| Practicum Case | |
|---|---|
| COMP6048 \| COMP6048001 \| COMP6048016 \| COMP6048049<br>Data Structures | |
| **Computer Science** | **O221-COMP6048-AM01-04** |
| ***Valid on*** *Even Semester Year 2021/2022* | **Revision 00** |

**Learning Outcomes**

- LO1 – Explain the concept of data structures and its usage in computer science
- LO2 – Illustrate any learned data structures and its usage in application
- LO3 – Apply data structures using C

**Topic**

- Session 4 – Hash Table

**Sub Topics**

- Push
- Pop
- Search

**Soal**
*Case*

# Bluejack Library

**Bluejack Library** is one of the popular libraries in the town. This library has more than 50.000 books. Sadly the senior librarian wants to resign and he doesn't have time to teach the new librarian. **Bluejack Library** needs a simple program to help the new librarian so he can easily manage and search books. **Bluejack Library** hires you as a programmer to help them create a program that can help a new librarian find and manage books in this library easily using a **C programming language** and **hashtable data structure**. The criteria of the program are:

- The program consists of **4 menus**, there are:
  1. **View Book**
  2. **Insert Book**
  3. **Remove Book**
  4. **Exit**

```
Bluejack Library
================
1. View Book
2. Insert Book
3. Remove Book
4. Exit
>>
```

**Figure 1. Main Menu**

- If the user chooses **View Book** (**Menu 1**), then:
  - If there is **no book**, then show "**There is no book(s) !**" message.

```
There is no book(s) !

Press Enter to continue ...
```

**Figure 2. There is No Book Message(View)**

  - **Otherwise**, the program will **show all the book** data.

```
-------------------------------------------------------------------------------------------------------------------
| Book ID                | Book Title                                | Book Author          | ISBN          | Page Number |
-------------------------------------------------------------------------------------------------------------------
| B00001-9780321344755-DM | Don't Make Me Think                      | Mr. Krug S.          | 9780321344755 | 106         |
-------------------------------------------------------------------------------------------------------------------
| B00002-9780735619937-AM | Agile Project Management With Scrum      | Mr. Ken Schwaber     | 9780735619937 | 192         |
-------------------------------------------------------------------------------------------------------------------

Press Enter to continue ...
```

**Figure 3. Show All Book Data**

- If the user chooses **Insert Book** (**Menu 2**), then:
  - The program will ask user to **input** the following data:
    - **Book Title**
      - Validate Book Title must be **between 5 and 50 characters**.
      - Validate Book Title must be **unique**.
    - **Book Author**
      - Validate Book Author must **start with** "**Mr.** " or "**Mrs.** " and its length must be **between 3 and 25 characters**.
    - **ISBN**
      - Validate ISBN must be **numeric** and its length must be **between 10 and 13 digits**.
    - **Page Number**
      - Validate Page Number must be **at least 16**.
  - After that, the program will generate a **unique id** for the inputted book data. The format will follow the following formula.

| [B**XXXXX**]-[ISBN]-[**A**][**T**]<br>**X** is the **last inserted Book Id** number **increased by 1**<br>**A** is the **first character** of **Book Author** in **uppercase** format<br>**T** is the **first character** of **Book Title** in **uppercase** format<br><br>**Example**: B00001-0123423123-JH, B00002-0123412341-AZ |
|---|

  - Then, the program will **store the inputted new book data** to the **next item** of the **last item** of the **chaining hash table** with **size 1007** using the following **hash function**.

| **KEY** = **SUM** % **SIZE**<br>**KEY** : the **hash table index** that will store the data<br>**SUM** : the **sum of the ascii** from **Book Id**<br>**SIZE** : the **size** of the hash table<br><br>**Example**:<br>Book Id : B00001-0123423123-JH<br>SUM : 1044<br>SIZE : 1007<br>KEY : 1044 % 1007 = 37<br>Then, the book data will be **stored at index 37** of hash table. |
|---|

```
Input book title[5-50][unique]: Don't Make Me Think
The book title is already exists !
Input book title[5-50][unique]: Agile Project Management With Scrum
Input author name[3-25][Mr. |Mrs. ]: Ken Schwaber
Input author name[3-25][Mr. |Mrs. ]: Mr. Ken Schwaber
Input ISBN[10-13][numeric]: test123123
Input ISBN[10-13][numeric]: 9780735619937
Input page number[>= 16]: 15
Input page number[>= 16]: 192

Insert success !
```

**Figure 4. Insert Menu**

- If user chooses **Remove Book** (**Menu 3**), then:
  - The program will ask user to input a **Book ID**
    - Validate if the inputted Book ID **doesn't exists** then show "**Book not found**".

```
Input book id to delete: B00001-0123423123-JH

Book not found !
```

**Figure 5. The Inputted Book ID Doesn't Exists**

    - Otherwise, **show the book data** and ask the **user for confirmation**. Validate input must be **either** "**y**" or "**n**".
  - If user **chooses** "**y**", then **delete** the data.
  - Otherwise, if user **chooses** "**n**", then **return** back to main menu.

```
Input book id to delete: B00002-9780735619937-AM


Book Id: B00002-9780735619937-AM
Book Title: Agile Project Management With Scrum
Book Author: Mr. Ken Schwaber
Book ISBN: 9780735619937
Page Number: 192
Are you sure [y|n]? Y
Are you sure [y|n]? y

Delete success !
```

**Figure 6. Confirmation To Delete**

- If user chooses **Exit** (**Menu 4**), then **terminate** the program.

**Please run the EXE file to see the sample program**