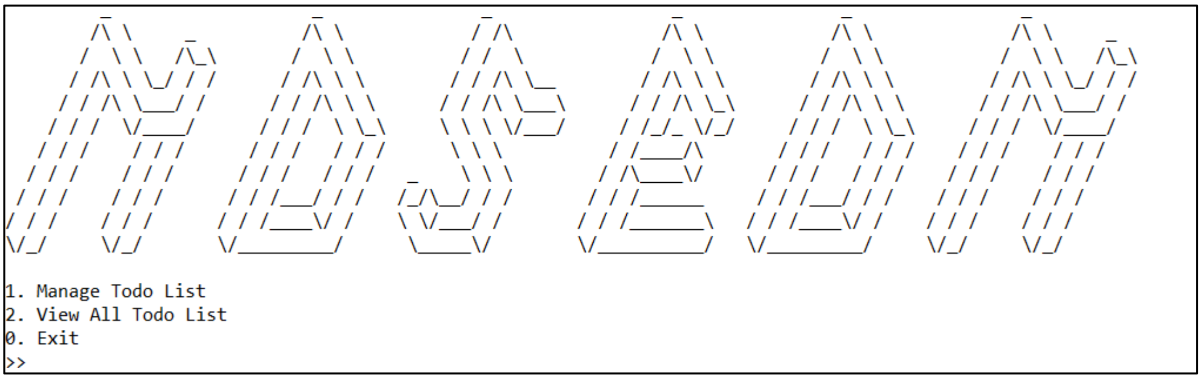
Soal

Case

noSEon

noSEon is a task application that makes you make the to-do list more organized. The **noSEon** has a **Todo** list then in the Todo list, you can add **Tasks**. The task has hour(s) and minute(s) that you give to give you information on how long it will take you to complete the task. You need to make this using **Single Linked List** and make sure to implement **Nested Structure**.

- **Home Page** (**Menu 1**)
  - This menu contains 3 menus, which are **Manage Todo List**, **View All Todo List**, and **Exit**.
  - **Prompt** user to **input chosen menu**. **Validate** the input must be **between 1 and 2 inclusively or 0 to exit**.



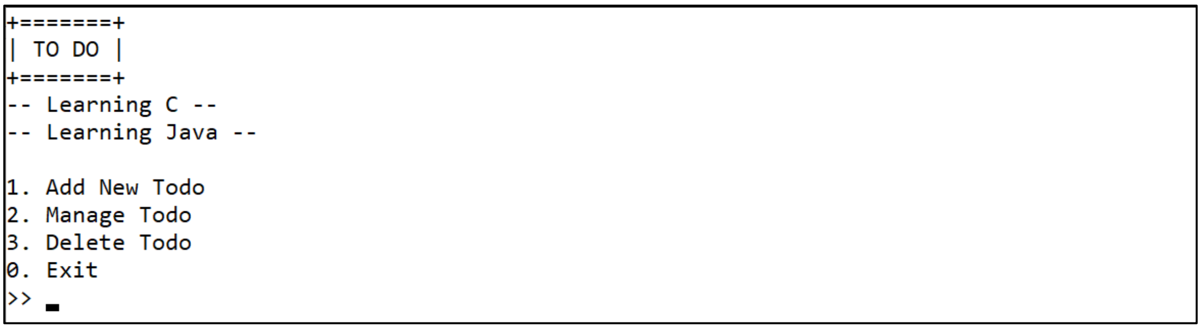
```
         _                   _               _                 _               _                _
        /\ \     _          /\ \            / /\              /\ \            /\ \             /\ \     _
       /  \ \   /\_\       /  \ \          / /  \            /  \ \          /  \ \           /  \ \   /\_\
      / /\ \ \_/ / /      / /\ \ \        / / /\ \__        / /\ \ \        / /\ \ \         / /\ \ \_/ / /
     / / /\ \___/ /      / / /\ \ \      / / /\ \___\      / / /\ \_\      / / /\ \ \       / / /\ \___/ /
    / / /  \/____/      / / /  \ \_\     \ \ \ \/___/     / /_/_ \/_/     / / /  \ \_\     / / /  \/____/
   / / /    / / /      / / /   / / /      \ \ \          / /____/\       / / /   / / /    / / /    / / /
  / / /    / / /      / / /   / / /   _    \ \ \        / /\____\/      / / /   / / /    / / /    / / /
 / / /    / / /      / / /___/ / /   /_/\__/ / /       / / /______     / / /___/ / /    / / /    / / /
/ / /    / / /      / / /____\/ /    \ \/___/ /       / / /_______\   / / /____\/ /    / / /    / / /
\/_/     \/_/       \/_________/      \_____\/        \/__________/   \/_________/     \/_/     \/_/

1. Manage Todo List
2. View All Todo List
0. Exit
>>
```

Figure 1. Home Page

1. If the user chooses **Manage Todo List** (**Menu 1**), then:
   - Display all **Todo(s)**.
   - This menu contains 4 menus, which are **Add New Todo**, **Manage Todo List**, **Delete Todo**, and **Exit**.

- **Prompt** user to **input chosen menu**. **Validate** the input must **between 1 and 3 inclusively or 0 to exit**.

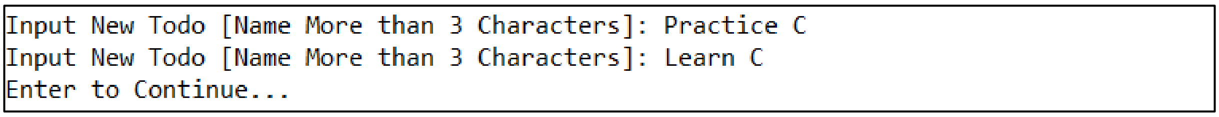


```
+=======+
| TO DO |
+=======+
-- Learning C --
-- Learning Java --

1. Add New Todo
2. Manage Todo
3. Delete Todo
0. Exit
>> 
```

Figure 2. Manage Todo List Page

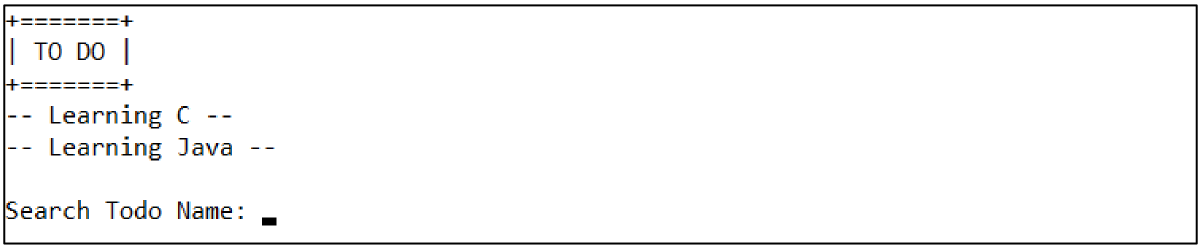
1. If the user chooses **Add New Todo** (**Menu 1**), then:
   - **Prompt** user to input **Todo Name**. **Validate** the Todo Name must be **more than 3 characters** and **cannot** be **empty**.
   - **Validate** that Todo Name must be **unique**.
   - **Todo Name must start with “Learn” (case sensitive)**



```
Input New Todo [Name More than 3 Characters]: Practice C
Input New Todo [Name More than 3 Characters]: Learn C
Enter to Continue...
```

Figure 3. Add New Todo Page

2. If the user chooses to **Manage Todo** (**Menu 2**), then:
   - Display all **Todo(s)**.
   - **Prompt** user to input **Todo Name**. **Validate** that user can search the Todo by Todo Name to manage the Todo. If the search **does not exist**, then go back to the **Manage Todo List Page (figure 2)**.



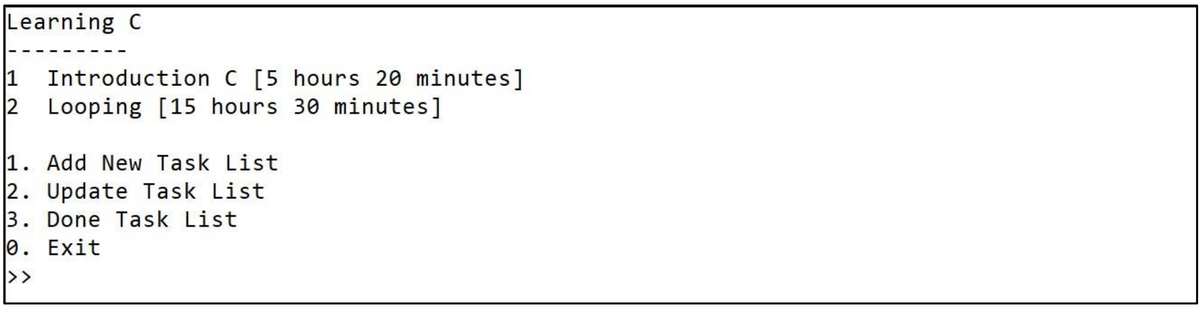
```
+=======+
| TO DO |
+=======+
-- Learning C --
-- Learning Java --

Search Todo Name: 
```

Figure 4. Search Page

   - If the search **exists**, then:

- Display **all Task(s)** in the **Todo**.
- This menu contains 4 menus, which are **Add New Task List**, **Update Task List**, **Done Task List**, and **Exit**.
- **Prompt** user to **input chosen menu**. **Validate** the input must **between 1 and 3 inclusively or 0 to exit**.

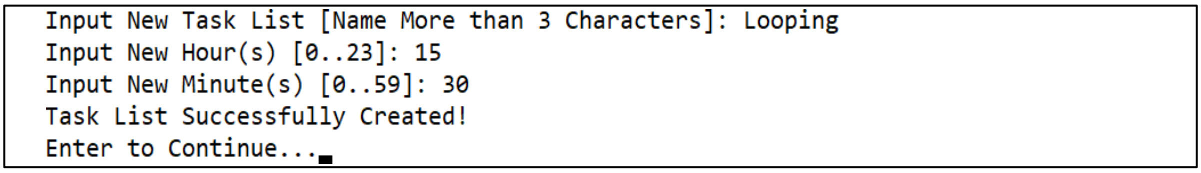


```
Learning C
---------
1  Introduction C [5 hours 20 minutes]
2  Looping [15 hours 30 minutes]

1. Add New Task List
2. Update Task List
3. Done Task List
0. Exit
>>
```

Figure 5. Manage Todo Page

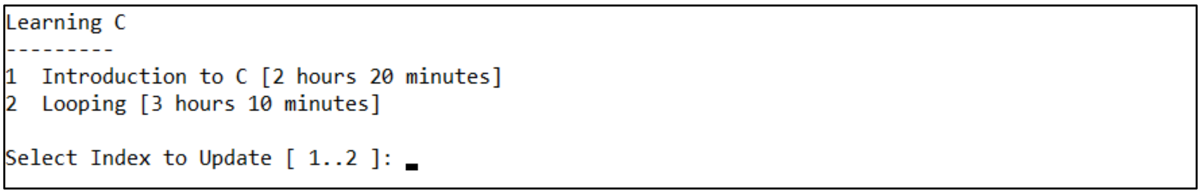
1. If the user chooses **Add New Task List** (**Menu 1**), then:
   - **Validate** that the Task Name must be **more than 3 characters** and **must be unique**.
   - **Validate** that hour **between 0 and 23** (**inclusive**).
   - **Validate** that minute **between 0 and 59** (**inclusive**).
   - **Validate** that **hours and minutes cannot be 0** (**need at least 1 minute**).



```
  Input New Task List [Name More than 3 Characters]: Looping
  Input New Hour(s) [0..23]: 15
  Input New Minute(s) [0..59]: 30
  Task List Successfully Created!
  Enter to Continue...
```

Figure 6. Add New Task List Page

2. If the user chooses **Update Task List** (**Menu 2**), then:
   - Display **all Task(s)** in the **Todo**.
   - **Validate** that the user needs to **select the index** of the Task to **update** the Task.
   - Display the **index** from **1 to the length of the Task(s)**.
   - **Validate** if the user selects a Task index that **does not exist** then, print “**There's No Task in that index**”.

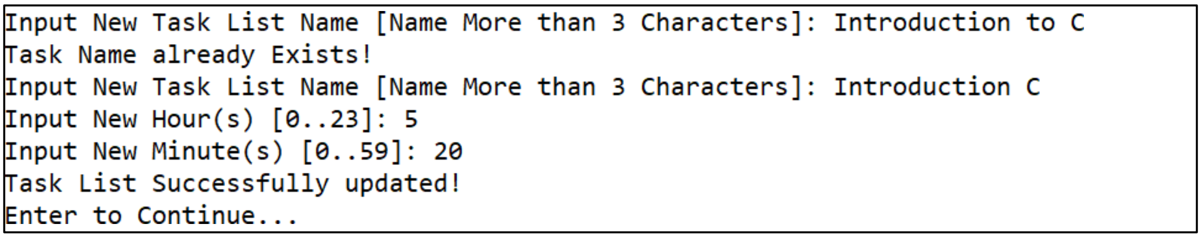


```
Learning C
---------
1  Introduction to C [2 hours 20 minutes]
2  Looping [3 hours 10 minutes]

Select Index to Update [ 1..2 ]: _
```

Figure 7a. Update Task List Page

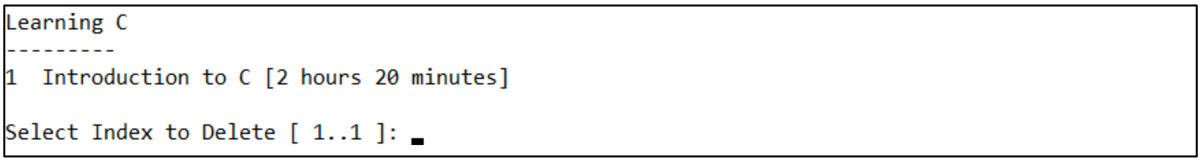
- If the user selects a Task index that **exists**, then:
    - **Validate** that the Task Name must be **more than 3 characters** and **must be unique**.
    - **Validate** that hour **between 0 and 23** (**inclusive**).
    - **Validate** that minute **between 0 and 59** (**inclusive**).
    - **Validate** that **hours and minutes cannot be 0** (**need at least 1 minute**).



```
Input New Task List Name [Name More than 3 Characters]: Introduction to C
Task Name already Exists!
Input New Task List Name [Name More than 3 Characters]: Introduction C
Input New Hour(s) [0..23]: 5
Input New Minute(s) [0..59]: 20
Task List Successfully updated!
Enter to Continue...
```

Figure 7b. Update Task List Page

3. If the user chooses **Done Task List** (**Menu 3**), then:
    - Display **all Task(s)** in the **Todo**.
    - **Validate** that the user needs to **select the index** of the Task to **delete** the Task.
    - Display the **index** from **1 to the length of the Task(s)**.
    - **Validate** if the user selects a Task index that **does not exist** then, print “**There's No Task in that index**”.
    - If the user selects a task index that exists, then **delete the task** from **todo**.



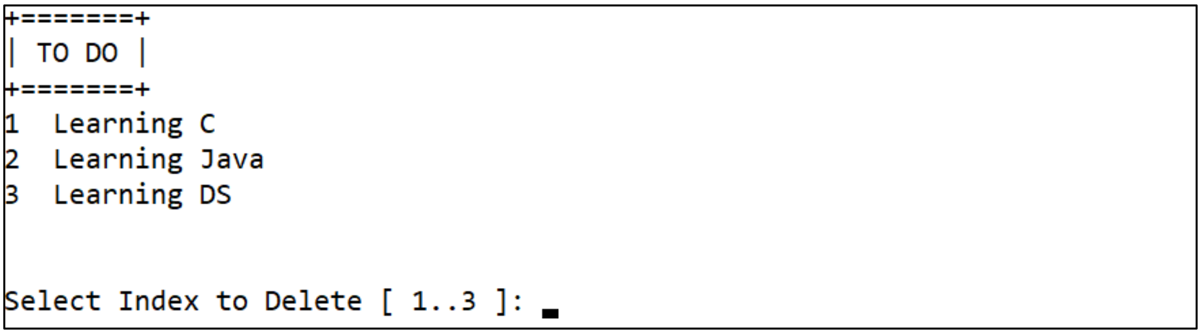
```
Learning C
---------
1  Introduction to C [2 hours 20 minutes]

Select Index to Delete [ 1..1 ]: _
```

Figure 8. Done Task List Page

3. If the user chooses **Delete Todo** (**Menu 3**), then:
    - Display all **Todo(s)**.

- **Validate** that the user needs to **select the index** of the Todo to **delete** the Todo.
- **Display** the index from **1 to the length of the Todo(s)**.
- **Validate** if the user selects a Todo index that **does not exist** then, print “**There's No Todo List in that index**”.
- If the user selects a Todo index that **exists**, then **delete the Todo**.

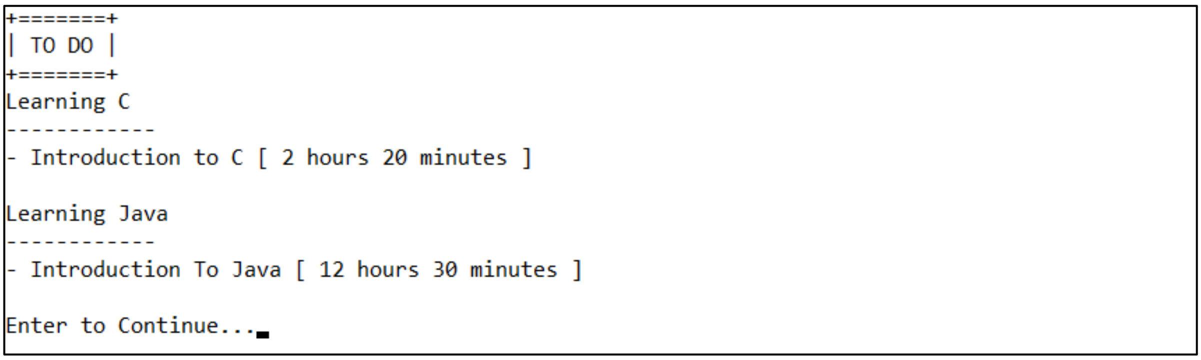


```
+=======+
| TO DO |
+=======+
1  Learning C
2  Learning Java
3  Learning DS


Select Index to Delete [ 1..3 ]: 
```

Figure 9. Delete Todo Page

2. If the user chooses **View All Todo List** (**Menu 2**), then:
   - Show **All Todo(s) and the Task(s)**.



```
+=======+
| TO DO |
+=======+
Learning C
------------
- Introduction to C [ 2 hours 20 minutes ]

Learning Java
------------
- Introduction To Java [ 12 hours 30 minutes ]

Enter to Continue...
```

Figure 10. View All Todo List

3. If the user **chooses Exit (Menu 0),** then:
   - Print “Goodbye!” and exit from the Program