Soal Praktikum *Practicum Case*	BINUS UNIVERSITY
COMP6362 Data Structures	
Teknik Informatika *Computer Science*	**CS-COMP6362-Var03.3**
Periode Berlaku Mulai Semester Genap 2020/2021 ***Valid on*** *Even Semester Year* 2020/2021	**Revisi 00** *Revision 00*

Learning Outcomes
- Demonstrate how to create any learned data structure
- Analyze the usage of data structure in application

Topic
- Session 09 - Binary Search Tree

Sub Topics
- Binary Search Tree Implementation
- Insert a node to Binary Search Tree
- Search and delete a node in Binary Search Tree
- Binary Search Tree Traversal

Soal
Case

Mr. Goku is an owner of a company named **Mamoru**. **Mamoru** is a developing company, so it has a lot of customer there. He wants to make the task of managing the customer not too complicated. He asks you as a skillful programmer to make a program using the **Binary Search Tree concept**. Here are the descriptions of the program:

- Program consists of 5 menus:
  1. Add a new data
  2. Update a certain data
  3. InOrder, PreOrder, PostOrder by Customer ID
  4. Delete a certain data
  5. Exit

- If user chooses **Add a new data**, then:
  - Ask user to input **customer ID**.
    Validate that the **ID** must be in this format **C[1-9][0-9]**.
  - Ask user to input **customer name**.
    Validate that the length of the **name** must be **between 3 and 20 characters**.

- Ask user to input **total years of loyality**.
  Validate that this must be **between 0 and 100**.
- Maximum tree height is 3.
  If height is already at maximum, show the message **"--- Maximum Tree Height is 3 ---"**
- If data has been successfully inputted, show the message **"--- Add New Data Success ---"**

• If user chooses **Update a certain data**, then:

- If there is no data in the tree, show the message **"--- There is no data in the tree ---"**
- Ask user to input **customer ID**.
  Validate that the **ID** must be in this format **C[1-9][0-9]**.
- If ID cannot be found, show the message **"-- Customer ID is not found –"**
- If ID is found, show customer ID, name, and total years of loyality.
  >> Then, ask user to update the **name** by inputing the **customer name**.
  Validate that the length of the **name** must be **between 3 and 20 characters**.

  >> Ask user to update **total years of loyality** by inputing **total years of loyality**.
  Validate that this must be **between 0 and 100**.
- If data has been successfully inputted, show the message **"--- Update Data Success ---"**

• If user chooses **Inorder, Preorder, Postorder by Customer ID**, then:

- If there is no data in the tree, show the message **"--- There is no data in the tree ---"**
- If data is in the tree, show the **customer ID, name, and total years of loyality** in in-order, pre-order, and post-order traversal of **customer ID**.

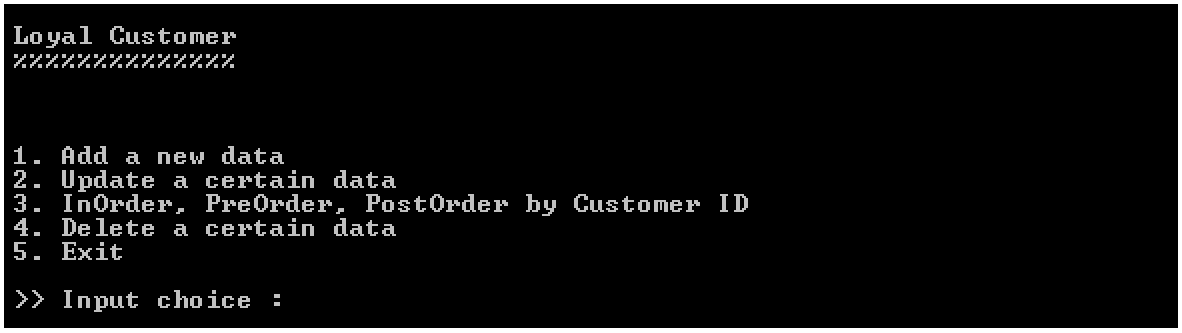
• If user chooses **Delete a certain data**, then:

- If there is no data in the tree, show the message **"--- There is no data in the tree ---"**
- Ask user to input **customer ID**.
  Validate that the **ID** must be in this format **C[1-9][0-9]**.
- If ID cannot be found, show the message **"-- Customer ID is not found –"**
- If ID is found, show the ID, name, and total years of loyality is deleted and show the message **"--- Delete data success ---"**

• If user chooses **Exit**, then:

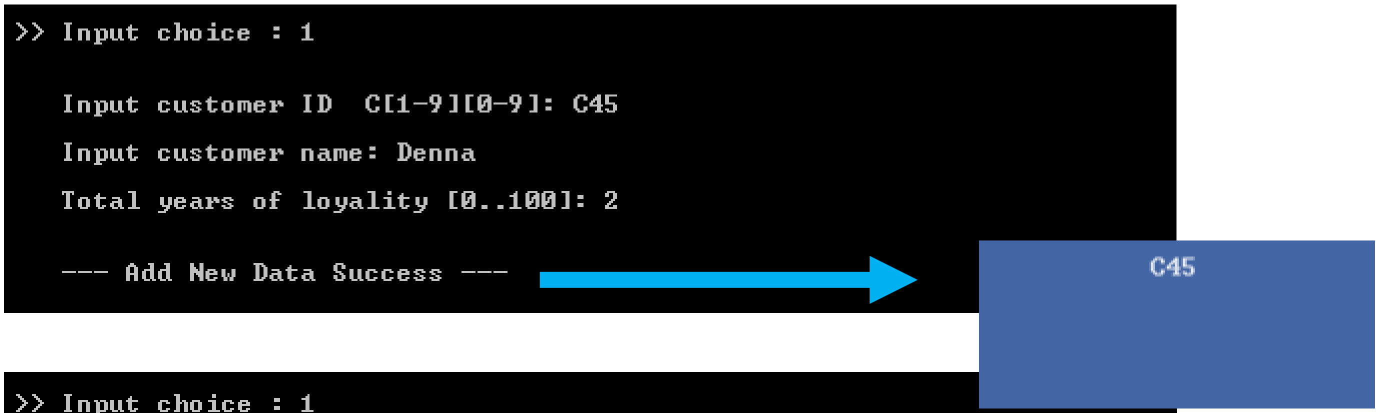
- Delete all data in the linked list.
- Program ends.

Print screen of main menu

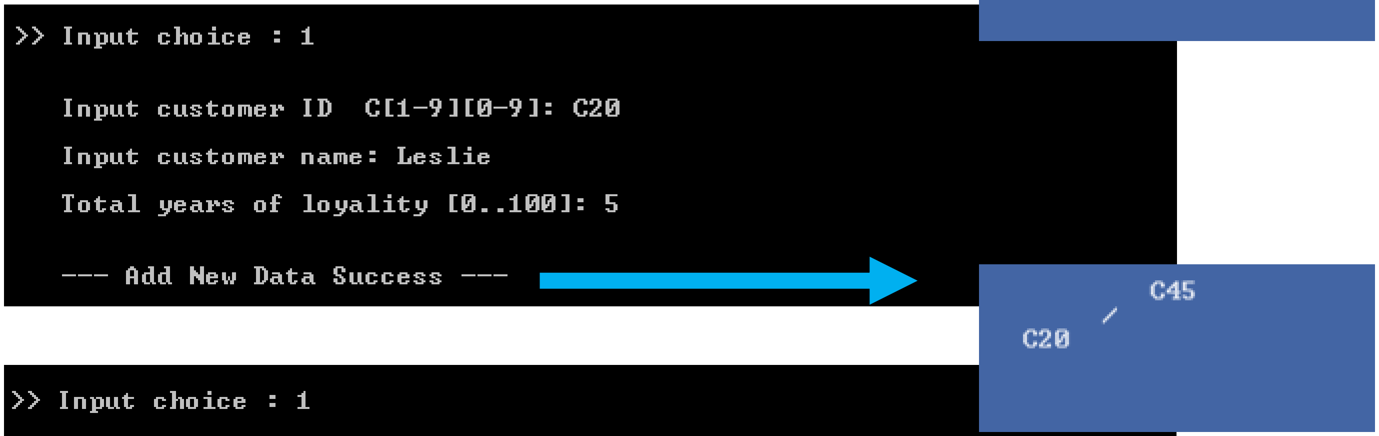


```
Loyal Customer
%%%%%%%%%%%%%%


1. Add a new data
2. Update a certain data
3. InOrder, PreOrder, PostOrder by Customer ID
4. Delete a certain data
5. Exit

>> Input choice :
```

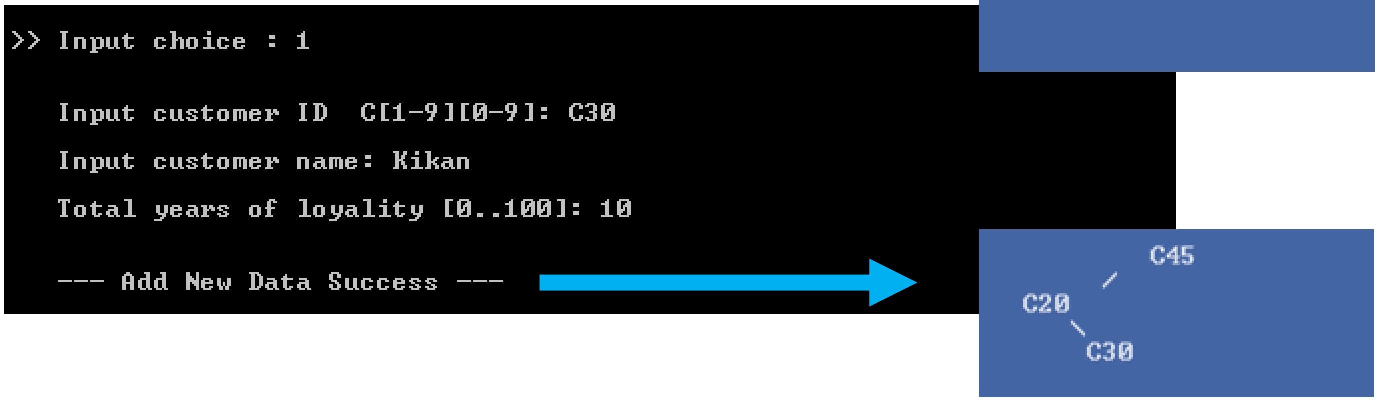
Print screen of Add a new data (Menu '1')



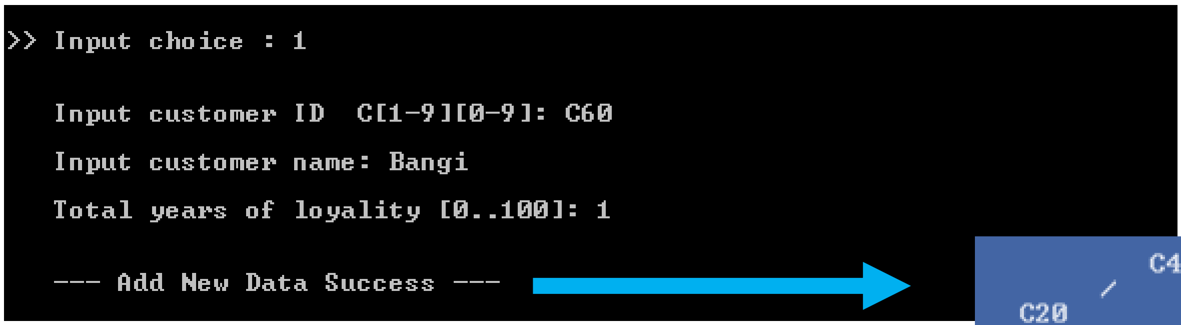
```
>> Input choice : 1

   Input customer ID  C[1-9][0-9]: C45

   Input customer name: Denna

   Total years of loyality [0..100]: 2

     --- Add New Data Success ---
```



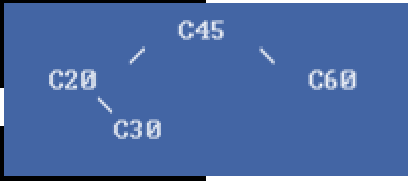
```
>> Input choice : 1

   Input customer ID  C[1-9][0-9]: C20

   Input customer name: Leslie

   Total years of loyality [0..100]: 5

     --- Add New Data Success ---
```

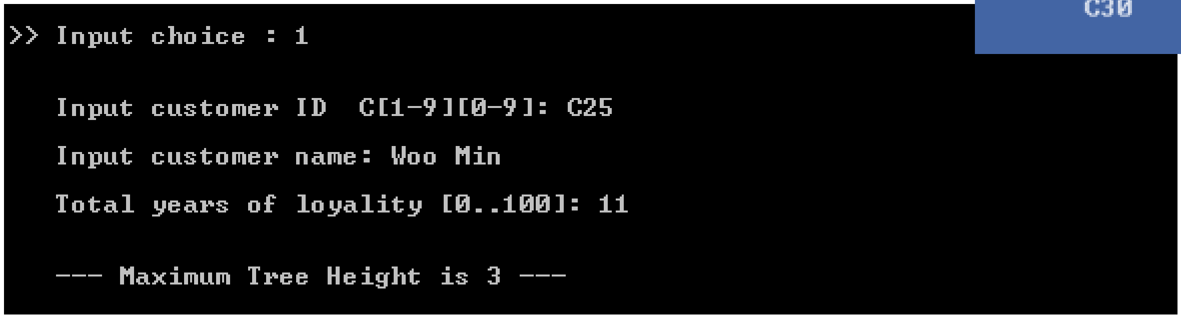


```
>> Input choice : 1

   Input customer ID  C[1-9][0-9]: C30

   Input customer name: Kikan

   Total years of loyality [0..100]: 10

     --- Add New Data Success ---
```



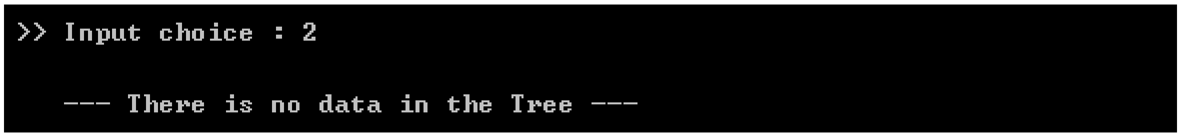
```
>> Input choice : 1

   Input customer ID  C[1-9][0-9]: C60
   Input customer name: Bangi
   Total years of loyality [0..100]: 1

   --- Add New Data Success ---
```





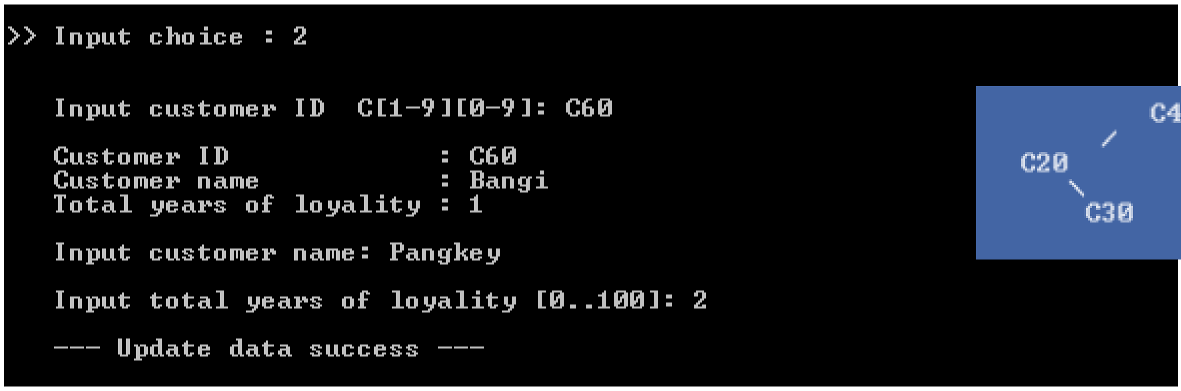
```
>> Input choice : 1

   Input customer ID  C[1-9][0-9]: C25
   Input customer name: Woo Min
   Total years of loyality [0..100]: 11

   --- Maximum Tree Height is 3 ---
```

Print screen of Update a new data (Menu '2') when the tree was still empty

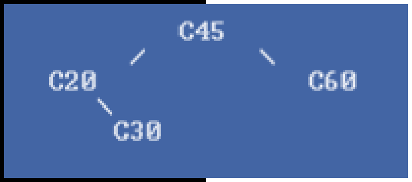


```
>> Input choice : 2

   --- There is no data in the Tree ---
```

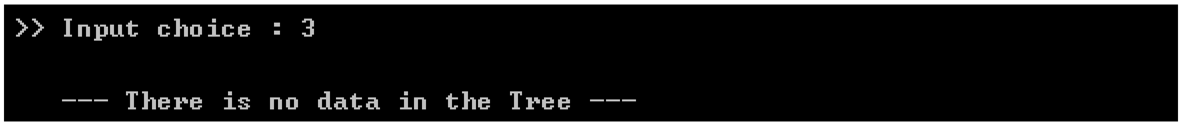
Print screen of Update a new data (Menu '2') when the tree is not empty



```
>> Input choice : 2

   Input customer ID  C[1-9][0-9]: C60

   Customer ID              : C60
   Customer name            : Bangi
   Total years of loyality  : 1

   Input customer name: Pangkey

   Input total years of loyality [0..100]: 2

   --- Update data success ---
```

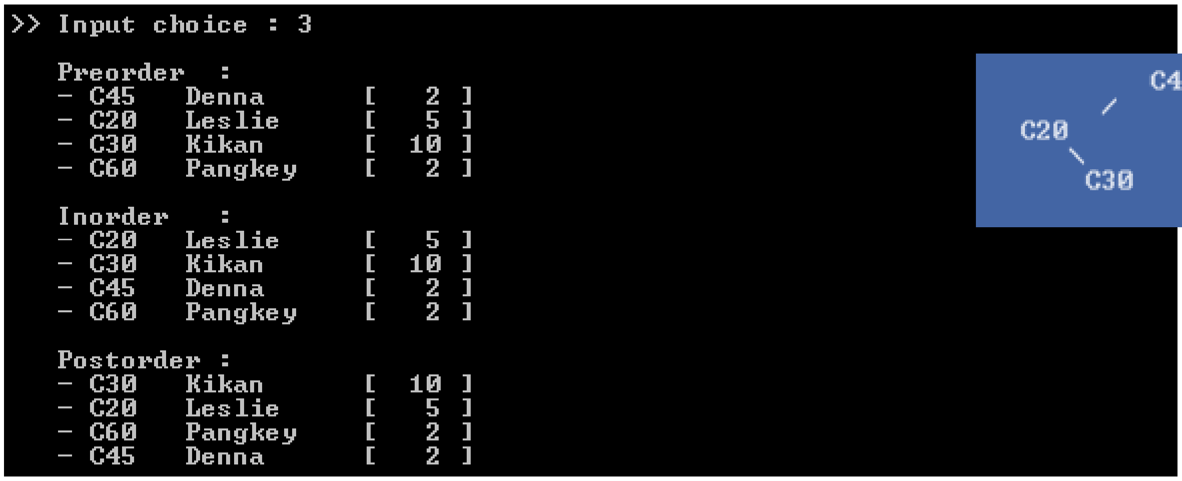


Print screen of Inorder, Preorder, Postorder menu (Menu '3') when the tree was still empty

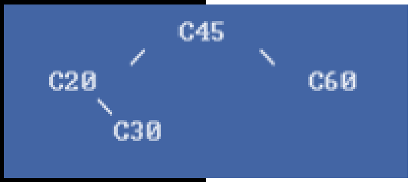


```
>> Input choice : 3

   --- There is no data in the Tree ---
```

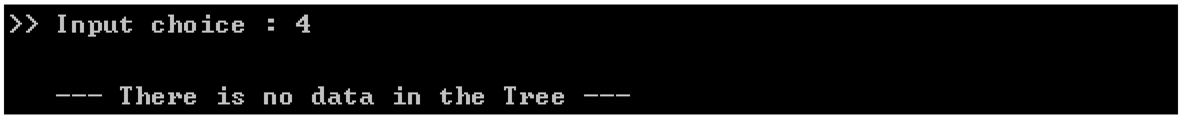
Print screen of Inorder, Preorder, Postorder menu (Menu '3') when the tree is not empty



```
>> Input choice : 3

   Preorder   :
   - C45    Denna       [   2 ]
   - C20    Leslie      [   5 ]
   - C30    Kikan       [  10 ]
   - C60    Pangkey     [   2 ]

   Inorder    :
   - C20    Leslie      [   5 ]
   - C30    Kikan       [  10 ]
   - C45    Denna       [   2 ]
   - C60    Pangkey     [   2 ]

   Postorder :
   - C30    Kikan       [  10 ]
   - C20    Leslie      [   5 ]
   - C60    Pangkey     [   2 ]
   - C45    Denna       [   2 ]
```

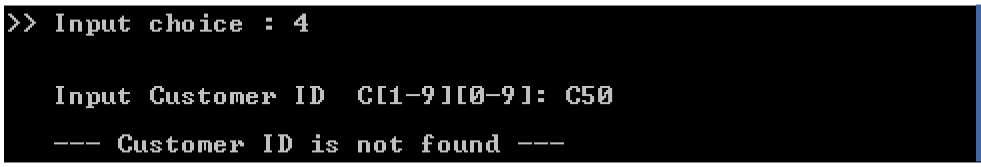


Print Screen of Delete a certain data (Menu '4') when the tree was still empty

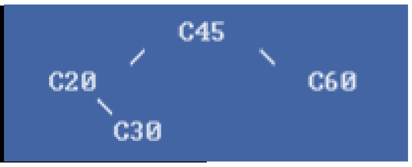


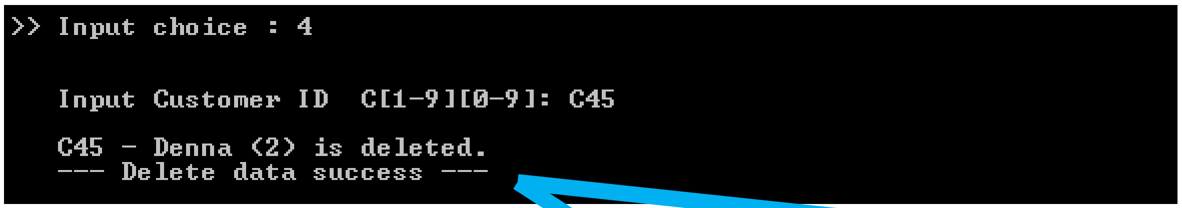
```
>> Input choice : 4

   --- There is no data in the Tree ---
```

Print Screen of Delete a certain data (Menu '4') when the tree is not empty

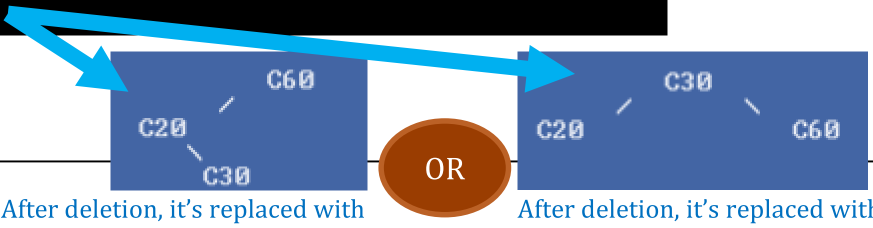


```
>> Input choice : 4

   Input Customer ID  C[1-9][0-9]: C50

   --- Customer ID is not found ---
```





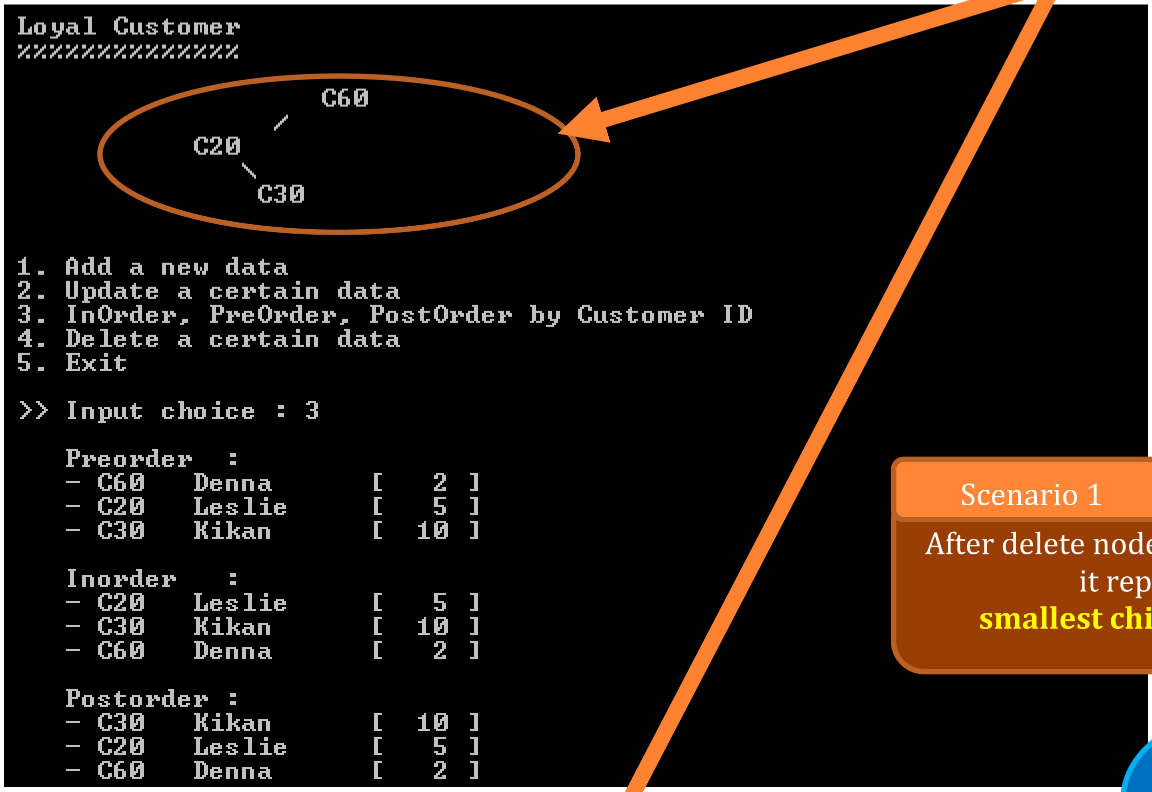
```
>> Input choice : 4

   Input Customer ID  C[1-9][0-9]: C45

   C45 - Denna (2) is deleted.
   --- Delete data success ---
```



After deletion, it's replaced with **smallest child** of **right child**

After deletion, it's replaced with **largest child** of **left child**

<u>CHALLENGE</u> = add the display/visualisation of this Binary Search Tree, like this!

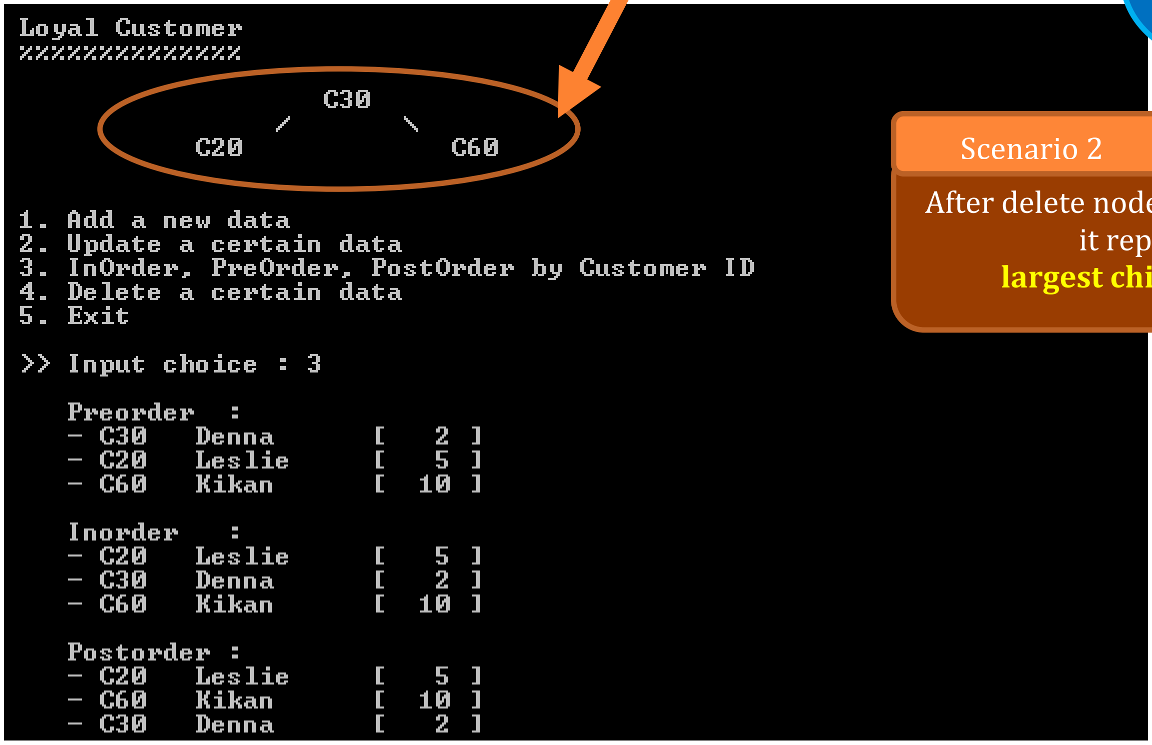


```
Loyal Customer
%%%%%%%%%%%%%%

                C60
              /
        C20
          \
            C30

1. Add a new data
2. Update a certain data
3. InOrder, PreOrder, PostOrder by Customer ID
4. Delete a certain data
5. Exit

>> Input choice : 3

   Preorder   :
   - C60    Denna      [   2 ]
   - C20    Leslie     [   5 ]
   - C30    Kikan      [  10 ]

   Inorder    :
   - C20    Leslie     [   5 ]
   - C30    Kikan      [  10 ]
   - C60    Denna      [   2 ]

   Postorder :
   - C30    Kikan      [  10 ]
   - C20    Leslie     [   5 ]
   - C60    Denna      [   2 ]
```

Scenario 1

After delete node that has 2 children, it replace with **smallest child** of **right child**

OR



```
Loyal Customer
%%%%%%%%%%%%%%

                C30
              /       \
        C20               C60

1. Add a new data
2. Update a certain data
3. InOrder, PreOrder, PostOrder by Customer ID
4. Delete a certain data
5. Exit

>> Input choice : 3

   Preorder   :
   - C30    Denna      [   2 ]
   - C20    Leslie     [   5 ]
   - C60    Kikan      [  10 ]

   Inorder    :
   - C20    Leslie     [   5 ]
   - C30    Denna      [   2 ]
   - C60    Kikan      [  10 ]

   Postorder :
   - C20    Leslie     [   5 ]
   - C60    Kikan      [  10 ]
   - C30    Denna      [   2 ]
```

Scenario 2

After delete node that has 2 children, it replace with **largest child** of **left child**

You can choose one of two replacement scenario