Computer Science programming team task – The Kashyyyk Hotel

You are to write a user-friendly self-service program for a small boutique hotel reception. The program should work in a loop so any guest can access the system during their stay.

You are to work in a group to **plan, design** and code the system described below. You should divide up the program so that each person in your team writes C subroutines that will be put together into one overall program.

As you are working in Reading Week, you need to make sure that you have set up a chat group in Teams so you can share work/ problems/ help each other (and possibly invite one of us to help too).

The main program should give the user the following four choices:

Check in:

- Your program should read in and store:
  - The **surname** of the main user and their **date of birth** (DD/MM/YY)
  - The **number of guests** staying in the hotel – including how many are adults and how many are children (16 or below)
  - The **board type** – FB (full board), HB (half board) or BB (Bed & breakfast).
  - The **length** of their stay (in whole days)
  - Whether they want a **daily wake-up call** or not
- It should give them a choice of the *available* rooms and display the prices of each room so that the user can make their choice.
  - There are 6 rooms in total. Bookings may consist of more than one room.
  - Rooms 1, 2 and 3 can sleep up to 4 people. Rooms 4, 5 and 6 can sleep up to 2 people.
  - Room rates **per room, per day**:
    - Room 1 & 2 = £100, Room 3 = £85, Room 4 & 5 = £75 and Room 6 = £50
  - Board rates are **per person, per day**:
    - Full board = £20, half board = £15, B&B = £5
- Once they have completed check in and been assigned accommodation, the booking is given a BookingID. This ID should be made up of their surname plus a random number. This ID must be used to book meals in the dining room.
- All of the above information will be used to create the hotel invoice (bill) at the end of their stay.

Book a dinner table:

There are 3 dining tables: Endor, Naboo and Tatooine. Each seat up to 4 people.

Meals are served at 2 sittings: 7pm and 9pm, every day of the week.

Your program should:

- Ask for the guest's BookingID and, if it exists in the system (i.e. they are a current guest), it should check:
  - Whether they are staying FB or HB. If not, they cannot book a table.
  - Then, it must check if there is a free table that day for the number of people in that party (number of people must be looked up from the booking data).
- Display options showing the table names and the times that have space OR output an unavailable message if no options exist for that day.
- It should then ask the user to input which table & time they want to book.
- It should give the user a message saying their table is booked and update the tables available in that area for that sitting time & day.

Check out:

- When the customer chooses to check out, your program should calculate and display their final bill.

- Your program will need to use the customer data to calculate:
  - The total cost of the rooms used
  - The total cost of all of the party's board (meals)
  - If the user had daily wake-up calls it should add a one off cost of £5.00 to the bill
  - If the main user is over 65 then they get a 10% discount on the room rate
  - Guests who are children then get 50% off their board rate total (For that person/ persons only NOT every person in the group)

- The bill should be well presented and show:
  - The user's BookingID
  - Each of the sub-total amounts from above (where there is a >0 amount)
  - An overall bill total for the stay

- Once someone has checked out, their accommodation should be available to use for other guests

Quit program:

This isn't really quitting at all – it should allow the user to end their use of the interface and return them to the main menu **BUT not end the whole program** so data is not lost (So you can test all of the above properly)

TIPS

Plan carefully – decide who is doing what parts and what data you will need to share.

Create a data dictionary of global shared variables, structure diagram, pseudocode/ flowchart plans of the algorithms BEFORE you begin.

Use good programming techniques such as indentation of code and use of comments. This will help when you put your programs together.

You **MUST NOT** use 3D arrays or external files – all data must be stored within the program.

This is going to make your heads hurt, do not be put off, there is nothing in this program that you haven't done before it's just bigger and needs A LOT more thought.

WHAT YOU NEED TO HAND IN AS EVIDENCE:

1. Your **planning** diagram(s)
2. **Flowcharts/ pseudo** of the main algorithms
3. A completed, annotated **program** ready to allow others to test (with simple instructions if needed)
4. A **test plan** for beta testing the programs (this must make sense for all programs, not just yours!)

Test Description	Test Data	Test Type	Expected	Actual
e.g. Check meal booking – cannot book a table if staying B&B	Enter details and choose **B&B rate**, then try to book a table for dinner.	**Valid** Invalid Extreme	Should show error msg and not display any tables or time or let a table be booked.	