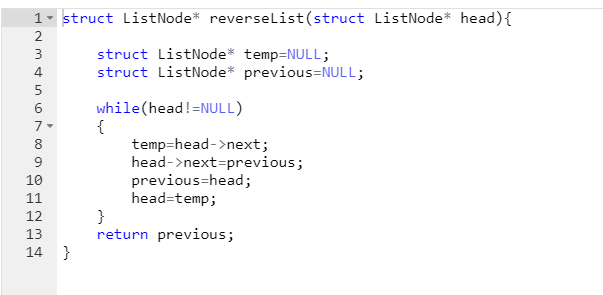
3. Design a Stack with Increment Operation



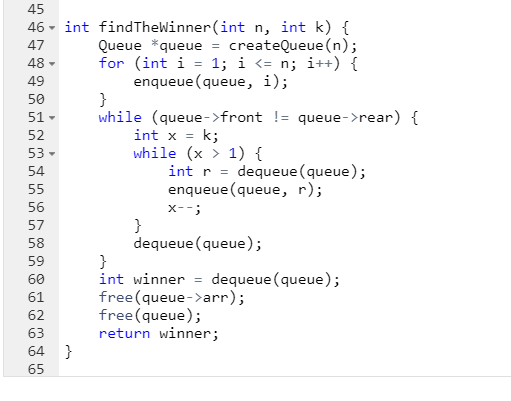
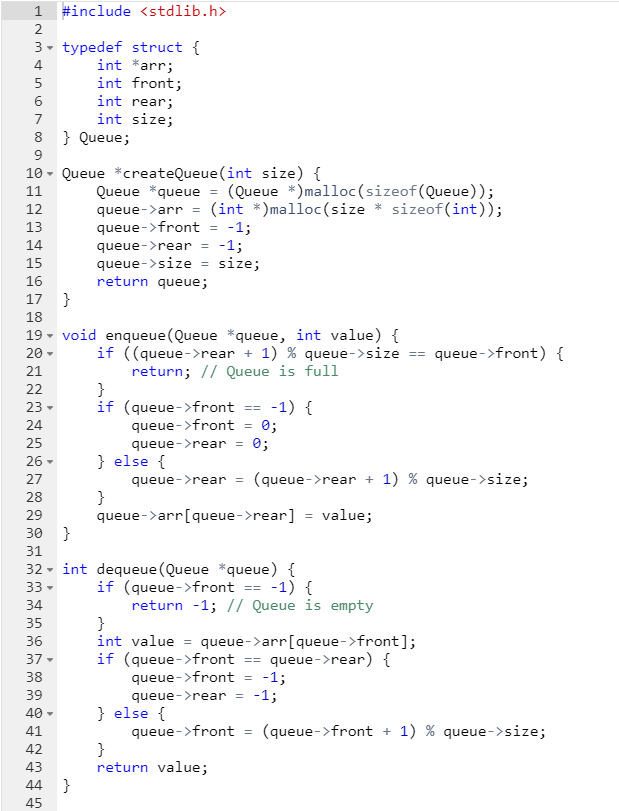
 

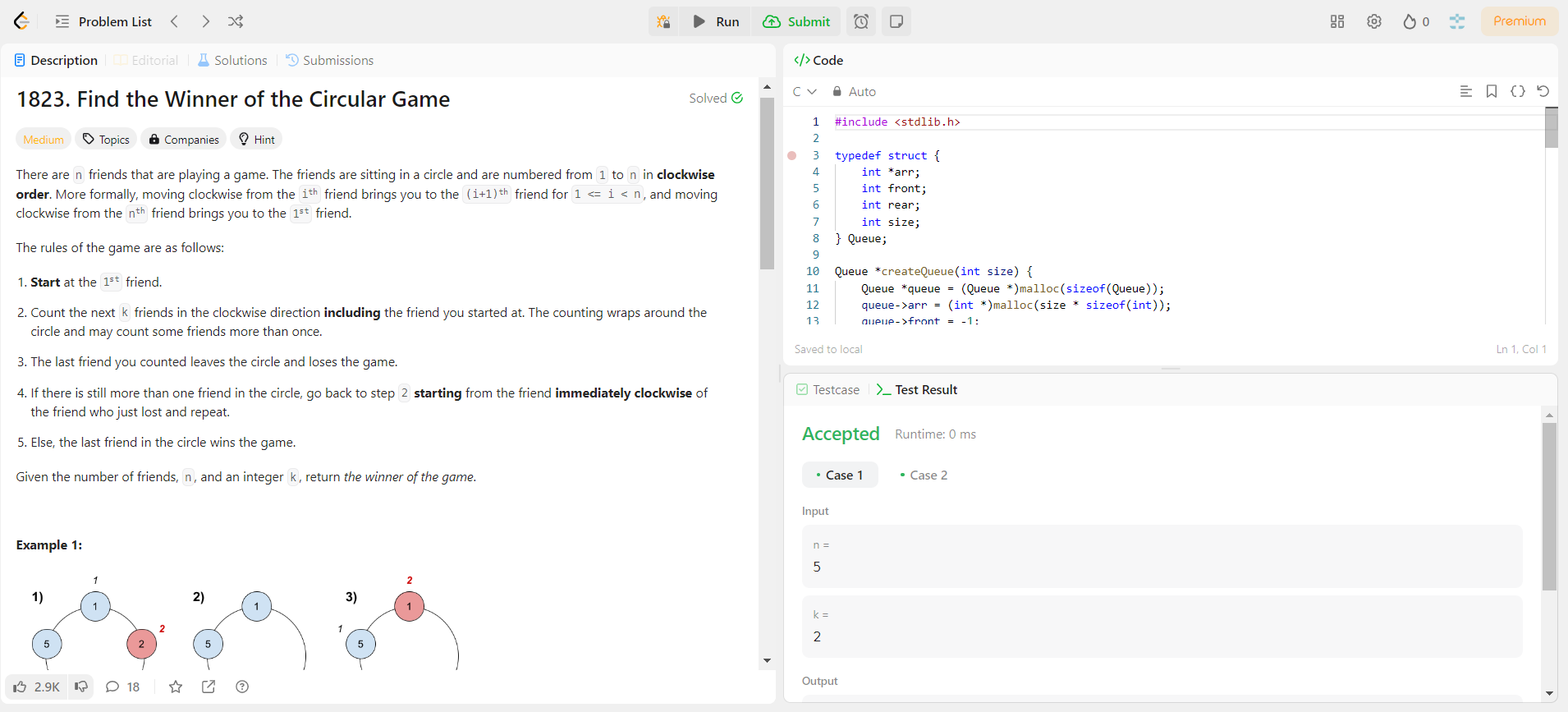
4. Reversed Linked List



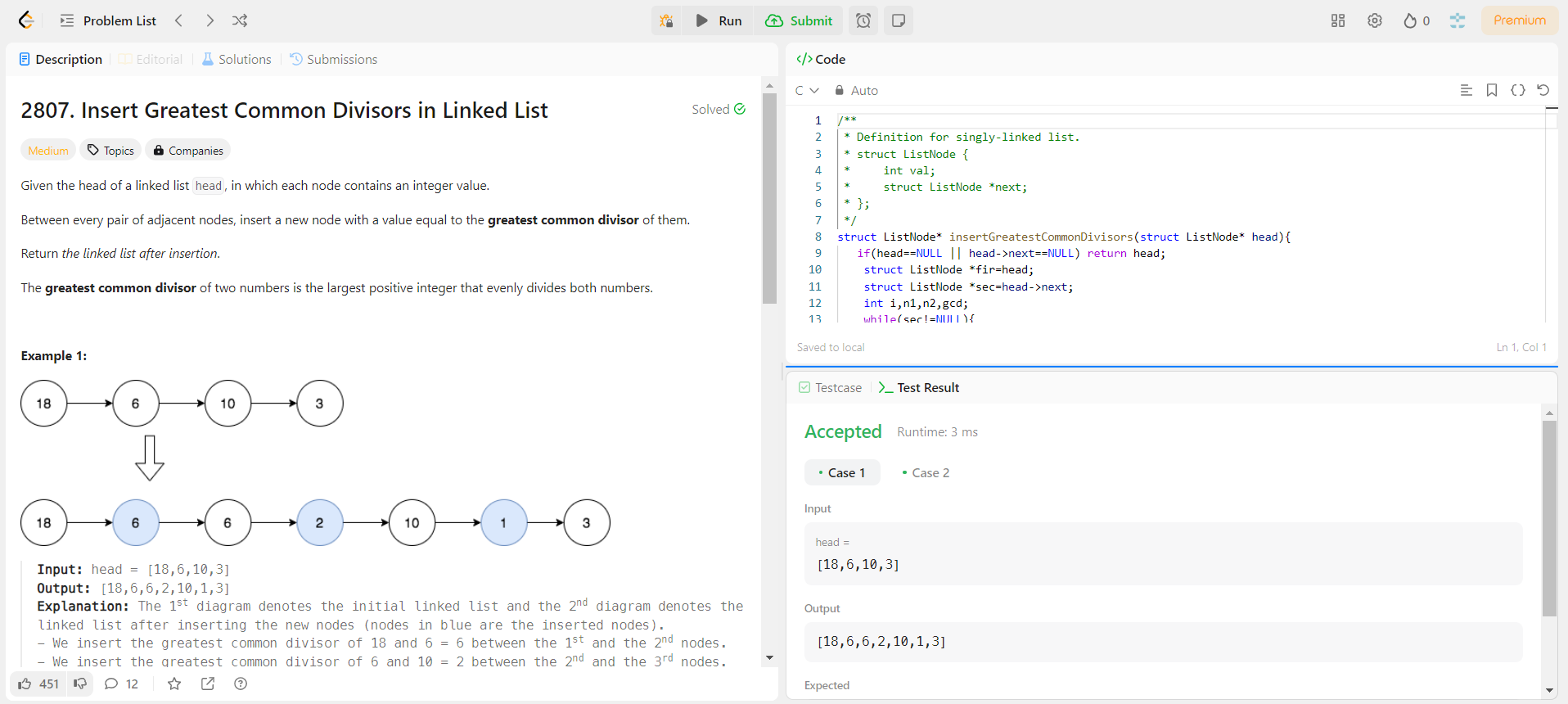


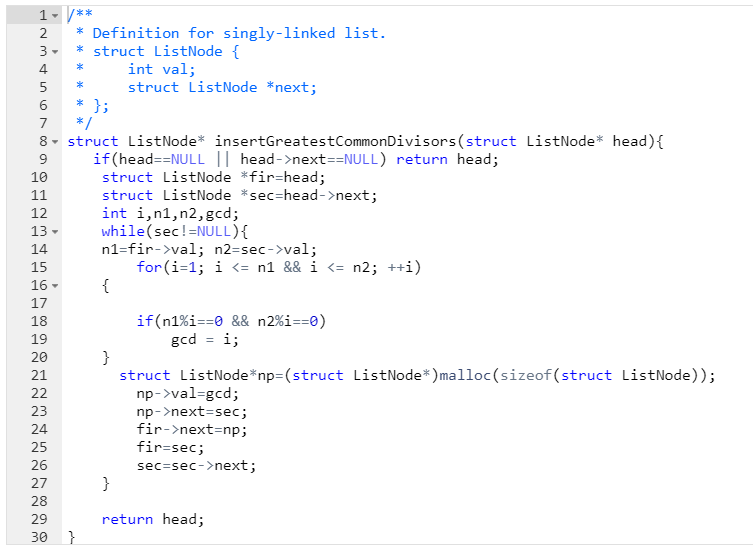
5.Find the Winner of the Circular Game



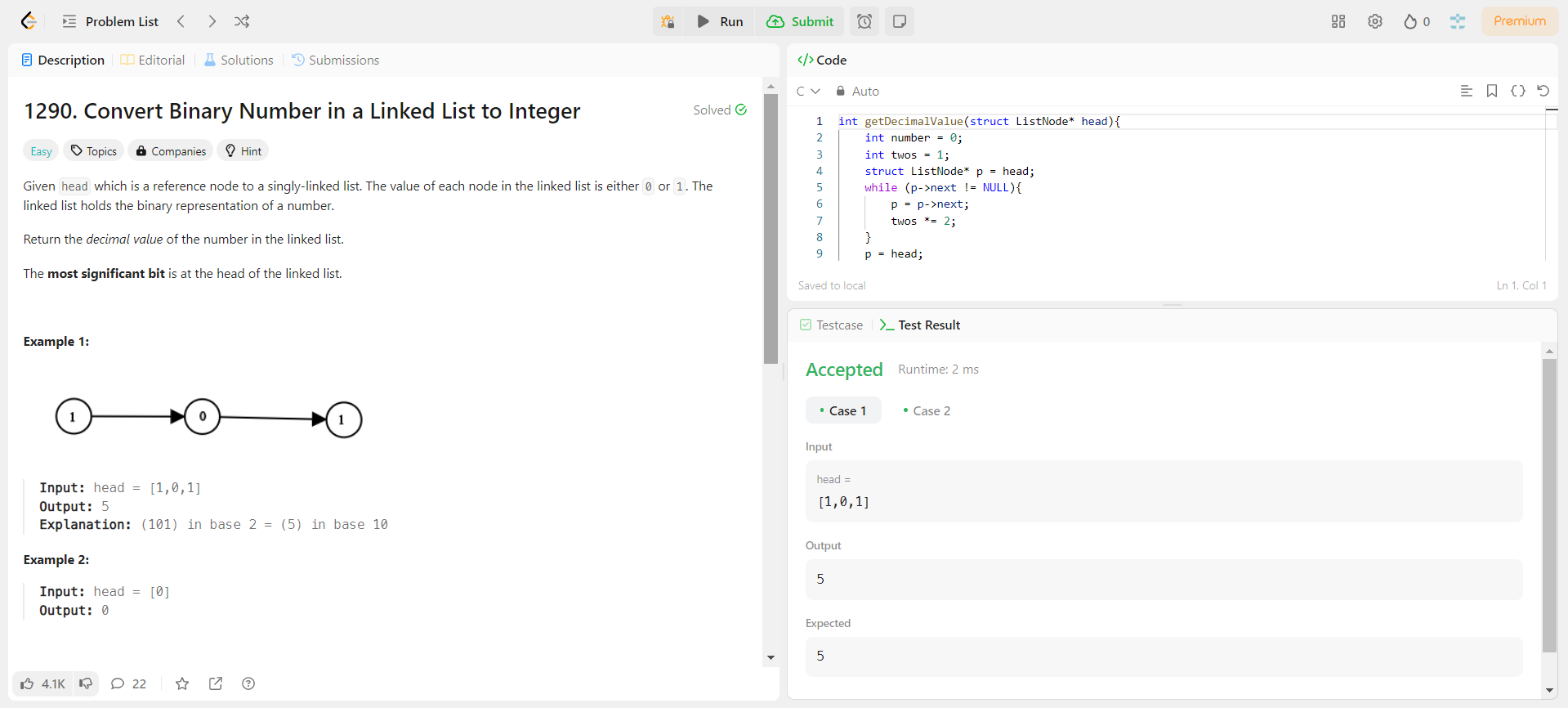


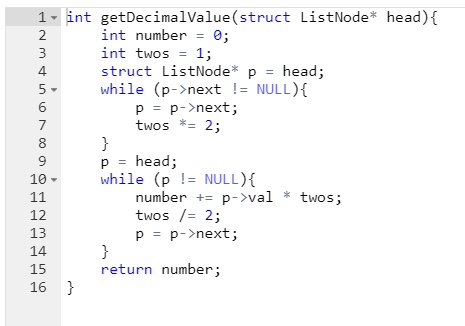
6.Insert Greatest Common Divisors in Linked Lists



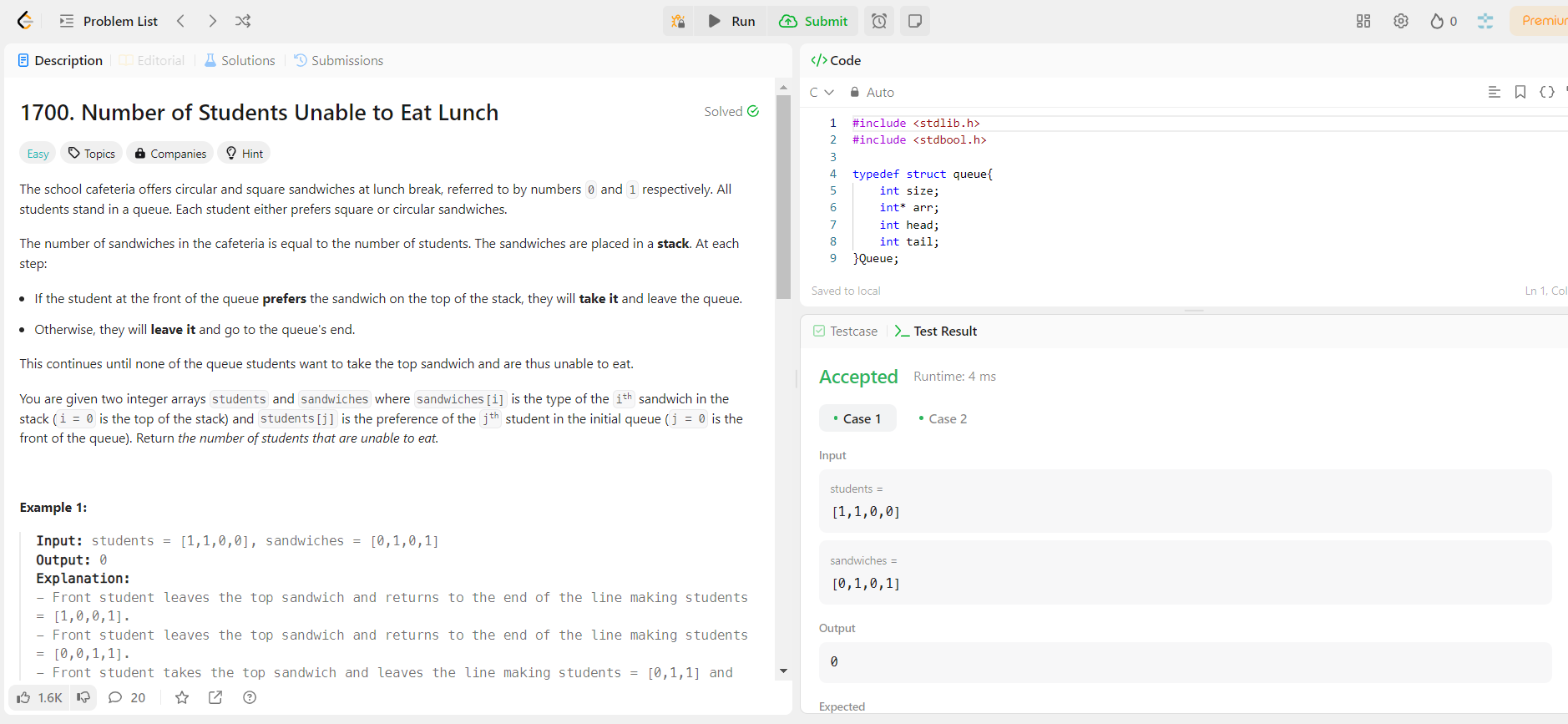


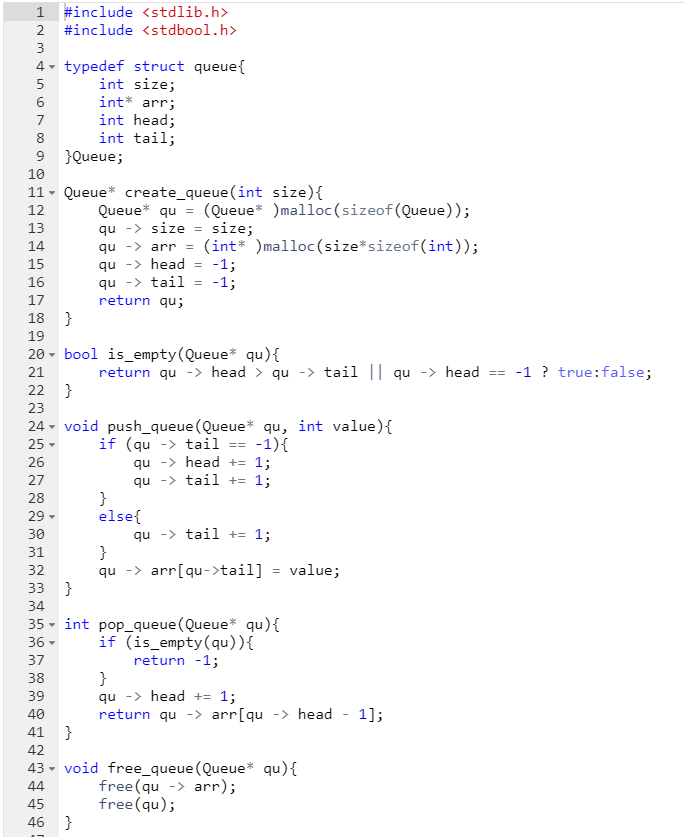
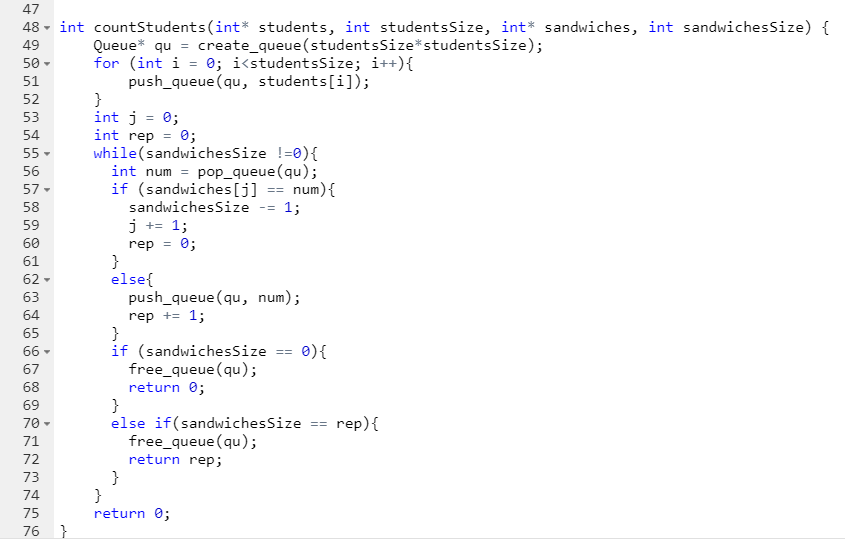
7. Convert Binary Number in a Linked List to Integer



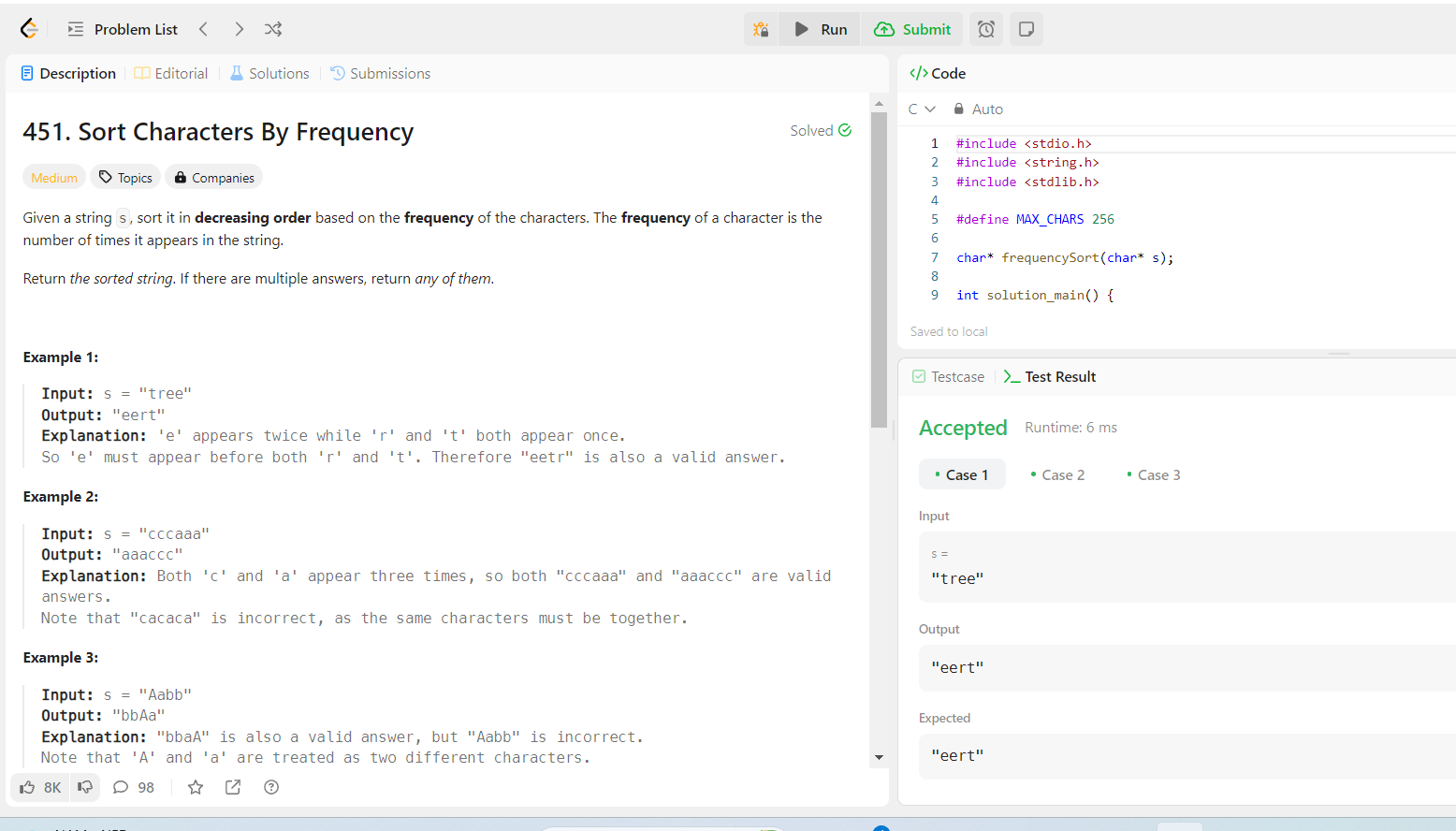


8. Number of Students unable to Eat Lunch



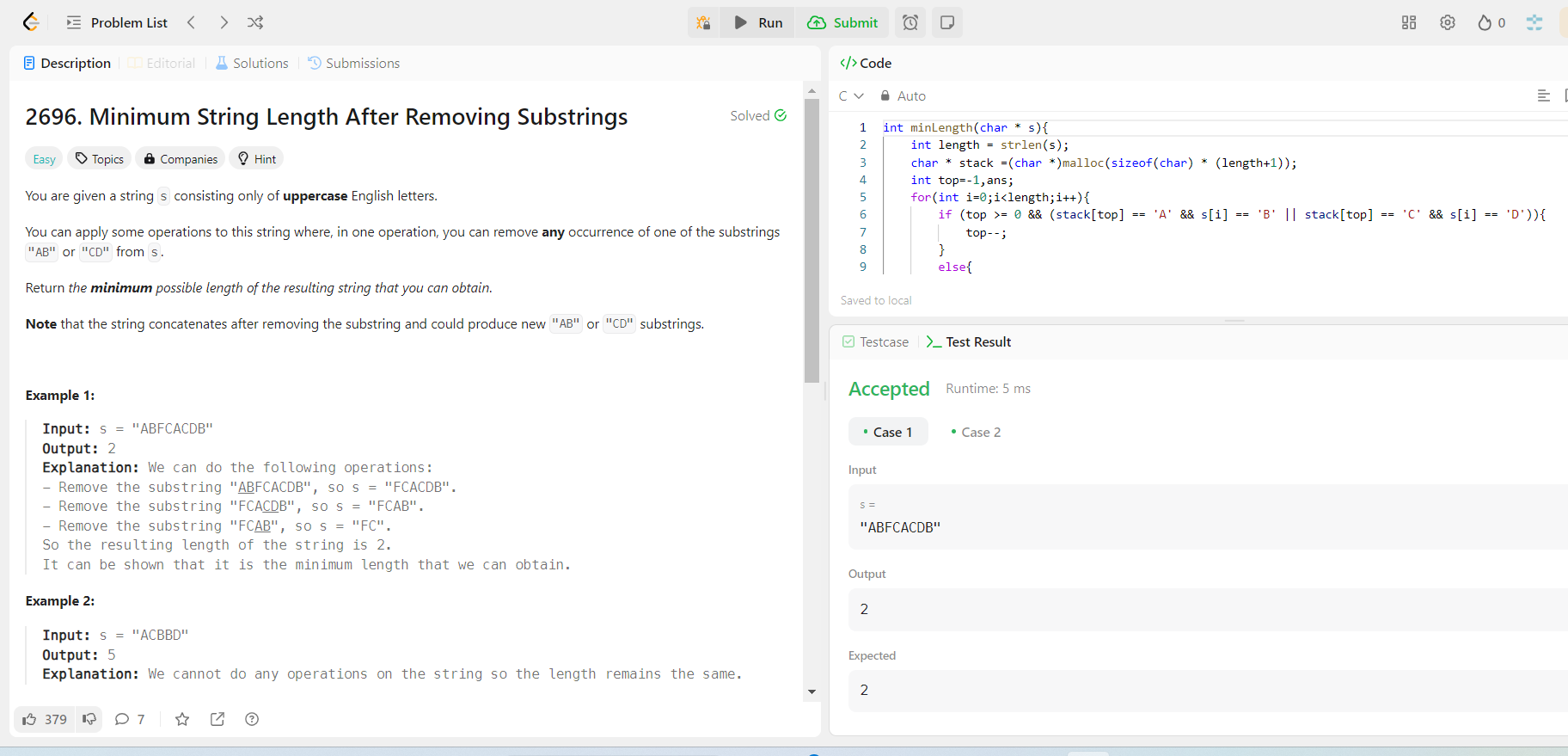
 

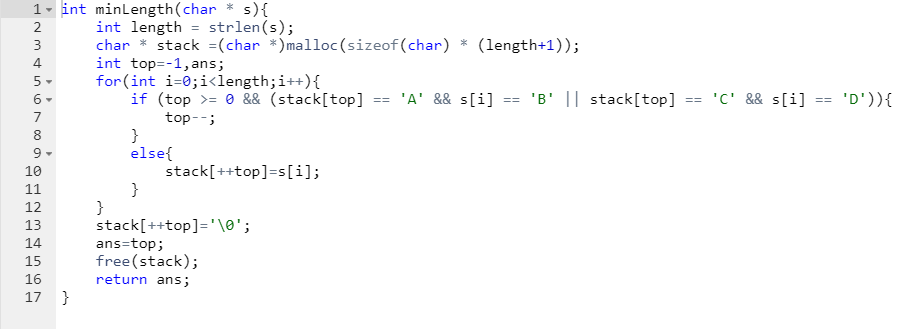
9. Sort Characters By Frequencies





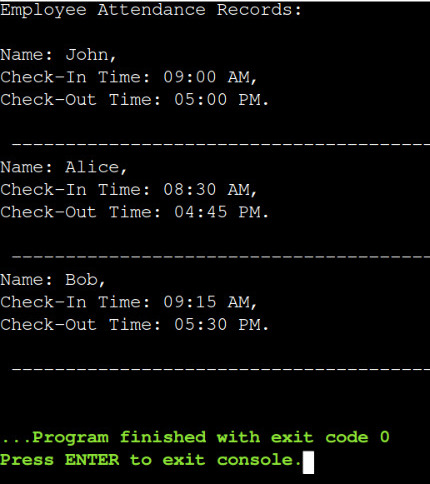
10. Minimum String Length After Removing Substrings







**OUTPUT :-**



**Real Time Solution:-**

To implement employee attendance tracking using a linked list in C:

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

typedef struct EmployeeAttendance {

char name[50];

char checkInTime[20];

char checkOutTime[20];

struct EmployeeAttendance \*next;

} EmployeeAttendance;

void addAttendance(EmployeeAttendance \*\*head, char \*name, char \*checkInTime, char \*checkOutTime) {

EmployeeAttendance \*newRecord = (EmployeeAttendance \*)malloc(sizeof(EmployeeAttendance));

if (newRecord == NULL) {

printf("Memory allocation failed!\n");

return;

}

strcpy(newRecord->name, name);

strcpy(newRecord->checkInTime, checkInTime);

strcpy(newRecord->checkOutTime, checkOutTime);

newRecord->next = NULL;

if (\*head == NULL) {

\*head = newRecord;

} else {

EmployeeAttendance \*temp = \*head;

while (temp->next != NULL) {

temp = temp->next;

} temp->next = newRecord;

} }

void displayAttendance(EmployeeAttendance \*head) {

if (head == NULL) {

printf("No attendance records found.\n");

return;

}

printf("Employee Attendance Records:\n");

while (head != NULL) {

printf("Name: %s,\n Check-In Time: %s\n, Check-Out Time: %s\n", head->name, head->checkInTime, head->checkOutTime);

printf(“\n-------------------------------------------------------------------------\n”);

head = head->next;

}

}

int main() {

EmployeeAttendance \*attendanceRecords = NULL;

addAttendance(&attendanceRecords, "John", "09:00 AM", "05:00 PM");

addAttendance(&attendanceRecords, "Alice", "08:30 AM", "04:45 PM");

addAttendance(&attendanceRecords, "Bob", "09:15 AM", "05:30 PM");

displayAttendance(attendanceRecords);

return 0;

}