Shri Vile Parle Kelavani Mandal's

INSTITUTE OF TECHNOLOGY

DHULE (M.S.)

DEPARMENT OF COMPUTER ENGINEERING

Subject: Competitive Programming Lab (BTCOL606)

Remark

Name : Mohammad Anas Aarif Baig Mirza **Roll No. :** 31

Class : T.Y Comp **Batch : T2** **Division: T**

Signature

Expt. No. :09 **Date :**

Title : Problem 6:Write a Program to implement Shoemaker' s Problem.

Code:

```
// MOHAMMAD_ANAS_31_TY_COMP

#include <bits/stdc++.h>
using namespace std;
// Structure to store appointment details
struct Appointment {
    int start; // Start time in minutes
    int end;   // End time in minutes
};
// Convert time string (hh:mm) to minutes since 00:00
int toMinutes(string time) {
    int hours = stoi(time.substr(0, 2));
    int minutes = stoi(time.substr(3, 2));
    return hours * 60 + minutes;
}
```

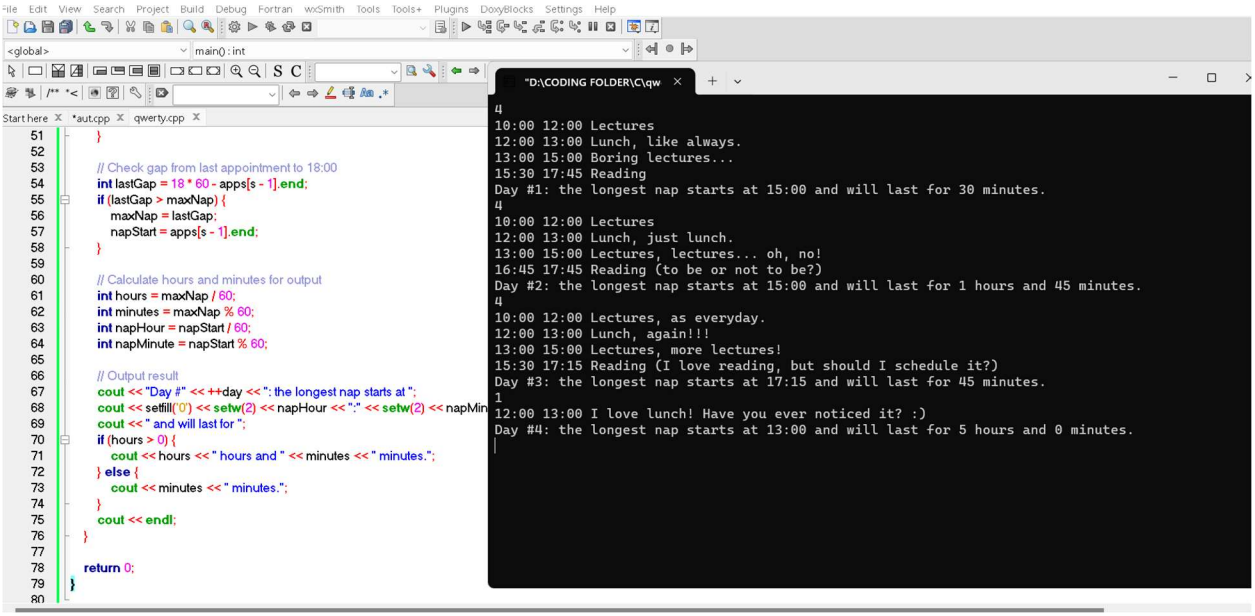
```
int main() {
    string line;
    int day = 0; // Day counter for output
    while (getline(cin, line)) { // Read until EOF
        int s = stoi(line); // Number of appointments
        if (s == 0) break; // In case of empty line or end
        vector<Appointment> apps(s);
        // Read each appointment
        for (int i = 0; i < s; i++) {
            getline(cin, line);
            string time1 = line.substr(0, 5); // Start time
            string time2 = line.substr(6, 5); // End time
            apps[i].start = toMinutes(time1);
            apps[i].end = toMinutes(time2);
        }
        // Sort appointments by start time
        sort(apps.begin(), apps.end(), [](const Appointment& a, const Appointment&
b) {
            return a.start < b.start;
        });

        // Initialize with nap from 10:00 to first appointment
        int maxNap = apps[0].start - 10 * 60; // From 10:00
        int napStart = 10 * 60; // Start at 10:00
        // Check gaps between consecutive appointments
        for (int i = 0; i < s - 1; i++) {
```

```
    int gap = apps[i + 1].start - apps[i].end;
    if (gap > maxNap) {
        maxNap = gap;
        napStart = apps[i].end;
    }
}
// Check gap from last appointment to 18:00
int lastGap = 18 * 60 - apps[s - 1].end;
if (lastGap > maxNap) {
    maxNap = lastGap;
    napStart = apps[s - 1].end;
}
// Calculate hours and minutes for output
int hours = maxNap / 60;
int minutes = maxNap % 60;
int napHour = napStart / 60;
int napMinute = napStart % 60;
// Output result
cout << "Day #" << ++day << ": the longest nap starts at ";
cout << setfill('0') << setw(2) << napHour << ":" << setw(2) << napMinute;
cout << " and will last for ";
if (hours > 0) {
    cout << hours << " hours and " << minutes << " minutes.";
} else {
    cout << minutes << " minutes.";
```

```
        }
        cout << endl;
    }
    return 0;
}
```

Output:



```
51      }
52
53      // Check gap from last appointment to 18:00
54      int lastGap = 18 * 60 - apps[s - 1].end;
55      if (lastGap > maxNap) {
56          maxNap = lastGap;
57          napStart = apps[s - 1].end;
58      }
59
60      // Calculate hours and minutes for output
61      int hours = maxNap / 60;
62      int minutes = maxNap % 60;
63      int napHour = napStart / 60;
64      int napMinute = napStart % 60;
65
66      // Output result
67      cout << "Day #" << ++day << ": the longest nap starts at ";
68      cout << setfill('0') << setw(2) << napHour << ":" << setw(2) << napMin
69      cout << " and will last for ";
70      if (hours > 0) {
71          cout << hours << " hours and " << minutes << " minutes.";
72      } else {
73          cout << minutes << " minutes.";
74      }
75      cout << endl;
76  }
77
78  return 0;
79  }
80
```