CS2810: OOAIA EndSem Lab Exam

Date: 29th April, 2025
Link: Hackerrank link

Problem Statement

In a realm of extraordinary events ranging from interstellar concerts to enchanted theaters and timeless weddings, Organizers must optimize their schedules for maximum profitability.
Each event type follows distinct financial models, incorporating various costs, revenues, and conditional profit adjustments.

The objective is to schedule non overlapping events to achieve maximum profit.

There are 3 types of events as mentioned below
Types of Events :
1. Concerts
Description: Ethereal performances by star-bound artistes, drawing crowds from across dimensions.
Attributes:

Ticket Price: Price per attendee.
Tickets Sold: Number of attendees.
Artist Fee: Cost to hire the celestial performer.
Logistic Cost: Expenses for interdimensional portals and stage setup.

Profit Secrets:
Revenue: Ticket Revenue = (Ticket Price * Tickets Sold)
Ticket Revenue = Ticket Revenue - 0.18*Ticket Revenue (gst)
Cost: Sum of total cost (artist fees and logistics).
Profit: Revenue - Cost
Net Profit = (profit>2*cost)? 0.7* profit : profit.

2. TheaterShow
Description: Illusionary plays where VIP seats float mid-air, and enchanted popcorn rains from the skies.
Attributes:

Base Ticket Price: Standard seat cost.
Total Seats: Capacity of the theater.
Venue Cost: Fee to maintain the magical arena.
Assume Theater fills with people, i.e Total attendees = Total Seats

Profit Secrets:
Ticket Revenue: 0.25 * Total seats * (2 * Base Ticket Price) + 0.75 * Total Seats * Base Ticket Price.
Ticket Revenue= Ticket Revenue - 0.18 * Ticket Revenue. (gst)
Popcorn Price =150
Popcorn Revenue: 0.25 * Total Seats * Popcorn Price.
Net Profit: Ticket Revenue+ Popcorn Revenue- venue cost.
Note : Total seats attribute is always divisible by 4.

3. Wedding
Description: Timeless unions where guest lists defy mortality—but so do the costs.
Attributes:

Base Amount: Fixed payment from the couple.
Decoration Cost: Expenses for celestial floral arrangements.
Guest Count: Number of attendees.
Venue Cost: Base fee for the timeless hall.

Profit Secrets:
Total Venue Cost: (Guestcount > 200) ? Venue Cost*3 : Venue Cost
Catering Cost: (Guest Count > 100) ? Guest Count * 70 : Guest Count * 100 .
Net Profit: Base amount - Total Venue Cost - decoration Cost - catering costs.

Design Specifications

Class Hierarchy:
1. Event Class:
Attributes: startTime, endTime (integers). (endTime > startTime)
Methods: Pure virtual function calculateProfit().
Overload stream extraction operator(>>) in this class to take input.

2. Derived Classes:
All Derived classes will override calculateProfit() function.
Concert: Uses ticket sales, taxes, artist fees, and logistics to compute profit.
TheaterShow: Calculates profit from VIP/base tickets, popcorn sales, and venue costs.
Wedding: Determines profit using guest-count penalties and catering discounts.

3. EventScheduler Class:
Attributes: Stores a vector of Event objects or Event* pointers.
Methods: Algorithm to Maximize Profit obtained by scheduling Non overlapping events.
Note: If ***i*** and ***j*** are two events and ***starttime[i] < starttime[j]*** then ***i*** and ***j*** are non overlapping If ***starttime[j] >= endtime[i]***

You can define additional classes and methods as needed.
You can use stl functions.
If you are using sort function to **sort events** then you need to have a **custom comparator functor** to compare the event objects.

Design weightage(4 Marks):

1) Class Hierarchy(Event, Concert, TheaterShow, Wedding, EventScheduler) - 1 Marks
2) Overriding calculateProfit() in each derived class - 1 Marks
3) Overloading stream extraction operator(>>) in Event class - 1 Marks
4) Comments or self explanatory function names - 1 Marks

Testcase Distribution:

Brute Force solution - 12/48 test cases will pass
Optimized solution - 42/48 test cases will pass
Most efficient solution - All test cases will pass

Constraints

Time: 0 ≤ Start < End ≤ 1e3.
Events: 1 ≤ N ≤ 2* 1e5.
Profits: Each event yields 1 ≤ profit ≤ 1e4.
All the attributes of all classes are integers and can be stored in **int** data structure

---

Input Format

First Line: N (total number of Events).
Next N Lines:
[Type] [Start] [End] [Attributes...]
Type 1 (Concert): 1 Start End TicketPrice TicketsSold ArtisteFee LogisticCost
Type 2 (TheaterShow): 2 Start End BasePrice TotalSeats VenueCost
Type 3 (Wedding): 3 Start End BaseAmount DecorationCost GuestCount VenueCost

---

Output Format

Print maximum achievable profit up to 2 decimal places as follows
cout << fixed << setprecision(2) << maxprofit << "\n";
Where maxprofit is the calculated answer.
NOTE: No Overlaps, Scheduled events must not overlap in time.

Sample Test case 1:
4
1 778 911 79 104 3140 2607
1 675 695 88 66 3385 465
2 508 675 93 204 26293
3 360 828 16723 4742 82 2466
Output: 2705.98
Explanation:
After calculating profits for all events we have
778 911 990.12
675 695 912.56
508 675 803.3
360 828 1315
We can achieve maximum profit 2705.98 by scheduling events 1st,2nd and 3rd events in the following way.
3 - 508 675 803.3
2 - 675 695 912.56
1 - 778 911 990.12

Sample Test case 2:
5
2 484 842 111 140 15308
3 985 996 27116 9659 55 2981
1 712 813 79 156 5213 2631
1 205 709 83 137 1038 289
2 742 934 68 196 15072
Output : 20513.25
Explanation :
After calculating profits for all events we have
484 842 5870.5
985 996 8976
712 813 2261.68
205 709 5598.05
742 934 5939.2
We can achieve maximum profit 20513.25 by scheduling events 2nd,4th and 5th events in the following way
4 - 205 709 5598.05
5 - 742 934 5939.2
2 - 985 996 8976

ALL THE BEST !😄