Module Code	:	Introduction to C Programming (032022-MTG) - Computing
Intake Code	:	APD1F2109IT
Lecturer Name	:	Ms. Marry Ting
Hand in Date	:	22nd June 2022
Name	:	Lee Lik Shann
TP number	:	TP060916

Table of Contents

1.0 Introduction

This report demonstrates a personal task management system that is created using C programming language. This system is aimed to help user to better manage and monitor their task with basic functions such as storing, displaying, update and deleting their tasks.

From the user's perspective, one will be starting off with a user menu with a list of options that contains add task, view task, update task, delete task, search specific task, sort completed task, sort incomplete task, check current date and time and exit. User can choose any option and input the data accordingly. For adding task function, user will have to enter the state, deadline, task name and category to store the data into a text file. While viewing, user will be able to see all of the task records.

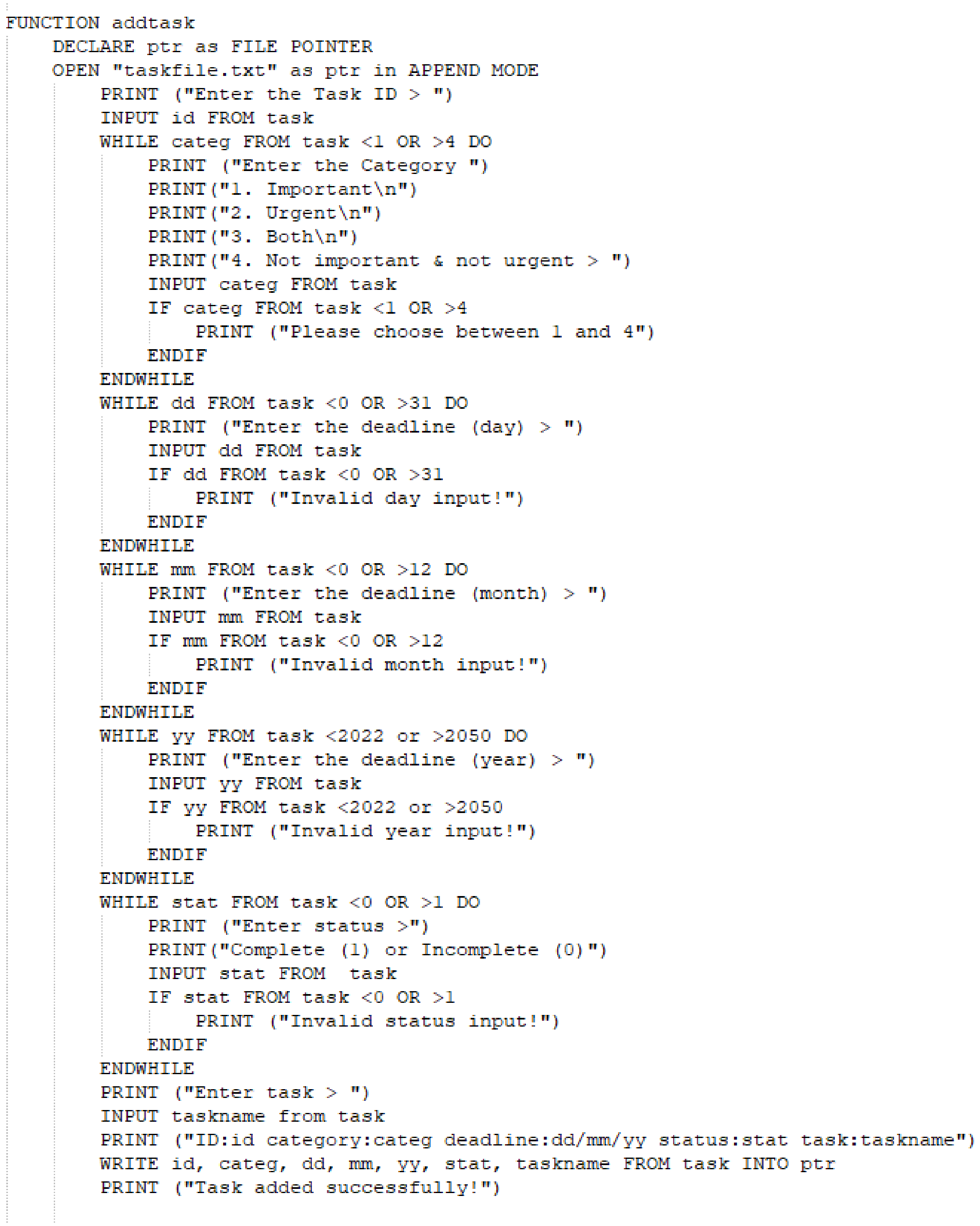
1.1 Assumptions

1. I assume that user remember and able to recognize different category with number from 1 to 4.
2. I assume that user uses same task ID for different task.
3. I assume that user do not know the current date and time
4. I assume that user will update when the date is done
5. I assume that user's task deadlines year do not exceed 2050.
6. I assume that user is following Malaysia's date and time

2.0 Design of the program

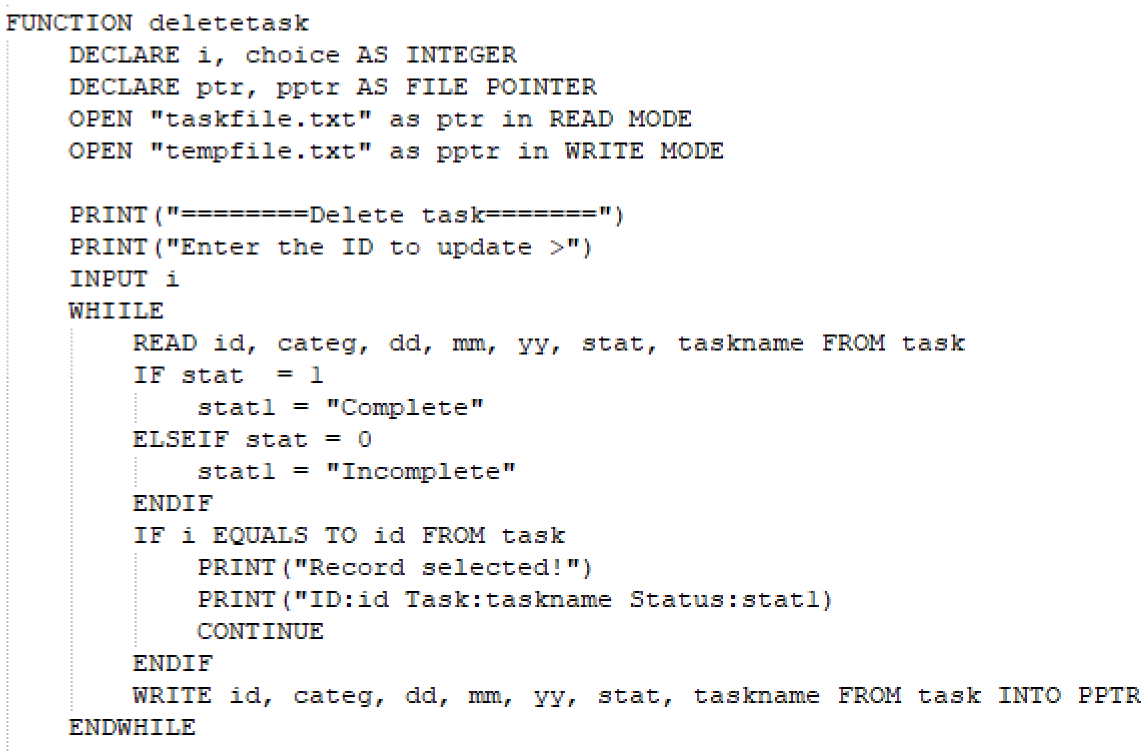
2.1 Pseudocode

Add tasks



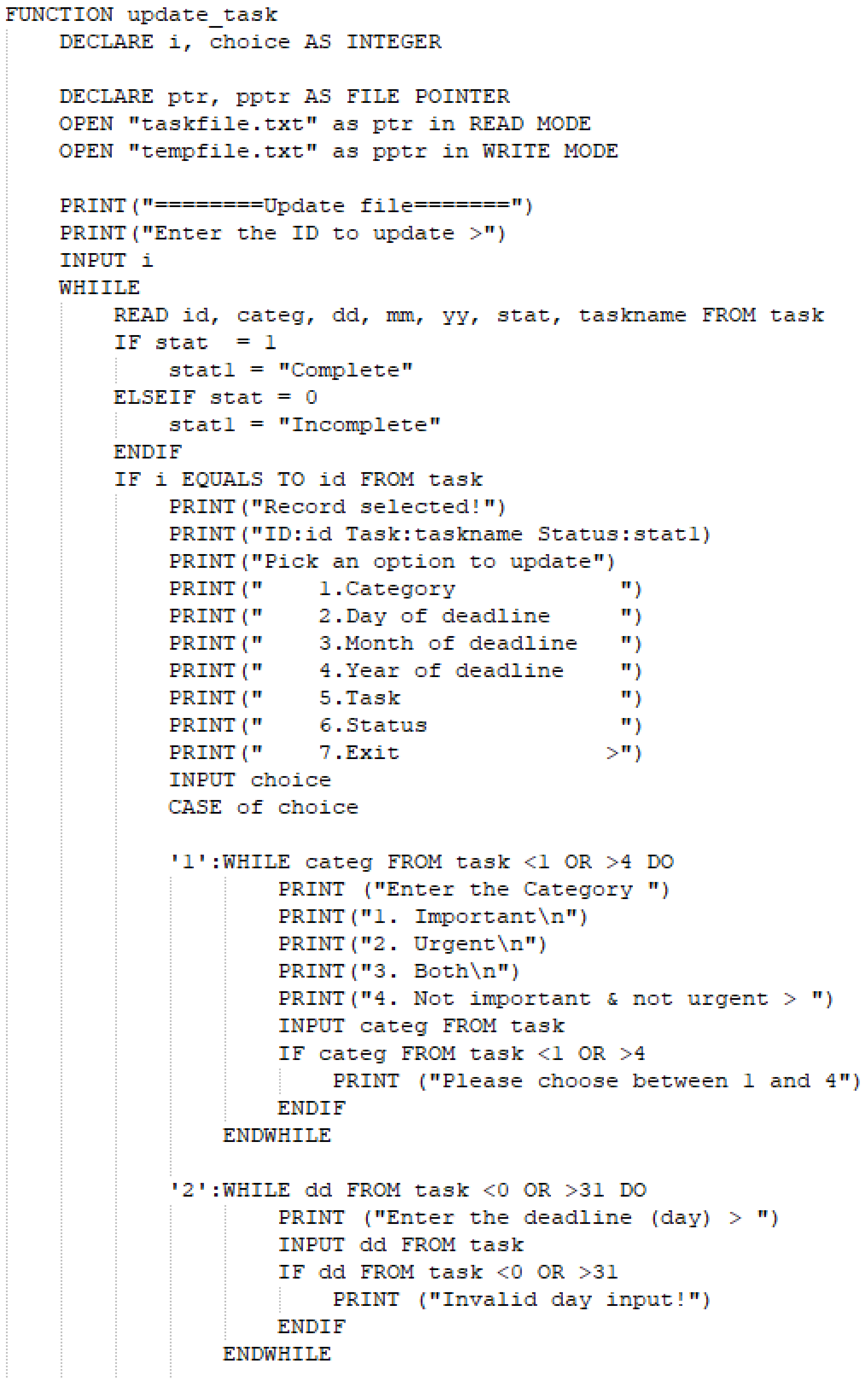
```
FUNCTION addtask
    DECLARE ptr as FILE POINTER
    OPEN "taskfile.txt" as ptr in APPEND MODE
        PRINT ("Enter the Task ID > ")
        INPUT id FROM task
        WHILE categ FROM task <1 OR >4 DO
            PRINT ("Enter the Category ")
            PRINT("1. Important\n")
            PRINT("2. Urgent\n")
            PRINT("3. Both\n")
            PRINT("4. Not important & not urgent > ")
            INPUT categ FROM task
            IF categ FROM task <1 OR >4
                PRINT ("Please choose between 1 and 4")
            ENDIF
        ENDWHILE
        WHILE dd FROM task <0 OR >31 DO
            PRINT ("Enter the deadline (day) > ")
            INPUT dd FROM task
            IF dd FROM task <0 OR >31
                PRINT ("Invalid day input!")
            ENDIF
        ENDWHILE
        WHILE mm FROM task <0 OR >12 DO
            PRINT ("Enter the deadline (month) > ")
            INPUT mm FROM task
            IF mm FROM task <0 OR >12
                PRINT ("Invalid month input!")
            ENDIF
        ENDWHILE
        WHILE yy FROM task <2022 or >2050 DO
            PRINT ("Enter the deadline (year) > ")
            INPUT yy FROM task
            IF yy FROM task <2022 or >2050
                PRINT ("Invalid year input!")
            ENDIF
        ENDWHILE
        WHILE stat FROM task <0 OR >1 DO
            PRINT ("Enter status >")
            PRINT("Complete (1) or Incomplete (0)")
            INPUT stat FROM  task
            IF stat FROM task <0 OR >1
                PRINT ("Invalid status input!")
            ENDIF
        ENDWHILE
        PRINT ("Enter task > ")
        INPUT taskname from task
        PRINT ("ID:id category:categ deadline:dd/mm/yy status:stat task:taskname")
        WRITE id, categ, dd, mm, yy, stat, taskname FROM task INTO ptr
        PRINT ("Task added successfully!")
```

Delete tasks

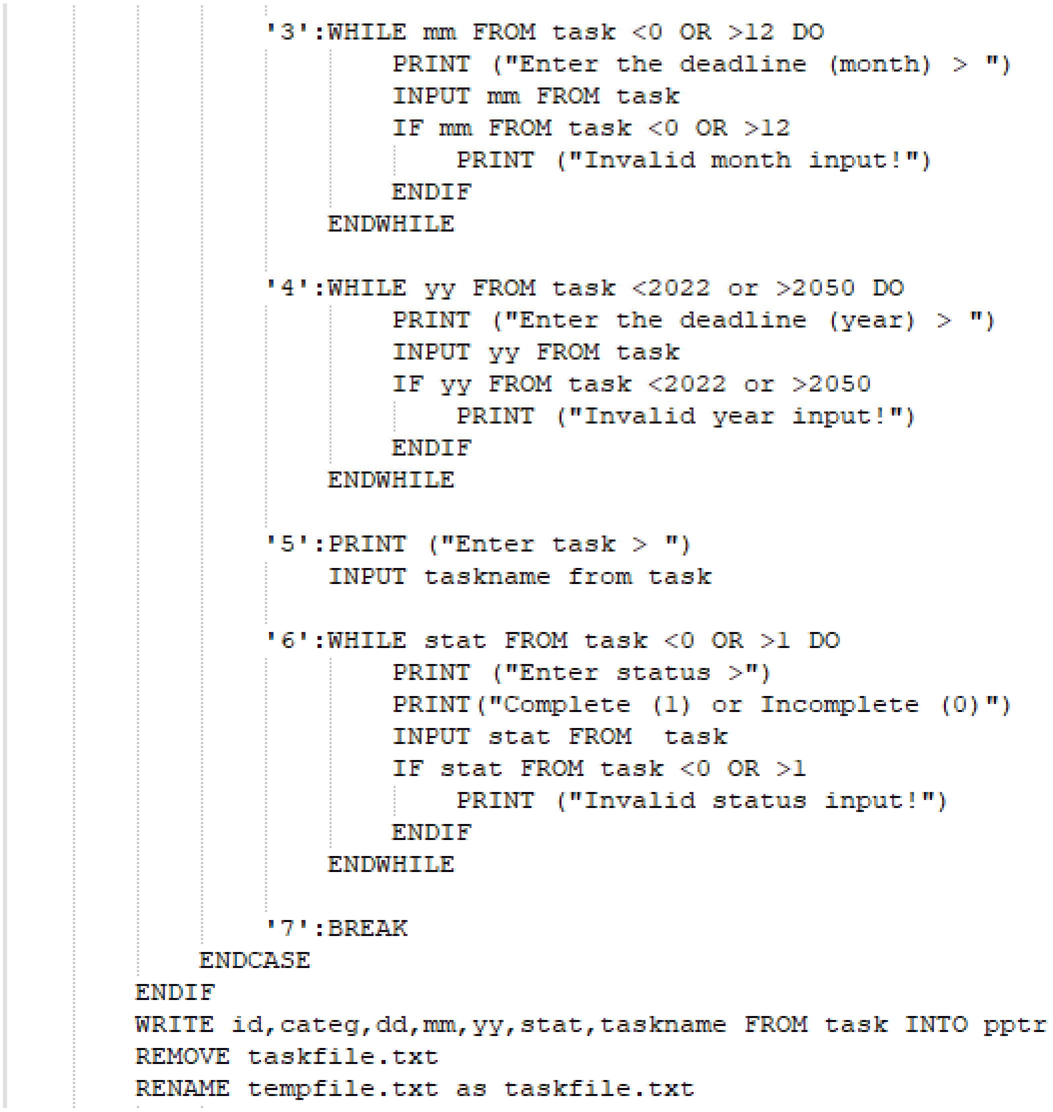


```
FUNCTION deletetask
    DECLARE i, choice AS INTEGER
    DECLARE ptr, pptr AS FILE POINTER
    OPEN "taskfile.txt" as ptr in READ MODE
    OPEN "tempfile.txt" as pptr in WRITE MODE

    PRINT("========Delete task=======")
    PRINT("Enter the ID to update >")
    INPUT i
    WHIILE
        READ id, categ, dd, mm, yy, stat, taskname FROM task
        IF stat  = 1
            statl = "Complete"
        ELSEIF stat = 0
            statl = "Incomplete"
        ENDIF
        IF i EQUALS TO id FROM task
            PRINT("Record selected!")
            PRINT("ID:id Task:taskname Status:statl)
            CONTINUE
        ENDIF
        WRITE id, categ, dd, mm, yy, stat, taskname FROM task INTO PPTR
    ENDWHILE
```

Update tasks

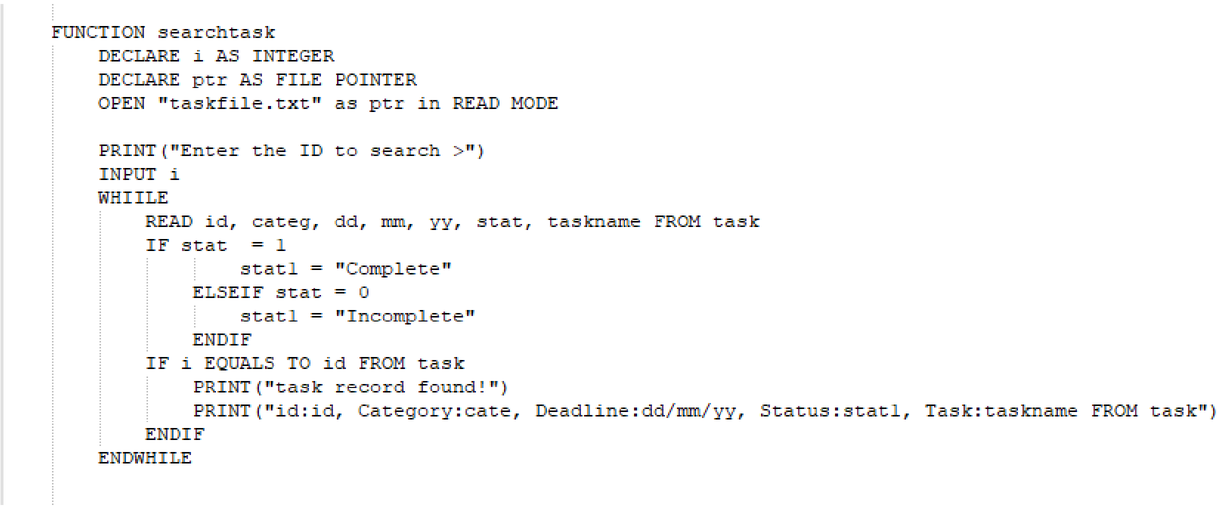


```
FUNCTION update_task
    DECLARE i, choice AS INTEGER

    DECLARE ptr, pptr AS FILE POINTER
    OPEN "taskfile.txt" as ptr in READ MODE
    OPEN "tempfile.txt" as pptr in WRITE MODE

    PRINT("========Update file=======")
    PRINT("Enter the ID to update >")
    INPUT i
    WHIILE
        READ id, categ, dd, mm, yy, stat, taskname FROM task
        IF stat  = 1
            stat1 = "Complete"
        ELSEIF stat = 0
            stat1 = "Incomplete"
        ENDIF
        IF i EQUALS TO id FROM task
            PRINT("Record selected!")
            PRINT("ID:id Task:taskname Status:stat1)
            PRINT("Pick an option to update")
            PRINT("    1.Category                ")
            PRINT("    2.Day of deadline         ")
            PRINT("    3.Month of deadline       ")
            PRINT("    4.Year of deadline        ")
            PRINT("    5.Task                    ")
            PRINT("    6.Status                  ")
            PRINT("    7.Exit                   >")
            INPUT choice
            CASE of choice

            '1':WHILE categ FROM task <1 OR >4 DO
                    PRINT ("Enter the Category ")
                    PRINT("1. Important\n")
                    PRINT("2. Urgent\n")
                    PRINT("3. Both\n")
                    PRINT("4. Not important & not urgent > ")
                    INPUT categ FROM task
                    IF categ FROM task <1 OR >4
                        PRINT ("Please choose between 1 and 4")
                    ENDIF
                ENDWHILE

            '2':WHILE dd FROM task <0 OR >31 DO
                    PRINT ("Enter the deadline (day) > ")
                    INPUT dd FROM task
                    IF dd FROM task <0 OR >31
                        PRINT ("Invalid day input!")
                    ENDIF
                ENDWHILE
```



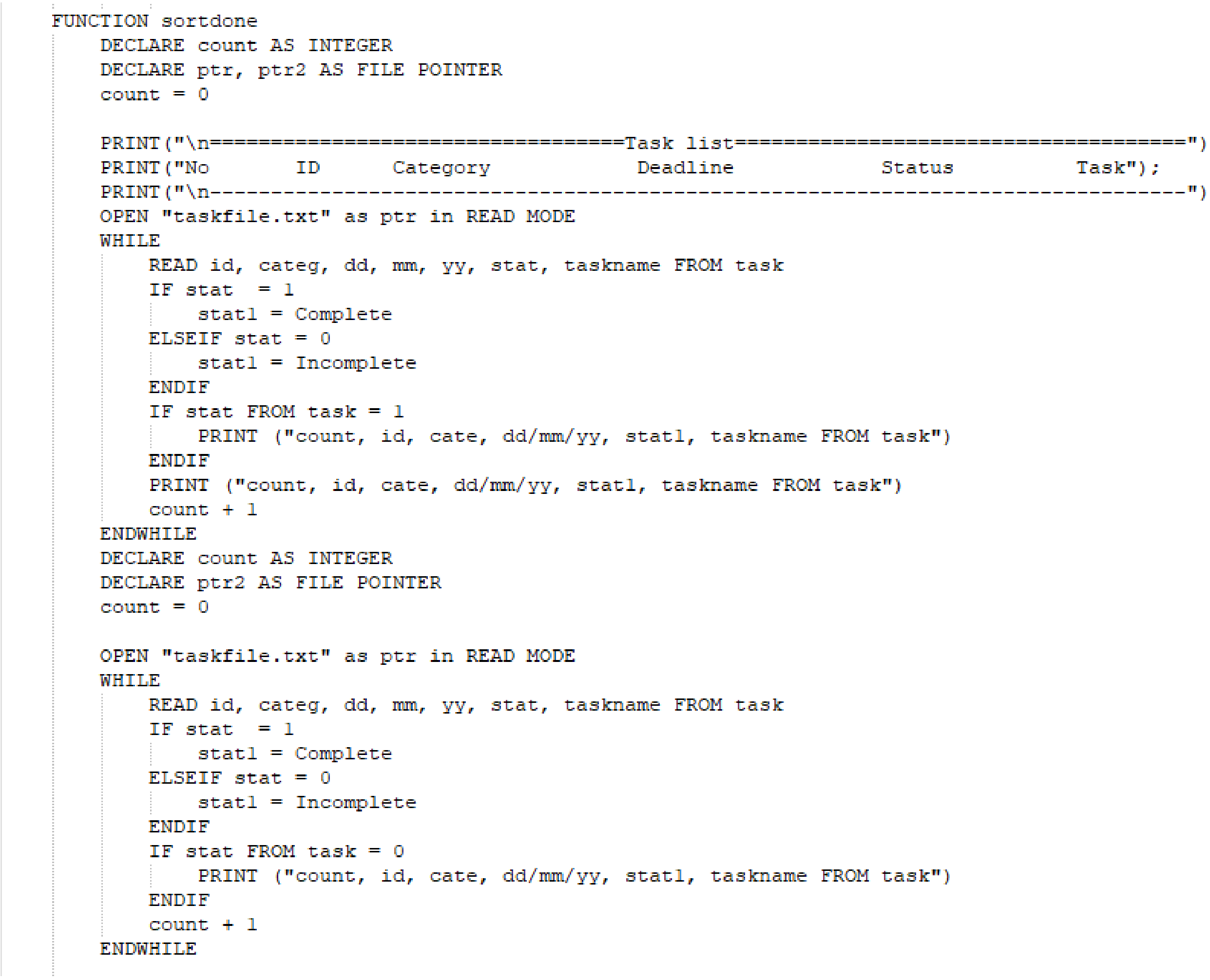
```
'3':WHILE mm FROM task <0 OR >12 DO
        PRINT ("Enter the deadline (month) > ")
        INPUT mm FROM task
        IF mm FROM task <0 OR >12
            PRINT ("Invalid month input!")
        ENDIF
    ENDWHILE

'4':WHILE yy FROM task <2022 or >2050 DO
        PRINT ("Enter the deadline (year) > ")
        INPUT yy FROM task
        IF yy FROM task <2022 or >2050
            PRINT ("Invalid year input!")
        ENDIF
    ENDWHILE

'5':PRINT ("Enter task > ")
    INPUT taskname from task

'6':WHILE stat FROM task <0 OR >1 DO
        PRINT ("Enter status >")
        PRINT("Complete (1) or Incomplete (0)")
        INPUT stat FROM  task
        IF stat FROM task <0 OR >1
            PRINT ("Invalid status input!")
        ENDIF
    ENDWHILE

'7':BREAK
ENDCASE
ENDIF
WRITE id,categ,dd,mm,yy,stat,taskname FROM task INTO pptr
REMOVE taskfile.txt
RENAME tempfile.txt as taskfile.txt
```

Search tasks



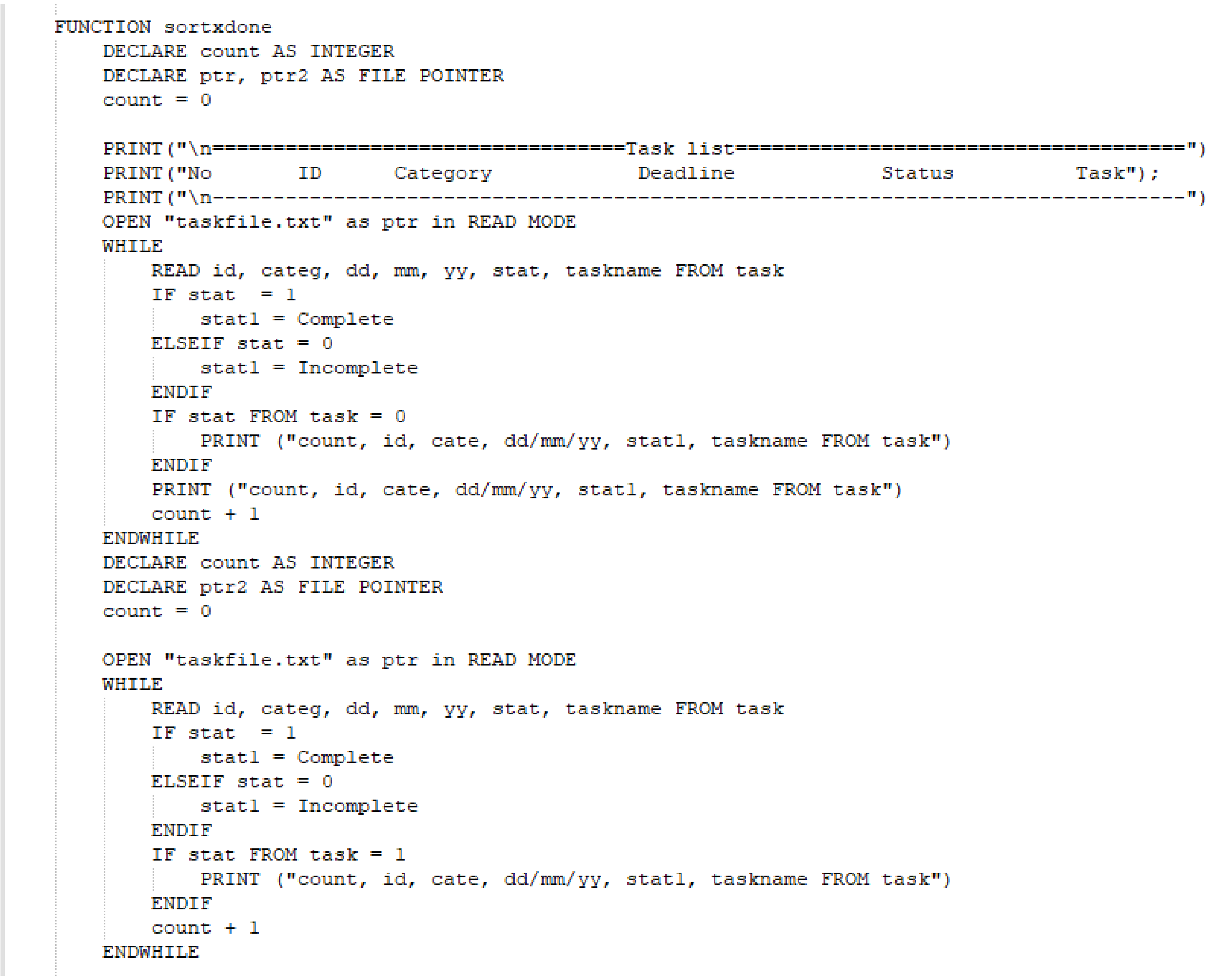
```
FUNCTION searchtask
    DECLARE i AS INTEGER
    DECLARE ptr AS FILE POINTER
    OPEN "taskfile.txt" as ptr in READ MODE

    PRINT("Enter the ID to search >")
    INPUT i
    WHIILE
        READ id, categ, dd, mm, yy, stat, taskname FROM task
        IF stat  = 1
                stat1 = "Complete"
            ELSEIF stat = 0
                stat1 = "Incomplete"
            ENDIF
        IF i EQUALS TO id FROM task
            PRINT("task record found!")
            PRINT("id:id, Category:cate, Deadline:dd/mm/yy, Status:stat1, Task:taskname FROM task")
        ENDIF
    ENDWHILE
```

Sort completed task



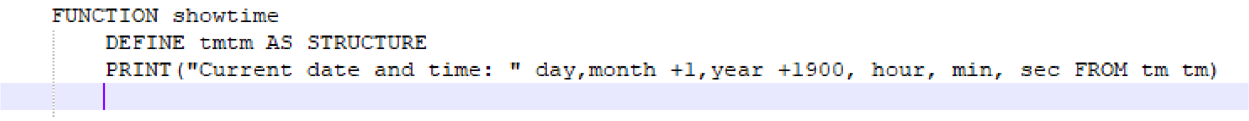
```
FUNCTION sortdone
    DECLARE count AS INTEGER
    DECLARE ptr, ptr2 AS FILE POINTER
    count = 0

    PRINT("\n==================================Task list=====================================")
    PRINT("No       ID      Category            Deadline            Status          Task");
    PRINT("\n--------------------------------------------------------------------------------")
    OPEN "taskfile.txt" as ptr in READ MODE
    WHILE
        READ id, categ, dd, mm, yy, stat, taskname FROM task
        IF stat  = 1
            stat1 = Complete
        ELSEIF stat = 0
            stat1 = Incomplete
        ENDIF
        IF stat FROM task = 1
            PRINT ("count, id, cate, dd/mm/yy, stat1, taskname FROM task")
        ENDIF
        PRINT ("count, id, cate, dd/mm/yy, stat1, taskname FROM task")
        count + 1
    ENDWHILE
    DECLARE count AS INTEGER
    DECLARE ptr2 AS FILE POINTER
    count = 0

    OPEN "taskfile.txt" as ptr in READ MODE
    WHILE
        READ id, categ, dd, mm, yy, stat, taskname FROM task
        IF stat  = 1
            stat1 = Complete
        ELSEIF stat = 0
            stat1 = Incomplete
        ENDIF
        IF stat FROM task = 0
            PRINT ("count, id, cate, dd/mm/yy, stat1, taskname FROM task")
        ENDIF
        count + 1
    ENDWHILE
```

Sort incomplete task



```
FUNCTION sortxdone
    DECLARE count AS INTEGER
    DECLARE ptr, ptr2 AS FILE POINTER
    count = 0

    PRINT("\n==================================Task list=====================================")
    PRINT("No       ID      Category            Deadline            Status          Task");
    PRINT("\n--------------------------------------------------------------------------------")
    OPEN "taskfile.txt" as ptr in READ MODE
    WHILE
        READ id, categ, dd, mm, yy, stat, taskname FROM task
        IF stat  = 1
            stat1 = Complete
        ELSEIF stat = 0
            stat1 = Incomplete
        ENDIF
        IF stat FROM task = 0
            PRINT ("count, id, cate, dd/mm/yy, stat1, taskname FROM task")
        ENDIF
        PRINT ("count, id, cate, dd/mm/yy, stat1, taskname FROM task")
        count + 1
    ENDWHILE
    DECLARE count AS INTEGER
    DECLARE ptr2 AS FILE POINTER
    count = 0

    OPEN "taskfile.txt" as ptr in READ MODE
    WHILE
        READ id, categ, dd, mm, yy, stat, taskname FROM task
        IF stat  = 1
            stat1 = Complete
        ELSEIF stat = 0
            stat1 = Incomplete
        ENDIF
        IF stat FROM task = 1
            PRINT ("count, id, cate, dd/mm/yy, stat1, taskname FROM task")
        ENDIF
        count + 1
    ENDWHILE
```

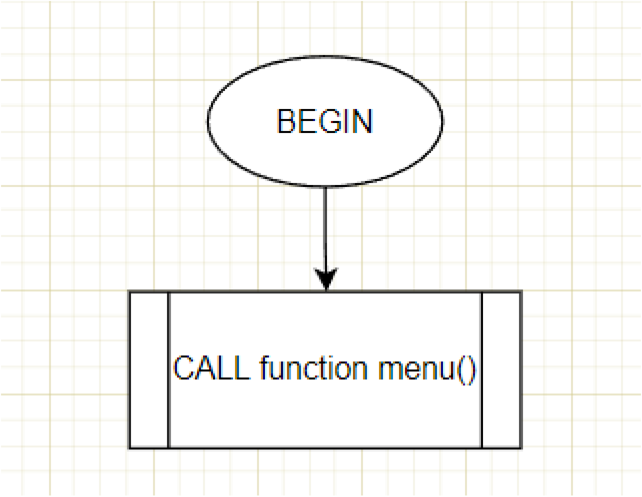
Show current time and date



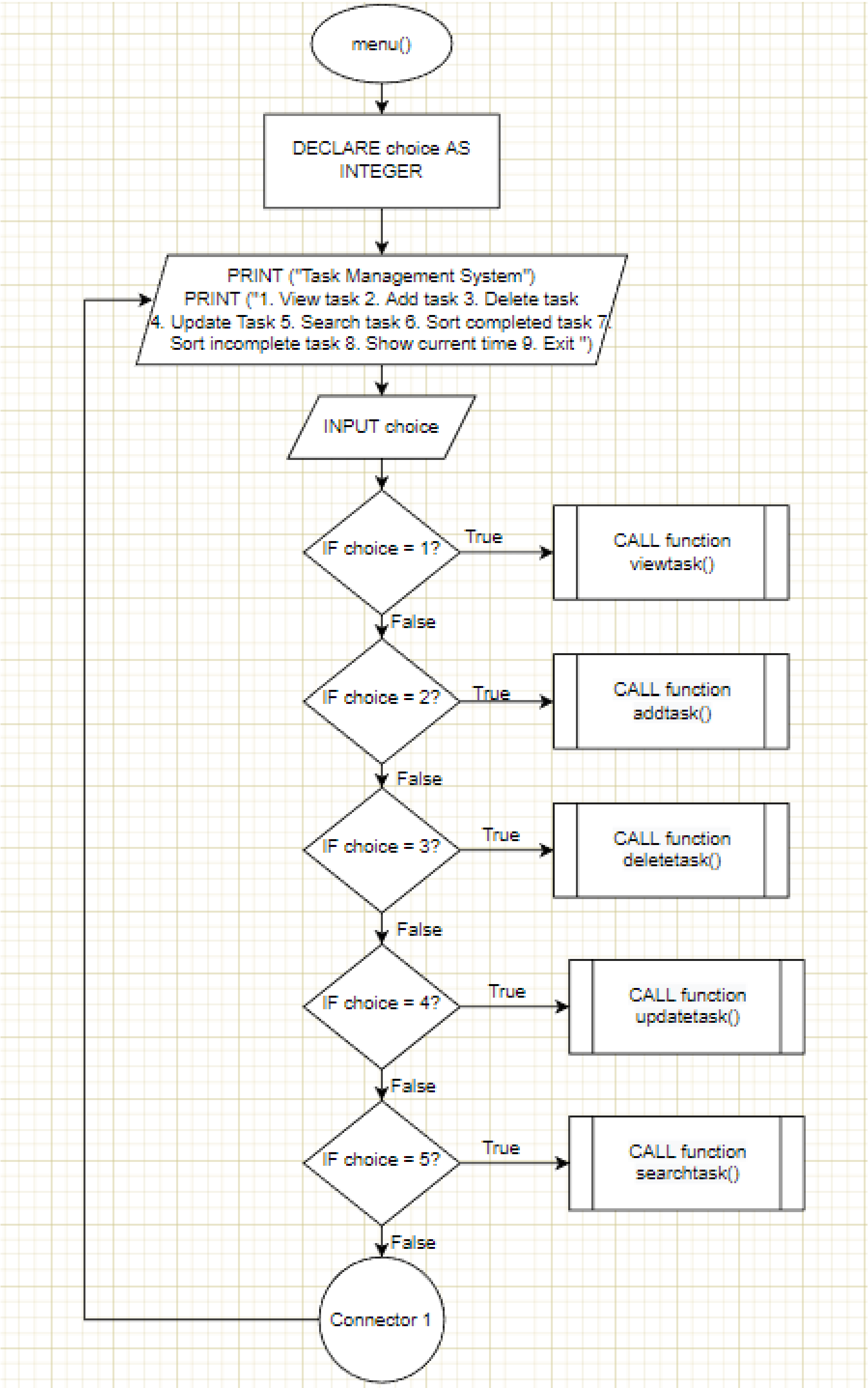
```
FUNCTION showtime
    DEFINE tmtm AS STRUCTURE
    PRINT("Current date and time: " day,month +1,year +1900, hour, min, sec FROM tm tm)
```

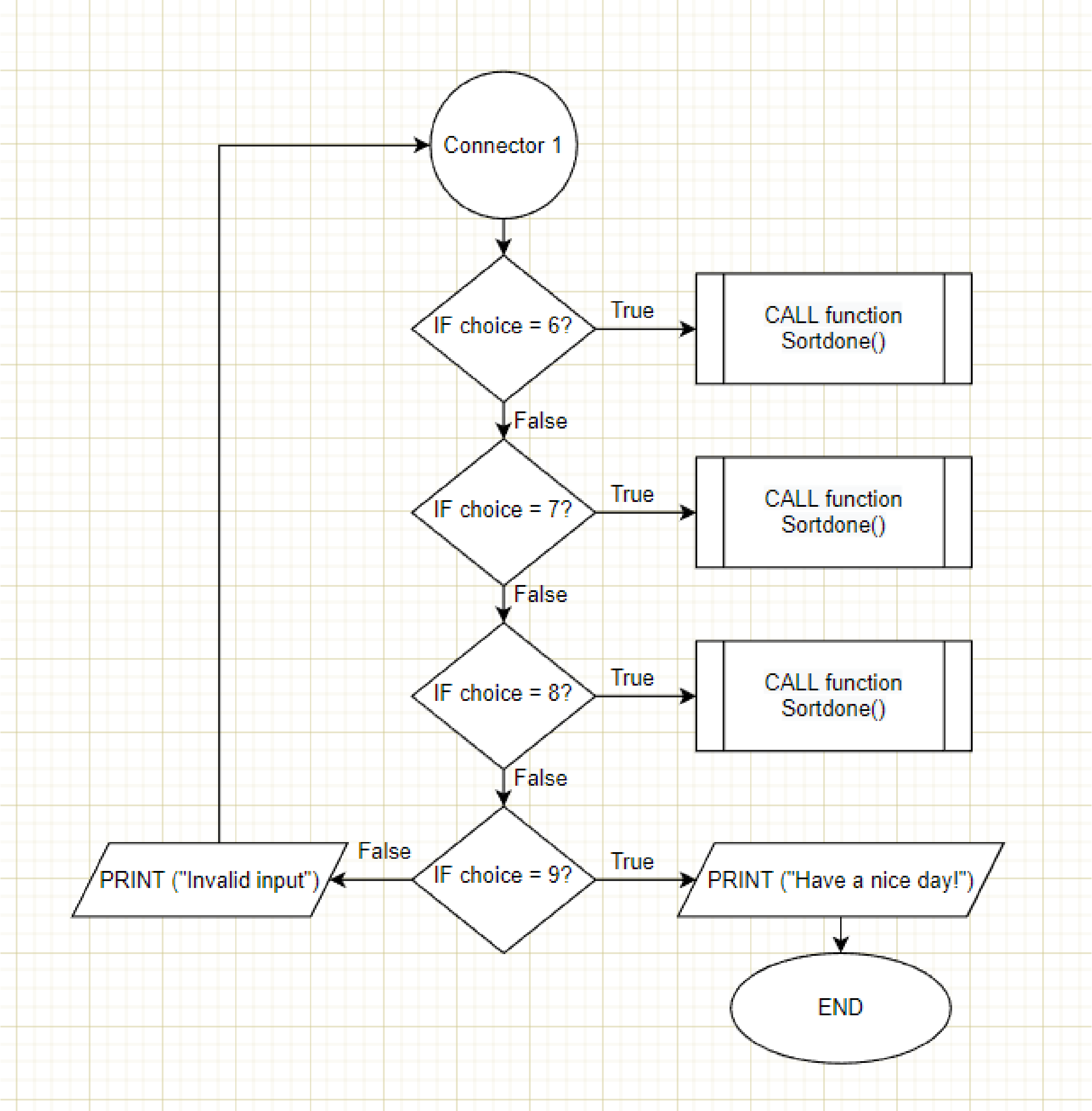
2.2 Flowchart

Main

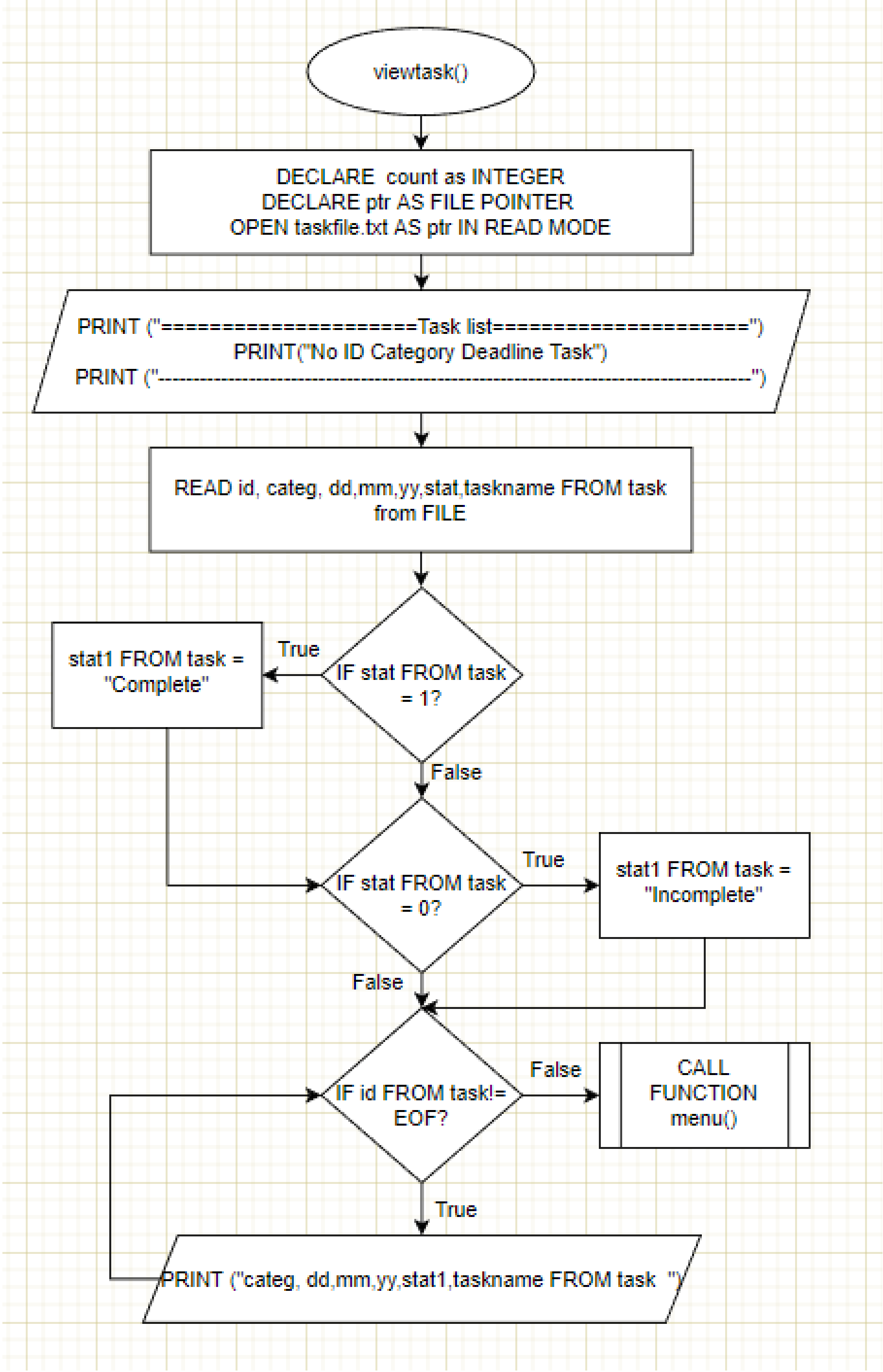


Menu


menu()
DECLARE choice AS INTEGER
PRINT ("Task Management System")
PRINT ("1. View task 2. Add task 3. Delete task 4. Update Task 5. Search task 6. Sort completed task 7. Sort incomplete task 8. Show current time 9. Exit ")
INPUT choice
IF choice = 1?
True
CALL function viewtask()
False
IF choice = 2?
True
CALL function addtask()
False
IF choice = 3?
True
CALL function deletetask()
False
IF choice = 4?
True
CALL function updatetask()
False
IF choice = 5?
True
CALL function searchtask()
False
Connector 1

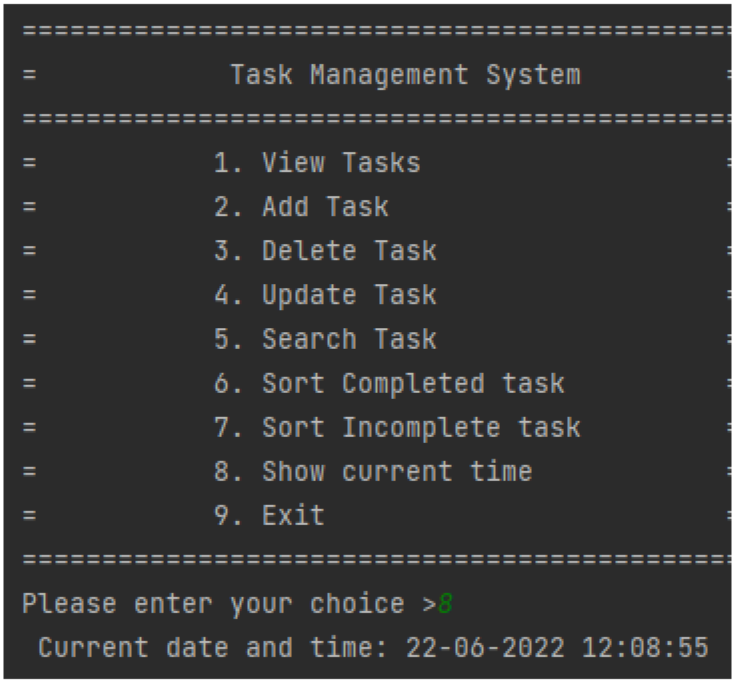

Connector 1
IF choice = 6?
True
CALL function Sortdone()
False
IF choice = 7?
True
CALL function Sortdone()
False
IF choice = 8?
True
CALL function Sortdone()
False
IF choice = 9?
False
PRINT ("Invalid input")
True
PRINT ("Have a nice day!")
END

View tasks


viewtask()
DECLARE count as INTEGER
DECLARE ptr AS FILE POINTER
OPEN taskfile.txt AS ptr IN READ MODE
PRINT ("===================Task list====================")
PRINT("No ID Category Deadline Task")
PRINT ("---")
READ id, categ, dd,mm,yy,stat,taskname FROM task from FILE
IF stat FROM task = 1?
True
stat1 FROM task = "Complete"
False
IF stat FROM task = 0?
True
stat1 FROM task = "Incomplete"
False
IF id FROM task!= EOF?
False
CALL FUNCTION menu()
True
PRINT ("categ, dd,mm,yy,stat1,taskname FROM task ")

3.0 Additional features

In this system, there is an additional feature for users which is a displaying current time function. This function is called showtime. This function is using <time.h> as external library. This is created for users to be aware of the time they have while referring to the task list that the user has stored in the system. Then, user will be able to measure and know how much time they have left with each task.



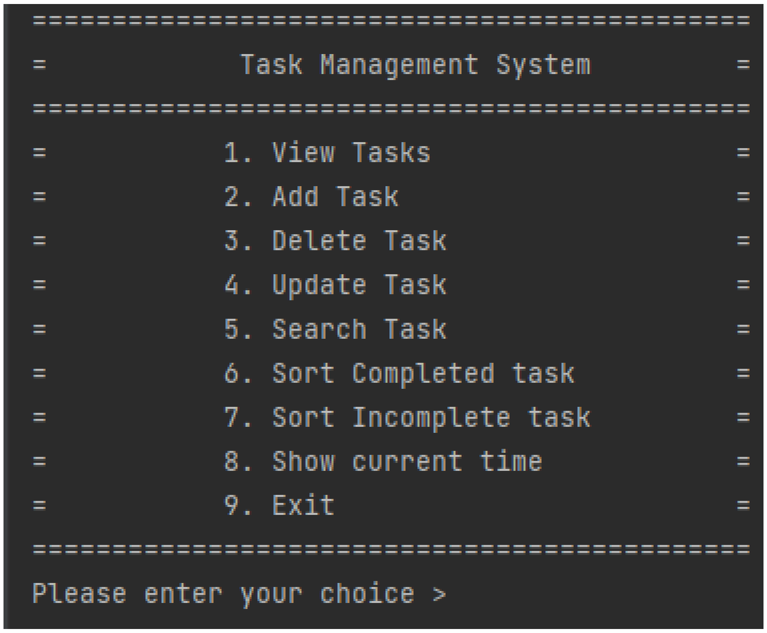
```
============================================
=            Task Management System        =
============================================
=           1. View Tasks                  =
=           2. Add Task                    =
=           3. Delete Task                 =
=           4. Update Task                 =
=           5. Search Task                 =
=           6. Sort Completed task         =
=           7. Sort Incomplete task        =
=           8. Show current time           =
=           9. Exit                        =
============================================
Please enter your choice >8
 Current date and time: 22-06-2022 12:08:55
```

Figure 3.1

In Figure 3.1 as shown above, user is able to print the current time with day of date, month of date, and the current year. Then there is also a clock that shows in 24-hour system that includes the hours, minutes and seconds.

4.0 Sample output

4.1.1 Menu

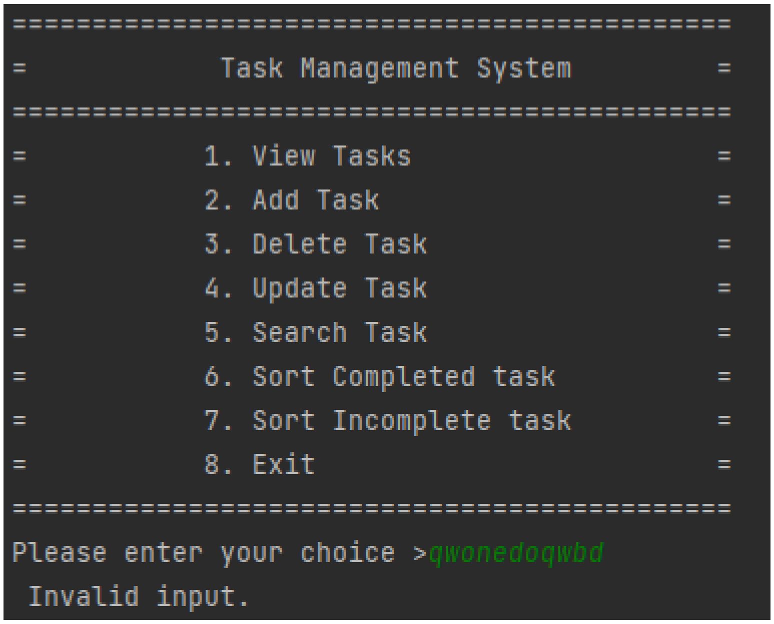


```
==============================================
=            Task Management System          =
==============================================
=           1. View Tasks                    =
=           2. Add Task                      =
=           3. Delete Task                   =
=           4. Update Task                   =
=           5. Search Task                   =
=           6. Sort Completed task           =
=           7. Sort Incomplete task          =
=           8. Show current time             =
=           9. Exit                          =
==============================================
Please enter your choice >
```

Figure 4.1.1

Figure 4.1.1 above is the first menu for user to view. In this menu, user will have a total of 8 choices which are view task, add task, delete task, update task, search task, sort complete task, sort incomplete task and exit.

4.1.2 Menu wrong input

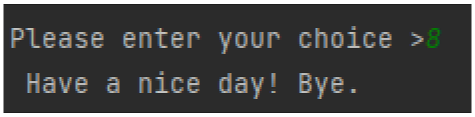


```
=============================================
=            Task Management System         =
=============================================
=           1. View Tasks                   =
=           2. Add Task                     =
=           3. Delete Task                  =
=           4. Update Task                  =
=           5. Search Task                  =
=           6. Sort Completed task          =
=           7. Sort Incomplete task         =
=           8. Exit                         =
=============================================
Please enter your choice >qwonedoqwbd
 Invalid input.
```

Figure 4.1.2

Figure 4.1.2 has shown an error when an invalid input which is given other than number 1, 2, 3, 4,5,6,7,8 is entered into the system. After that, it will tell the user that there invalid input is detected and will loop from the start again for user to re-enter the input.

4.1.2 Menu wrong input

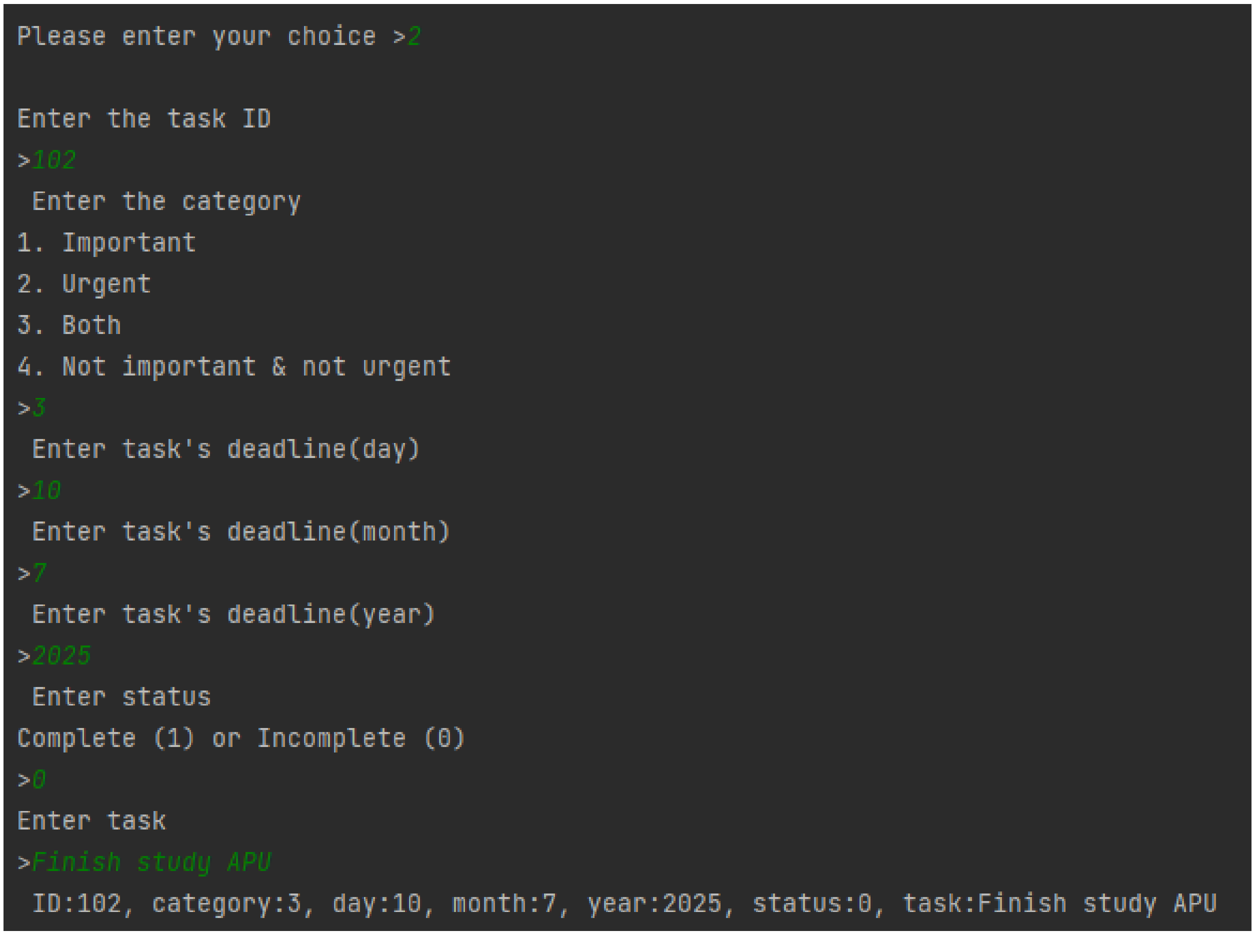


```
Please enter your choice >8
 Have a nice day! Bye.
```

Figure 4.1.3

In Figure 4.1.3 shows the loop of the menu stop looping after user has entered 8 as exit choice and displaying ‘Have a nice day! Bye.’

4.2.1 Add task

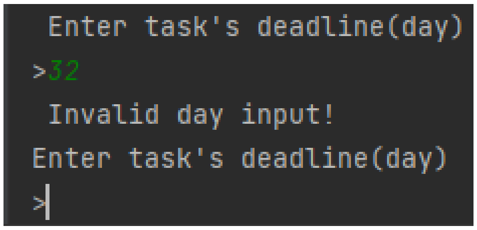


```
Please enter your choice >2

Enter the task ID
>102
 Enter the category
1. Important
2. Urgent
3. Both
4. Not important & not urgent
>3
 Enter task's deadline(day)
>10
 Enter task's deadline(month)
>7
 Enter task's deadline(year)
>2025
 Enter status
Complete (1) or Incomplete (0)
>0
Enter task
>Finish study APU
 ID:102, category:3, day:10, month:7, year:2025, status:0, task:Finish study APU
```

Figure 4.2.1

As shown figure 4.2.1 above, this is how user will be inputting a task. The system will ask user to provide task ID, category with 4 different choices which are ‘Important’, ‘Urgent’, ’Both’, and ‘Not important & not urgent’. Then, it will ask for deadline starting from day, month and then year. After that, user will be asked to enter status which are either ‘complete’ or ‘Incomplete’. In the last input, users will ask for the description of the task. Then, the system will show what is inputted by the user.

4.2.2 Add task invalid day input

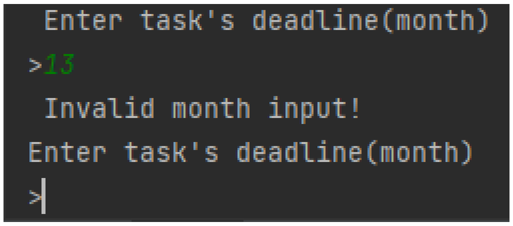


```
 Enter task's deadline(day)
>32
 Invalid day input!
Enter task's deadline(day)
>
```

Figure 4.2.2

In figure 4.2.2 above shows that an invalid input of ‘32’ is given by the user as the system do not accept a value that is greater than 31 for day of deadline. It will then show ‘Invalid day input!’ and loop until user has entered a validate input.

4.2.3 Add task invalid month input

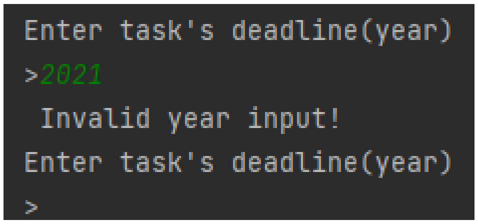


```
 Enter task's deadline(month)
>13
 Invalid month input!
Enter task's deadline(month)
>
```

Figure 4.2.3

In figure 4.2.4 above shows that an invalid input of ‘13’ is given by the user as the system do not accept a value that is greater than 12 for month of deadline. It will then show ‘Invalid month input!’ and loop until user has entered a proper input.

4.2.4 Add task invalid year input

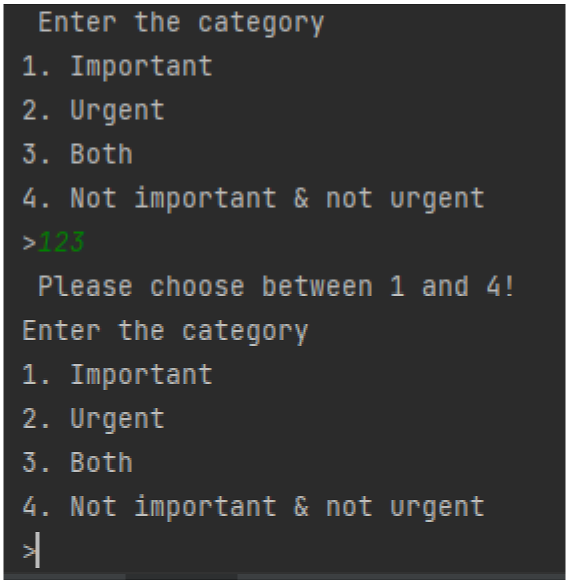


```
Enter task's deadline(year)
>2021
 Invalid year input!
Enter task's deadline(year)
>
```

Figure 4.2.4

In figure 4.2.2 above shows that an invalid input of ‘2021’ is given by the user as the system do not accept a value that is less than 2022 for day of deadline. It will then show ‘Invalid year input!’ and loop until user has entered a proper input.

4.2.5 Add task invalid category input

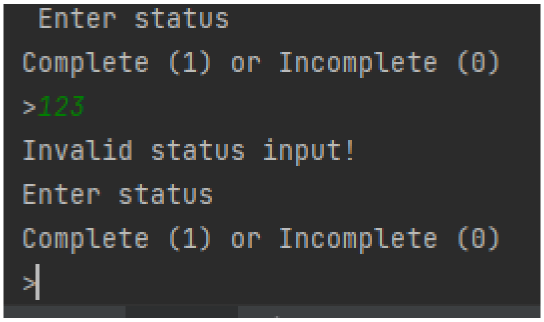


```
 Enter the category
1. Important
2. Urgent
3. Both
4. Not important & not urgent
>123
 Please choose between 1 and 4!
Enter the category
1. Important
2. Urgent
3. Both
4. Not important & not urgent
>
```

Figure 4.2.5

In figure 4.2.2 above shows that an invalid input of category. It will then show ‘Please choose between 1 and 4!’ and loop until user has entered a proper input.

4.2.6 Add task invalid status input

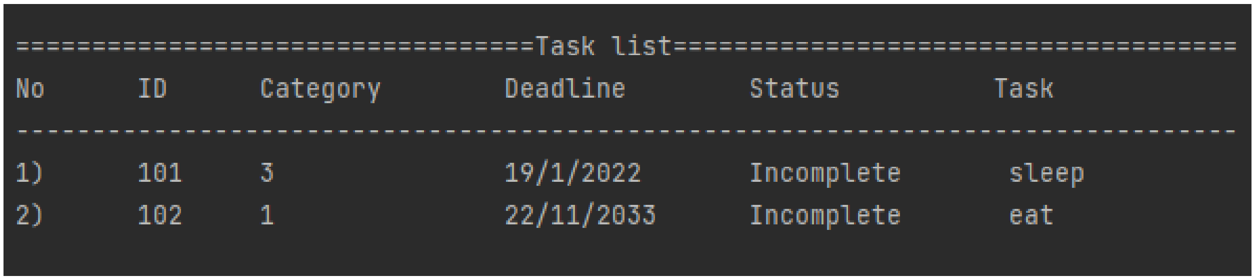


```
 Enter status
Complete (1) or Incomplete (0)
>123
Invalid status input!
Enter status
Complete (1) or Incomplete (0)
>
```

Figure 4.2.6

In figure 4.2.2 above shows that an invalid input of ‘2021’ is given by the user as the system do not accept a value that is less than 2021 for day of deadline.

4.3.1 View task

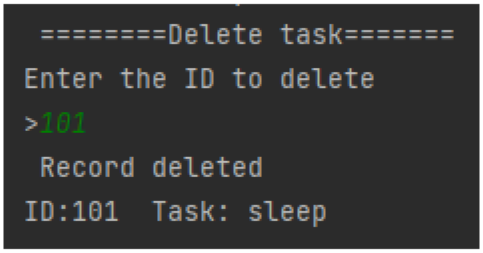


```
==================================Task list=======================================
No      ID      Category        Deadline        Status          Task
----------------------------------------------------------------------------------
1)      101     3               19/1/2022       Incomplete       sleep
2)      102     1               22/11/2033      Incomplete       eat
```

Figure 4.3.1

As shown above in Figure 4.3.1, it will show Numbers of record, and each ID, Category, Deadline, Status, and the task.

4.4.1 Delete task

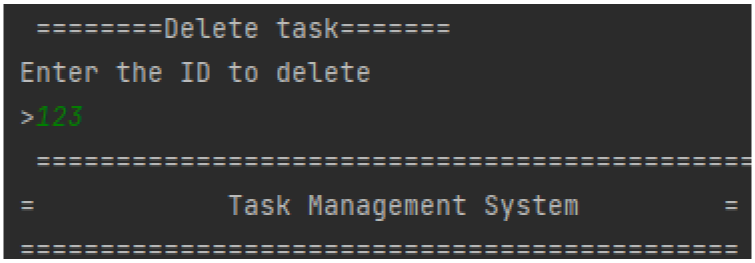


```
 ========Delete task=======
Enter the ID to delete
>101
 Record deleted
ID:101  Task: sleep
```

Figure 4.4.1

In Figure 4.4.1 shown when user has picked the delete task option in the menu. The system will ask for a ID to delete. If the entered ID is found, it will tell the user ‘Record deleted’ and show the ID and task description deleted.

4.4.2 Delete task

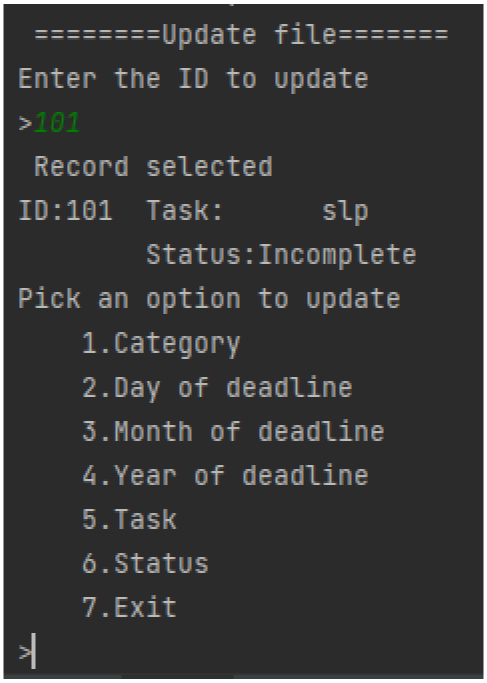


```
 ========Delete task=======
Enter the ID to delete
>123
 ============================================
=            Task Management System        =
============================================
```

Figure 4.4.2

Figure 4.4.2 shows that the user’s input is not found, it will direct return user back to the menu function.

4.5.1 Update task

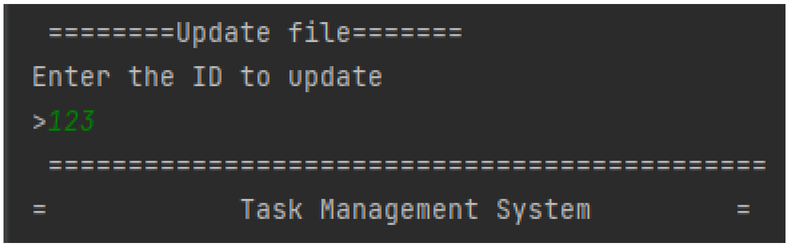


```
 ========Update file=======
Enter the ID to update
>101
 Record selected
ID:101  Task:     slp
        Status:Incomplete
Pick an option to update
    1.Category
    2.Day of deadline
    3.Month of deadline
    4.Year of deadline
    5.Task
    6.Status
    7.Exit
>
```

Figure 4.5.1

In figure 4.5.1 above show when the user’s input match the record in the text file. Then, the system will display the record that match the user’s input. Then, a list of display will be shown that includes category, day of deadline, month of deadline, year of deadline, task, status and exit.

4.5.2 Update task no record found

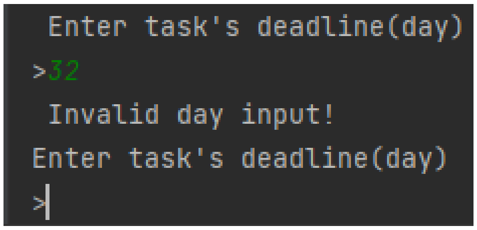


```
 ========Update file=======
Enter the ID to update
>123
 ============================================
=            Task Management System        =
```

Figure 4.5.2

In this figure 4.5.2 shows that the ID cannot be found in the text file. The system will then direct the user back to the menu function.

4.5.3 Update task invalid day input

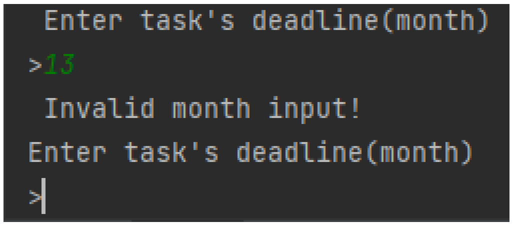


```
 Enter task's deadline(day)
>32
 Invalid day input!
Enter task's deadline(day)
>
```

Figure 4.5.3

As shown in Figure 4.5.3, the user has picked to update the day of the task's due date by entering an invalid input while updating the task's deadline. The system will loop until the user has entered a valid day input for the deadline.

4.5.4 Update task invalid month input

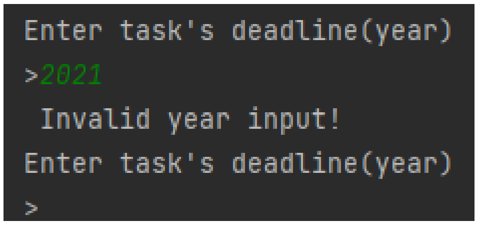


```
 Enter task's deadline(month)
>13
 Invalid month input!
Enter task's deadline(month)
>
```

Figure 4.5.4

As shown in Figure 4.5.4, the user has picked to update the month of the task's deadline with an invalid input for month while updating the task's deadline. The system will loop until the user has entered a valid day input for the deadline.

4.5.5 Update task invalid year input

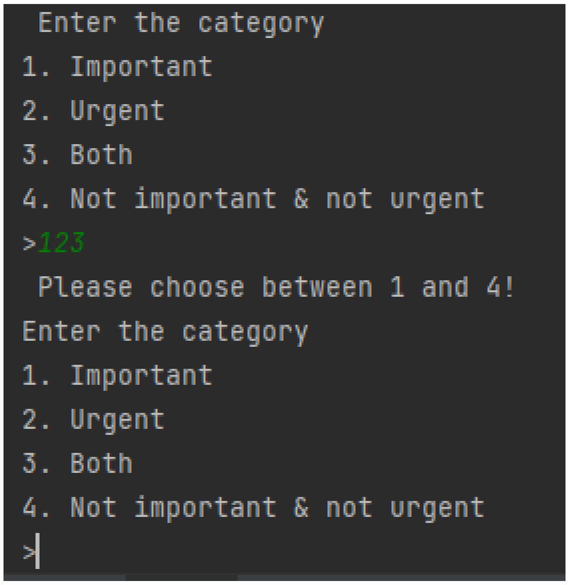


```
Enter task's deadline(year)
>2021
 Invalid year input!
Enter task's deadline(year)
>
```

Figure 4.5.5

As shown in Figure 4.5.4, the user has picked to update the month of the task's deadline with an invalid input for month while updating the task's deadline. The system will loop until the user has entered a valid day input for the deadline

4.5.6 Update task invalid category input

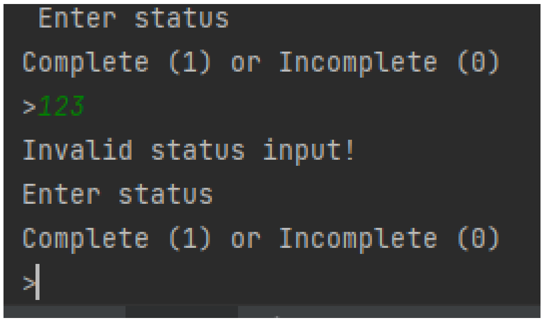


```
 Enter the category
1. Important
2. Urgent
3. Both
4. Not important & not urgent
>123
 Please choose between 1 and 4!
Enter the category
1. Important
2. Urgent
3. Both
4. Not important & not urgent
>
```

Figure 4.5.6

In figure 4.5.6, shows when the user gives an invalid input for task category, the system will loop to ask user for a valid input.

4.5.7 Update task invalid status

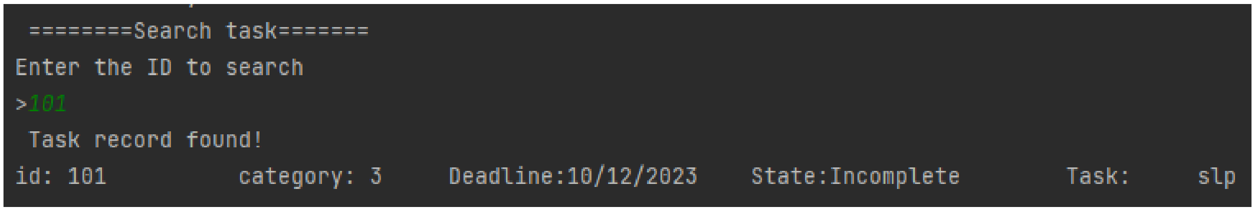


```
 Enter status
Complete (1) or Incomplete (0)
>123
Invalid status input!
Enter status
Complete (1) or Incomplete (0)
>
```

Figure 4.5.7

As shown in Figure 4.5.4, the user has picked to update the month of the task’s deadline with an invalid input for month while updating the task’s deadline. The system will loop until the user has entered a valid day input for the deadline.

4.6.1 Search task

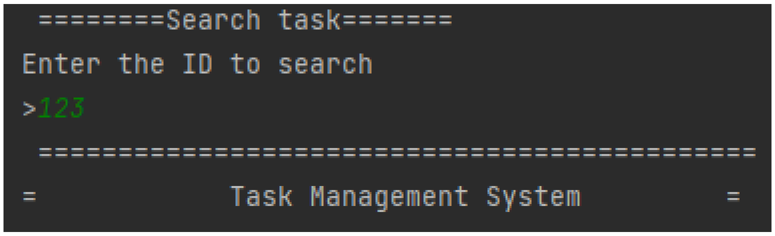


```
 ========Search task=======
Enter the ID to search
>101
 Task record found!
id: 101          category: 3     Deadline:10/12/2023    State:Incomplete       Task:     slp
```

Figure 4.6.1

In this figure 4.6.1 shows the search function. The system will ask for user to input the ID to search and will show the records that matches the user’s input along with other related description such as category, deadline, status, task.

4.6.2 Search task no record found

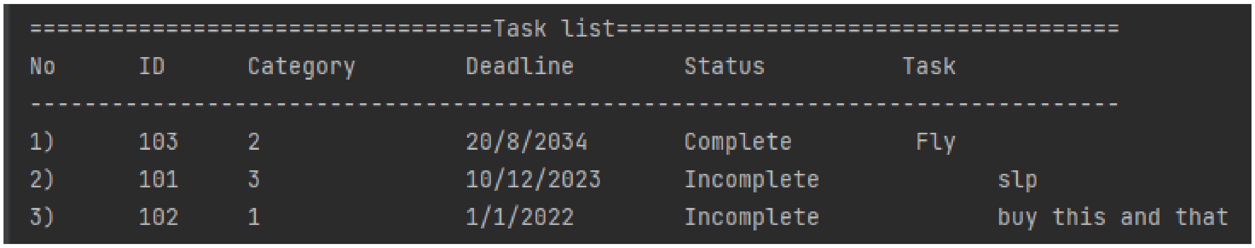


```
 ========Search task=======
Enter the ID to search
>123
 =============================================
=            Task Management System         =
```

Figure 4.6.2

In the figure 4.6.2 shown above, the system return user to the menu function after 0 result found in the text file.

4.7 Sort completed task

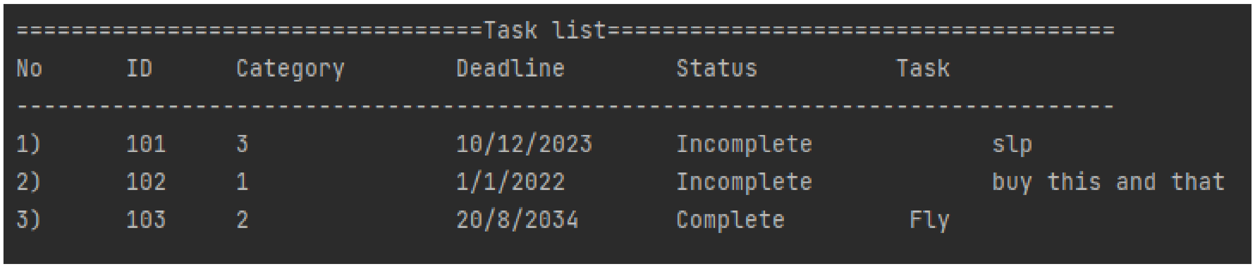


```
==================================Task list=====================================
No      ID      Category        Deadline        Status          Task
--------------------------------------------------------------------------------
1)      103     2               20/8/2034       Complete         Fly
2)      101     3               10/12/2023      Incomplete             slp
3)      102     1               1/1/2022        Incomplete             buy this and that
```

Figure 4.7.1

As Figure 4.7.1 shown above, the system has shown a sorted records of task from the text file. In the 1st record, it has shown a sorted completed records then incomplete record.

4.8.1 Sort incomplete task

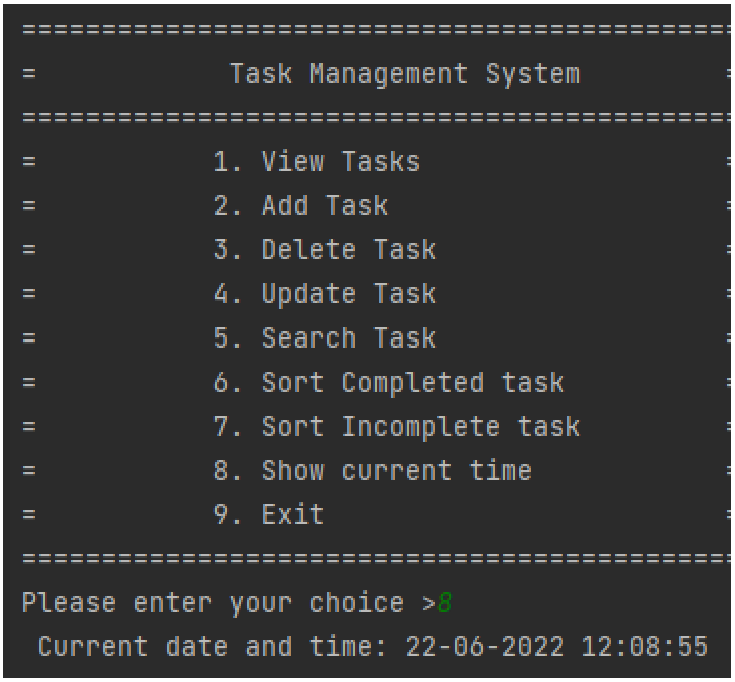


```
==================================Task list=====================================
No      ID      Category        Deadline        Status          Task
--------------------------------------------------------------------------------
1)      101     3               10/12/2023      Incomplete             slp
2)      102     1               1/1/2022        Incomplete             buy this and that
3)      103     2               20/8/2034       Complete         Fly
```

Figure 4.8.2

As Figure 4.8.1 shown above, the system has shown a sorted records of task from the text file. It has sorted the result with incomplete task records, then will only show completed task records after

4.9.1 Show current date & time



```
===========================================
=            Task Management System
===========================================
=           1. View Tasks
=           2. Add Task
=           3. Delete Task
=           4. Update Task
=           5. Search Task
=           6. Sort Completed task
=           7. Sort Incomplete task
=           8. Show current time
=           9. Exit
===========================================
Please enter your choice >8
 Current date and time: 22-06-2022 12:08:55
```

Figure 4.9.1

In Figure 4.9.1 shown above, the system shows the current date and time function.

5.0 Conclusion

As a conclusion, I have learned a lot of programming knowledge in this C programming module and further understand how important logical thinking to programming language is. During the process of coding, there were many challenges I have gone through. However, I have managed to overcome those difficulties with the help from many resources including Ms Marry Ting which is my lecturer. In regards of the personal task management system, there will always improvement space for me to make effort for to fully master C programming. Thus, I will not stop looking forward to learning deeper about coding and programming language not only C programming but also others even if it brings me a higher difficulty level of challenges.

6.0 References

Gowtham, S. (2020, October 2). *C program to display Current Date and Time*. Reactgo. https://reactgo.com/c-program-current-date-time/#:%7E:text=In%20C%20language%2C%20we%20can,()%20and%20ctime()%20functions.&text=Note%3A%20The%20time()%20%2C%20ctime,h%20header%20file.