Assignment 1 : Hotel Reservation System

Deadline: Sunday January 30 at 23:59
Type: Individual Assignment
Weight: 5%

Marking Scheme:

- Program correctness (70%)
- Program clarity (output format, comments, completeness, readability) (10%)
- Test cases to be comprehensive enough to cover your program to test if it is bug free (20%)

Q1. (50 marks) Hotel "Cplusplus" has 20 rooms (room numbers are from 1 to 20). You are supposed to design and implement the Hotel Reservation system in C++ for the March break holidays. The reservations take place between March 1 to March 8 (7 nights).

Define the following classes to manage the hotel "Cplusplus" reservation system. Include one or more constructors, destructor and the necessary set and get functions for each class. Also, provide a print function for each class, which outputs all the data members of that class.

A) Define a class ***`Date`*** that has the following data members: ***`Month`***, ***`day`*** and ***`year`***;

B) Define a class ***`information`*** with the following data members: ***`first name`*** (array of characters), ***`last name`*** (array of characters), and ***`date of birth`*** (from part A)

C) Define a class ***`Guests`*** with the following data members:

- *Check in date (from part A)*
- *Check out data (from part A)*
- *An array that keeps the information of all the guests in a room (from part B, maximum capacity of the array is 4)*
- *Room number (1 to 20) (assume maximum number of reserved room is 1)*

D) Define a class ***`Guests_Res_Request`*** with the following data members:

- *one* ***`Guests`*** *object from part C*
- *An integer counter that generates reservation IDs (unique)*
- *An integer variable as a reservation ID. (for simplicity you can assume the reservation ID starts from 1 and it is incremented by one whenever a* ***`Gusts_Res_Request`*** *object is created.*
- *Number of nights*

E) Define a class ***`Reservation_Manager`*** with the following data members:

- Variable maximum number of nights (**`max_no_of_nights`**) for the reservation (should be equal to 7 (March 1 to March 8) for this case
- Variable number of rooms (***`no_of_rooms`***) which shows the total number of available rooms (it should be 20)
- An array of pointers to the ***`Guests_Res_Request`*** objects (Guest_Res_request *arr)
- A two-dimensional (***`max_no_of_nights * no_of_rooms)`*** array of integers which is filled with the reservation ID when a reservation request is processes. (The array entries are initialized to 0 at the beginning).

The following table shows an example of this two-dimensional array:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Mar 1		1											3				6			
Mar 2		1							2				4				6			
Mar 3	10	1							2				4		5		6	7		
Mar 4	10								2						5		6	8		
Mar 5	10								2						5			9		
Mar 6									2						5			9		
Mar 7															5			9		

- All the empty spots are filled by 0

At least the following additional functions should be provided:

- A Member function that processes a ***`Gusts_Res_Request`*** that is received as an input parameter. The reservation is successful, if the a room (with a specific number) is available for all the duration of the guests stay (number of nights). If the reservation is successful, this function returns the reservation ID and the new reservation ID is added to the corresponding elements in the two dimensional array. Also, the ***`Gusts_Res_Request`*** object should be added to the array of requests.
  If the reservation is not successful, it should be deleted from the array of request objects. If the reservation is not successful, the ***`Gusts_Res_Request`*** object should be deleted from the array of requests.
- A member function that receives a reservation ID as a parameter and outputs the details of the reservation.
- A member function that receives a reservation ID and cancels the reservation request.

Key Considerations for the assignment:

- You must enforce encapsulation by keeping all data members private.
- ***Guests_Res_Request*** objects should be created using dynamic memory allocation and the objects that are not needed anymore should be deleted.
- In the implemented member functions the relevant data members should be updated.
- You need to make sure that your classes are well defined using the various concepts seen in the class including constant member functions, static members, inline functions, etc.