Program:

```
#include<bits/stdc++.h>
using namespace std;

struct app
{
    int SH, SM, EH, EM;
};

bool compareApps(app a1, app a2)
{
    int a1SM = a1.SH*60 + a1.SM;
    int a2SM = a2.SH*60 + a2.SM;
    return a1SM < a2SM;
}
int main()
{

    int n;
    char Input[256];
    int counter=0;
    while(1)
    {
        scanf("%d",&n);
        counter++;
        gets(Input);


        vector<app> apps;

        for (int i=0; i<n; i++)
        {
            gets(Input);
            app a;
            a.SH = (Input[0] - '0')*10 + Input[1]- '0';
            a.SM = (Input[3]- '0')*10 + Input[4]- '0';
            a.EH = (Input[6]- '0') * 10 + Input[7]- '0';
            a.EM = (Input[9]- '0') * 10 + Input[10]- '0';
            apps.push_back(a);
        }

        sort(apps.begin(), apps.end(), compareApps);
        int maxNapTime=-1, maxStartH, maxStartM;
        string output;
```

```
        if (apps[0].SH != 10 || apps[0].SM != 0)
        {
            maxNapTime = apps[0].SH*60 + apps[0].SM - 10*60;
            maxStartH = 10;
            maxStartM = 0;

        }
        for (int i=1; i<apps.size(); i++)
        {
            if ((apps[i-1].EH != apps[i].SH || apps[i-1].EM != apps[i].SM) &&
                maxNapTime < apps[i].SH*60 + apps[i].SM - apps[i-
1].EH*60 - apps[i-1].EM)
            {
                maxNapTime = apps[i].SH*60 + apps[i].SM - apps[i-
1].EH*60 - apps[i-1].EM;
                maxStartH = apps[i-1].EH;
                maxStartM = apps[i-1].EM;
            }
        }


        if (apps.back().EH != 18 && maxNapTime < 18*60 - apps.back().EH*60 - apps
.back().EM)
        {
            maxNapTime = 18*60 - apps.back().EH*60 - apps.back().EM;
            maxStartH = apps.back().EH;
            maxStartM = apps.back().EM;

        }

            ostringstream hours;

            if (maxNapTime / 60)
                hours << maxNapTime/60 << " hours and ";

            cout << "Day #" << counter
                << ": the longest nap starts at " << maxStartH/10 << maxStartH%10
 << ':' << maxStartM/10 << maxStartM%10 <<
                " and will last for " << hours.str() << maxNapTime%60 << " minute
s.\n";


    }
    return 0;
```

```
}
```

Output:

```
4
10:00 12:00 Lectures
12:00 13:00 Lunch, like always.
13:00 15:00 Boring lectures...
15:30 17:45 Reading
Day #1: the longest nap starts at 15:00 and will last for 30 minutes.
4
10:00 12:00 Lectures
12:00 13:00 Lunch, just lunch.
13:00 15:00 Lectures, lectures... oh, no!
16:45 17:45 Reading (to be or not to be?)
Day #2: the longest nap starts at 15:00 and will last for 1 hours and 45 minutes.
4
10:00 12:00 Lectures, as everyday.
12:00 13:00 Lunch, again!!!
13:00 15:00 Lectures, more lectures!
15:30 17:15 Reading (I love reading, but should I schedule it?)
Day #3: the longest nap starts at 17:15 and will last for 45 minutes.
1
12:00 13:00 I love lunch! Have you ever noticed it? :)
Day #4: the longest nap starts at 13:00 and will last for 5 hours and 0 minutes.
```