|  |  |
| --- | --- |
| **Assignment Case** |  |
| COMP6048 | COMP6048001 | COMP6048016 | COMP6048049  Data Structures |
| **Computer Science** | **Example Case** |
| ***Valid on*** *-* | **Revision 00** |

## Soal

*Case*

**Hotel GrAnsylvania**

**Hotel GrAnsylvania** is one of the largest hotels in Transylvania, Romania which is always have the busiest time serving their guests. The owner wants a new development in term of managing hotel bookings. He then asked you, an avid programmer to build a program using C language and **hash table** with **chaining method**. The program will be created based on the following requirements:

* The program will consists of **4 menus**:

1. **Booking Hotel**
2. **View Bookings**
3. **Delete Bookings**
4. **Exit**

Background pattern

Description automatically generated with low confidence

***Figure 1. Main Menu***

* If user choose **menu 1** (“**Booking Hotel**”), the program will:
* Prompt user to input **full name**. Validate the input must be **between** **3** and **30 characters** (**inclusive**).
* Prompt user to input **phone number**. Validate the input must **begin with** “**+62**”, has **1 space** at minimum, and must be **11 characters long** (exclude “+62” and space).
* Prompt user to input **age**. Validate the input must be **more than equals to** **18**.
* Prompt user to input **room type**. Validate the input must be **between** “**Regular**”, “**Deluxe**”, and “**Suite**” (**case sensitive**).
* Prompt user to input **stay duration**. Validate the input must be **between 0** and **30** (**inclusive**).

Text

Description automatically generated

***Figure 2. Insert Booking Input***

* **Generate booking id** based on the following formula.

|  |
| --- |
| **RRXXX**  **RR** is **first 2 character** from **room type** in **uppercase** format.  **X** is **random number** between 0 – 9.  **Example**: RE001, DE005, SU010 |

* **Store** new booking data to the **next item** of **the last item** of **chaining hash table** with size **100** based on following key which could be calculated as follows

|  |
| --- |
| **Key = X mod Y**  **X** is **sum of last 3 digits** of the **booking id** **minus 1**.  **Y** is **size of hash table**.  **Example**:  **Booking ID** : DE187  **X** : 1 + 8 + 7 - 1 = 15  **Y**  : 100  **Key** : 15 % 100  Therefore, **key** will be **15**. |

* **Display** inserted booking data.

Diagram

Description automatically generated with medium confidence

***Figure 3. Success Insert Information***

* If user choose **menu 2** (“**View Bookings**”), the program will:
* Check data from hash table. If there are **no booking**, **display** following message

Background pattern

Description automatically generated with low confidence

***Figure 4. No Booking Message***

* Otherwise, **display** all **booking data**.

Diagram

Description automatically generated with medium confidence

***Figure 5. View Booking***

* If user choose **menu 3** (“**Delete Bookings**”), the program will :
* Check data from hash table. If there are **no booking**, **display** following message and **redirect** to main menu.

Background pattern

Description automatically generated with low confidence

***Figure 6. No Booking Message***

* **Otherwise**,
* **Display** all booking data.
* Prompt user to input **booking id**.



***Figure 7. Delete ID Input***

* **Search** for booking data. If data **not exists**, **display** following message and **redirect** to **main menu**.

A screenshot of a computer

Description automatically generated with low confidence

***Figure 8. No Data Found***

* **Otherwise**, **remove** data from hash table, **display** following message, and **redirect** to main menu.

A screenshot of a computer

Description automatically generated with low confidence

***Figure 9. Success Delete Message***

* If user choose **menu 4** (“**Exit**”), the program will **close**.