INDIVIDUAL ASSIGNMENT

TECHNOLOGY PARK MALAYSIA

CT018-3-1-ICP

INTRODUCTION TO C PROGRAMMING

APD1F2111IT

HAND OUT DATE: 3/7/2022

HAND IN DATE: 13/8/2022

WEIGHTAGE: 50%

INSTRUCTIONS TO CANDIDATES:

1. Submit your assignment online in Moodle unless advised otherwise
2. Late submission will be awarded zero(0) unless Extenuating Circumstances (EC) are upheld
3. Cases of plagiarism will be penalized
4. You must obtain at least 50% in each component to pass this module

Table of Contents

1.)Introduction and Assumptions

In the last two years, the coronavirus disease (COVID-19) has killed millions of people. Despite the fact that we are approaching endemic status, countries are still following the relaxed COVID-19 SOP in order to prevent the spread of this disease. People are advised to wear face masks and maintain a social distance between each other.

All countries are actively assisting one another in combating the pandemic by exchanging masks and vaccines. Furthermore, they contribute to medical needs. One of the non-profit organizations (NGO) that receives medical supplies donations is the Malaysia Red Crescent Society.

In this scenario, The Malaysia Red Crescent Society has received millions of medical supplies from countries all around the world. COVID-19 Donation management system is required to keep track of the number of supplies given to the organization.

Assumptions

The donation management system will have few features which are important to track the record of supplies. These features are inventory creation, update inventory quantity, distribution of item to hospitals, search for specific item details, viewing the records and sorting it. The first function is where the system allows the creation of inventory by adding and writing the input from user to be saved into a file. The next function is the distribution of items where item quantity from storage will be deducted by the distribution quantity to its destination which records will be saved into another file.

There is also another function that is required in the system which is the function to show the records. The function that will be used is called view_inventory which will show the required data on the output terminal read from record file. Another essential function is to create a record of distribution, and this is where distribution_item function will do the work. In order to look for a specific data, search_item function will take the ID of the item and compare it with the record then if match is found the output terminal will print the details. Lastly the main function that is required is the sorting function which will sort all the item in descending order of quantities of items.

2.)Design of Program

Pseudocode

```
DECLARE id
    donate_code, distribute_code, NEXT id
DECLARE donate
    item_name, item_id, donator, donate_id, shipment, quantity, NEXT donate
DECLARE distribute
    distributed_item, distributed_quant, hospital, distribute_id, donation_did, NEXT distribute


DEFINE inventory_new FUNCTION
    DECLARE curr OF donate, save
    curr = start = NULL
    IF start = NULL THEN
        curr = start = ALLOCATE MEMORY
        CALL data_entry_donation FUNCTION
    ENDIF
    DISPLAY "Do you want to save the file ? 'y' to continue"
    READ save
    IF save = 'y' THEN
        CALL id_update FUNCTION
        CALL writetoFile FUNCTION
    ENDIF
END DEFINE
```

```
DEFINE inventory_add FUNCTION
    DECLARE curr OF donate, save
    CALL read_LinkedList FUNCTION
    curr = start = ALLOCATE MEMORY
    CALL data_entry_donation FUNCTION
    DISPLAY "Do you want to save the file ? 'y' to continue"
    READ save
    IF save = 'y' THEN
        CALL id_update FUNCTION
        CALL appendtoFile FUNCTION
    ENDIF
END DEFINE


DEFINE id_gen FUNCTION
    DECLARE fpt
    OPEN id.txt FILE
    IF fpt = NULL THEN
        DISPLAY "Error when opening file"
    ENDIF
    PRINT TO FILE "0DOID ; 0DIID"
    IF WRITING NOT = 0 THEN
        DISPLAY "ID has been generated succesfully"
```

```
    ELSE
        DISPLAY "Error when writing"
    ENDIF
    CLOSE id.txt FILE
END DEFINE


DEFINE id_update FUNCTION
    DECLARE t1, t2, fpt
    OPEN id.txt FILE
    IF fpt = NULL THEN
        DISPLAY "Error when opening file"
    ELSE
        READ t1, t2, FROM FILE
        IF idtype = 1 THEN
            DECLARE new_t1, new_t2
            CONVERT t1 to new_t1 + 1
            CONVERT t2 to new_t2
            PRINT TO FILE new_t1, new_t2
        ELSE IF idtype = 2 THEN
            DECLARE new_t1, new_t2
            CONVERT t1 to new_t1
            CONVERT t2 to new_t2 + 1
            PRINT TO FILE new_t1, new_t2
```

```
        ENDIF
    ENDIF
    CLOSE id.txt FILE
END DEFINE


DEFINE id_read FUNCTION
    DECLARE curr OF id, t1, t2, fpt
    OPEN id.txt
    IF fpt = NULL THEN
        DISPLAY "Error when opening file"
    ELSE
        READ t1, t2, FROM FILE
        IF idtype = 1 THEN
            donate_code = t1
        ELSE IF idtype = 2 THEN
            distribute_code = t2
        ENDIF
    ENDIF
    CLOSE id.txt FILE
    RETURN curr
END DEFINE


DEFINE data_entry_donation FUNCTION
```

```
    DECLARE idtype=1, curr OF donate, idcurr OF id, idnum
    curr = start = ALLOCATE MEMORY
    DISPLAY "item name : "
    READ item_name
    DISPLAY "Sup code : "
    READ item_id
    DISPLAY "donator : "
    READ donator
    DISPLAY "Shipment : "
    READ shipment
    DISPLAY "Qty : "
    READ quantity
    CALL id_read(idtype) FUNCTION
    DISPLAY "Donation ID is ", donate_code
    DISPLAY "Data Entry Succeed"
END DEFINE

DEFINE appendtoFile FUNCTION
    DECLARE curr OF donate, idcurr OF id, fpt
    OPEN donation.txt
    IF fpt = NULL THEN
        DISPLAY "Error when opening file"
    ELSE
```

```
        PRINT TO FILE, item_name,item_id,donator,shipment,quantity,donate_code
        NEXT curr = NULL
        IF WRITING NOT = 0 THEN
            DISPLAY "ID has been generated succesfully"
        ELSE
            DISPLAY "Error when writing"
        ENDIF
    ENDIF
    CLOSE donation.txt FILE
END DEFINE


DEFINE writetoFile FUNCTION
    DECLARE curr OF donate, idcurr OF id, fpt
    OPEN donation.txt
    IF fpt = NULL THEN
        DISPLAY "Error when opening file"
    ELSE
        PRINT TO FILE, item_name,item_id,donator,shipment,quantity,donate_code
        NEXT curr = NULL
        IF WRITING NOT = 0 THEN
            DISPLAY "ID has been generated succesfully"
        ELSE
            DISPLAY "Error when writing"
```

```
        ENDIF
    ENDIF
    CLOSE donation.txt FILE
END DEFINE


DEFINE search_id FUNCTION
    DECLARE start, curr OF donate, id, flag
    start = CALL read_LinkedList
    curr = start
    DISPLAY "Input Donation ID "
    READ id
    DO WHILE curr NOT = NULL
        IF id = donate_id THEN
            DISPLAY "Match is Found"
            DISPLAY "Item name, Sup code, Donator, Shipment, Quantity, Donation ID"
            flag = 1
            EXIT LOOP
        ENDIF
        NEXT curr
    END DO
    IF flag = 0 THEN
        DISPLAY "Item not Found"
    ENDIF
```

```
END DEFINE


DEFINE donation_add FUNCTION
    DECLARE start, curr OF donate, id, flag, quant_add
    start = CALL read_LinkedList
    curr = start
    DISPLAY "Input Donation ID "
    READ id
    DO WHILE curr NOT = NULL
        IF id = donate_id THEN
            DISPLAY "Match is Found"
            DISPLAY "Item name, Sup code, Donator, Shipment, Quantity, Donation ID"
            flag = 1
            DISPLAY "Qty to be added : "
            READ quant_add
            quantity = quantity + quant_add
            shipment = shipment + 1
            DISPLAY "New Qty is ", quantity
            EXIT LOOP
        ENDIF
        NEXT curr
    END DO
    IF flag = 1 THEN
```

```
        DECLARE fpt
        OPEN donation.txt
        IF fpt = NULL THEN
            DISPLAY "Error when opening file"
        ENDIF
        curr = start
        DO WHILE curr NOT = NULL
            PRINT TO FILE item_name,item_id,donator,shipment,quantity,donate_code
        END DO
        IF WRITING NOT = 0 THEN
            DISPLAY "ID has been generated succesfully"
        ELSE
            DISPLAY "Error when writing"
        ENDIF
        CLOSE donation.txt FILE
    ELSE IF flag = 0 THEN
        DISPLAY "Item not Found"
    ENDIF
END DEFINE


DEFINE view_donation FUNCTION
    DECLARE no=0, curr OF donate
    DISPLAY "Showing data of donation.txt"
```

```
    DISPLAY "NO , Item Name, Supplier Code, Donator, Shipment, Quantity, Donation ID"
    curr = CALL read_LinkedList
    DO WHILE curr NOT = NULL
        no = no+1
        DISPLAY no, item_name,item_id, donator, shipment, quantity, donate_code
        NEXT curr
    END DO
END DEFINE


DEFINE read_LinkedList FUNCTION
    DECLARE fpt
    OPEN donation.txt FILE
    IF fpt = NULL THEN
        DISPLAY "Error when opening file"
    ENDIF
    DECLARE curr OF donate, tempstart OF donate,t,t2,t3,t4,t5,t6
    REPEAT
 READ FROM FILE t,t2,t3,t4,t5,t6
        IF tempstart = NULL THEN
            tempstart = curr = ALLOCATE MEMORY
        ELSE
            NEXT curr = ALLOCATE MEMORY
            curr = NEXT curr
```

```
        ENDIF
        item_name = t
        item_id = t2
        donator = t3
        shipment = t4
        quantity = t5
        donate_id = t6
        NEXT curr = NULL
    UNTIL REACH END OF FILE
    CLOSE donation.txt
    RETURN tempstart
END DEFINE


DEFINE view_distribution FUNCTION
    DECLARE no=0, curr OF distribute
    DISPLAY "Showing data of dist.txt"
    DISPLAY "NO , Item Name, Donation ID, Hospital, Distributed Qty, Distribution ID"
    curr = CALL read_LinkedList1
    DO WHILE curr NOT = NULL
        no = no+1
        DISPLAY no, distributed_item,donation_did, hospital, distributed_quant, distribute_id
        NEXT curr
    END DO
```

```
END DEFINE


DEFINE read_LinkedList1 FUNCTION
   DECLARE fpt
   OPEN dist.txt FILE
   DECLARE curr OF distribute, tempstart OF distribute,t,t2,t3,t5,t6
  REPEAT
       READ FROM FILE t,t2,t3,t5,t6
      IF tempstart = NULL THEN
         tempstart = curr = ALLOCATE MEMORY
      ELSE
         NEXT curr = ALLOCATE MEMORY
         curr = NEXT curr
      ENDIF
      distributed_item = t
      donation_did = t2
      hospital = t3
      distributed_quant = t5
      distribute_id = t6
      NEXT curr = NULL
   UNTIL REACH END OF FILE
   CLOSE dist.txt FILE
   RETURN tempstart
```

```
END DEFINE


DEFINE distribution_add FUNCTION
    DECLARE start, curr OF donate, id, flag=0,hospital, quant_dist, idcurr OF id
    start = CALL read_LinkedList
    curr = start
    DISPLAY "Input Donation ID of item to be distributed "
    READ id
    DO WHILE curr NOT = NULL
        IF id = donate_id THEN
            DISPLAY "Match is Found"
            DISPLAY "Donation ID, Item Name, Current Quantity"
            DISPLAY "Qty to be distributed : "
            READ quant_dist
            IF quantity <= quant_dist THEN
                flag = 2
                DISPLAY "Quantity Exceed Stock!"
            ELSE
                flag = 1
                DISPLAY "Distribution Target : "
                READ hospital
                quantity = quantity - quant_dist
                DISPLAY "New qty is",quantity
```

```
            CALL id_update FUNCTION
            idcurr = CALL id_read FUNCTION
            DISPLAY "Distribution Code is",distribution_code
            DECLARE dispt
            OPEN dist.txt as mode
            IF dispt = NULL THEN
                DISPLAY "file open error"
            ELSE
                PRINT TO FILE item_name, donate_id, hospital, quant_dist, distribute_code
                IF WRITING NOT = 0 THEN
                    DISPLAY "Distribution Record has been updated succesfully"
                ELSE
                    DISPLAY "Error when writing"
                ENDIF
            ENDIF
            CLOSE dist.txt FILE
        END IF
        EXIT LOOP
    ENDIF
    NEXT curr
END DO
IF flag = 1 THEN
    DECLARE fpt
```

```
        OPEN donation.txt
        IF fpt = NULL THEN
            DISPLAY "Error when opening file"
        ELSE
            curr = start
            DO WHILE curr NOT = NULL
                PRINT TO FILE item_name,item_id,donator,shipment,quantity,donate_code
                NEXT curr
            END DO
            IF WRITING NOT = 0 THEN
                DISPLAY "Record has been Updated succesfully"
            ELSE
                DISPLAY "Error when writing"
            ENDIF
        ENDIF
        CLOSE donation.txt FILE
    ELSE IF flag = 0 THEN
        DISPLAY "Item not Found"
    ENDIF
END DEFINE


DEFINE update_data FUNCTION
    DECLARE ans
```

```
DO WHILE ans NOT = 4
    DISPLAY "1. add quantity"
    DISPLAY "2. distribute item"
    DISPLAY "3. input new distribution"
    DISPLAY "4. exit"
    DISPLAY Choose an option
    READ ans
    IF ans = 1
        CALL donation_add
        EXIT LOOP
    ELSE IF ans = 2
        DECLARE mode = "a"
        CALL distribution_add(mode)
        EXIT LOOP
    ELSE IF ans = 3
        DECLARE mode = "w"
        CALL distribution_add(mode)
        EXIT LOOP
    ELSE IF ans = 4
        EXIT LOOP
    DEFAULT
        DISPLAY “INVALID CHOICE”
END DO
```

```
END DEFINE


DEFINE view_all_sorted FUNCTION
    DECLARE no=0, curr OF donate, dcurr OF distribute, tempstart OF distribute
    curr = CALL read_LinkedList FUNCTION
    CALL bubbleSort_Donation(curr) FUNCTION
    dcurr = CALL read_LinkedList1 FUNCTION
    CALL bubbleSort_Distribute(dcurr) FUNCTION
    DISPLAY "Displaying Sorted Donation and Distributed Record"
    DO WHILE CURR NOT = NULL
        DISPLAY "NO, item name, Supply code, donator, shipment, quantity, donation id"
        DISPLAY no, item_name, item_id, shipment, quantity, donate_code
        no = no + 1
        tempstart = dcurr
        REPEAT
            IF donate_id = donation_did THEN
                DISPLAY distributed_quant, hospital, distribute_id
            ENDIF
            tempstart = NEXT tempstart
        UNTIL tempstart = NULL
        curr = NEXT curr
    END DO
END DEFINE
```

```
DEFINE bubbleSort_Donation FUNCTION
   DECLARE swapped, curr OF donate, tempcurr OF donate
   IF start = NULL THEN
      DISPLAY "Record is Empty"
   END IF
   DO WHILE swapped IS TRUE
      swapped = 0
      curr = start
      REPEAT
         IF quantity<NEXT quantity THEN
            CALL swap(curr, NEXT curr)
            swapped = 1
         ENDIF
         curr = NEXT curr
      UNTIL NEXT curr NOT = tempcurr
      tempcurr = curr
   END DO
END DEFINE


DEFINE bubbleSort_distribute FUNCTION
   DECLARE swapped, curr OF distribute, tempcurr OF distribute
   IF start = NULL THEN
```

```
        DISPLAY "Record is Empty"
    END IF
    DO WHILE swapped IS TRUE
        swapped = 0
        curr = start
        REPEAT
            IF distributed_quant<NEXT distribute_quant THEN
                CALL swap1(curr, NEXT curr)
                swapped = 1
            ENDIF
            curr = NEXT curr
        UNTIL NEXT curr NOT = tempcurr
        tempcurr = curr
    END DO
END DEFINE


DEFINE swap FUNCTION
    DECLARE temp,itemtemp,shiptemp,itemidtemp,donatortemp,donateidtemp
    temp = x quantity
    x quantity = y quantity
    y quantity = temp
    itemtemp = x quantity
    x quantity = y quantity
```

```
    y quantity = itemtemp
    shiptemp = x quantity
    x quantity = y quantity
    y quantity = shiptemp
    itemidtemp = x quantity
    x quantity = y quantity
    y quantity = itemidtemp
    donatortemp = x quantity
    x quantity = y quantity
    y quantity = donatortemp
    donateidtemp = x quantity
    x quantity = y quantity
    y quantity = donateidtemp
END DEFINE

DEFINE swap1 FUNCTION
    DECLARE temp, hospitaltemp, distidtemp
    temp = x quantity
    x quantity = y quantity
    y quantity = temp
    hospitaltemp = x quantity
    x quantity = y quantity
    y quantity = hospitaltemp
```

```
    distidtemp = x quantity
    x quantity = y quantity
    y quantity = distidtemp
END DEFINE


BEGIN
    DECLARE opt
    DO WHILE opt NOT = 8
        DISPLAY "1. Input new donation"
        DISPLAY "2. Add donation Record"
        DISPLAY "3. View donation"
        DISPLAY "4. Update Record"
        DISPLAY "5. Search"
        DISPLAY "6. View distribution"
        DISPLAY "7. view sorted donation and distribution"
        DISPLAY "8. Exit"
        READ opt
        IF opt = 1 THEN
            CALL id_gen FUNCTION
            CALL inventory_new FUNCTION
            EXIT LOOP
        ELSE IF opt = 2 THEN
            CALL inventory_add FUNCTION
```

```
            EXIT LOOP
        ELSE IF opt = 3 THEN
            CALL view_donation FUNCTION
            EXIT LOOP
        ELSE IF opt = 4 THEN
            CALL search_id FUNCTION
            EXIT LOOP
        ELSE IF opt = 5 THEN
            CALL view_distribution FUNCTION
            EXIT LOOP
        ELSE IF opt = 6 THEN
            CALL view_all_sorted FUNCTION
            EXIT LOOP
        ELSE IF opt = 7 THEN
            CALL inventory_add FUNCTION
            EXIT LOOP
        ELSE IF opt = 8 THEN
            EXIT LOOP
        ELSE
            DISPLAY "INVALID CHOICE"
    END DO
END
```

Flow Chart

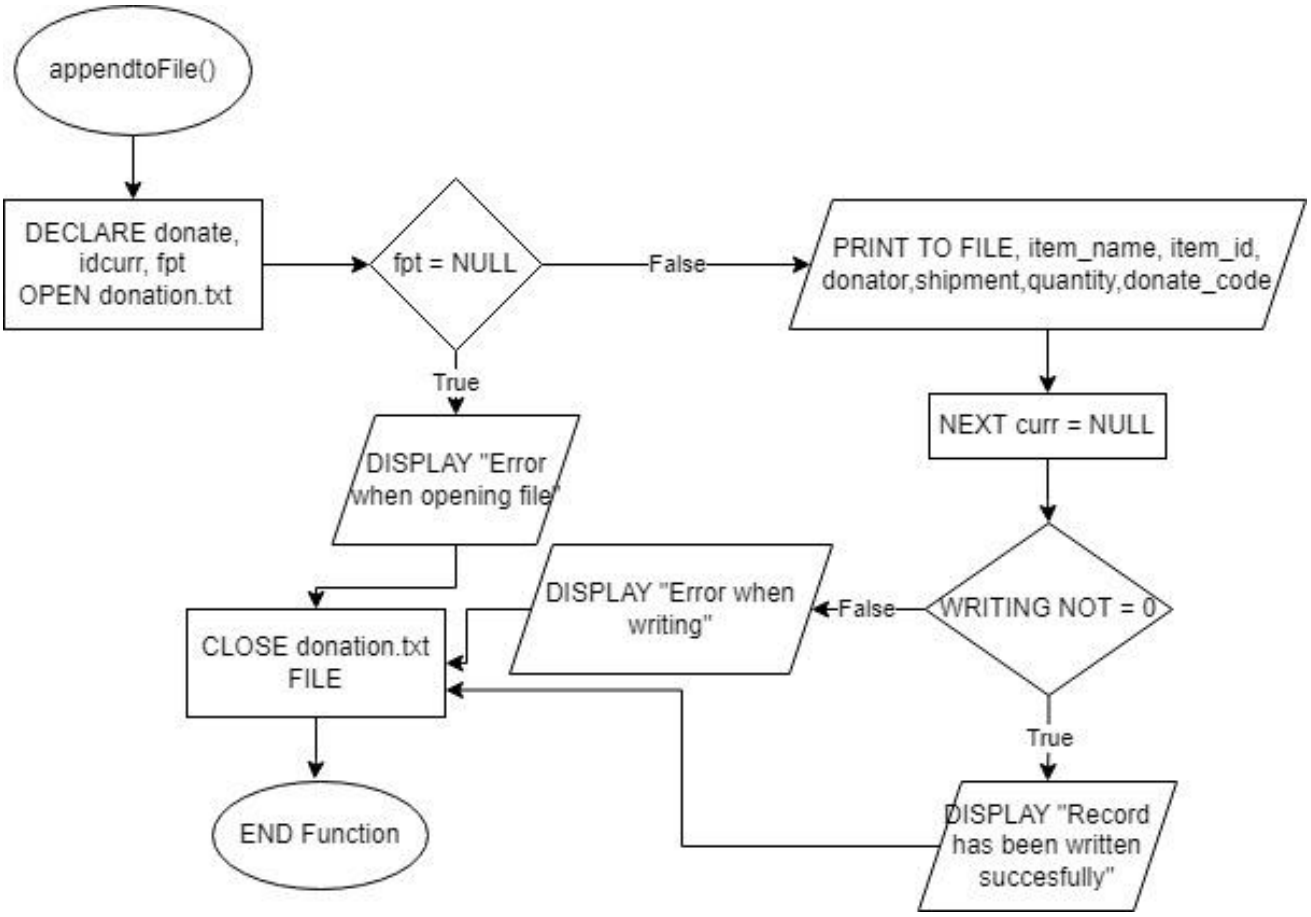

Figure 2.1 appendtoFile Function

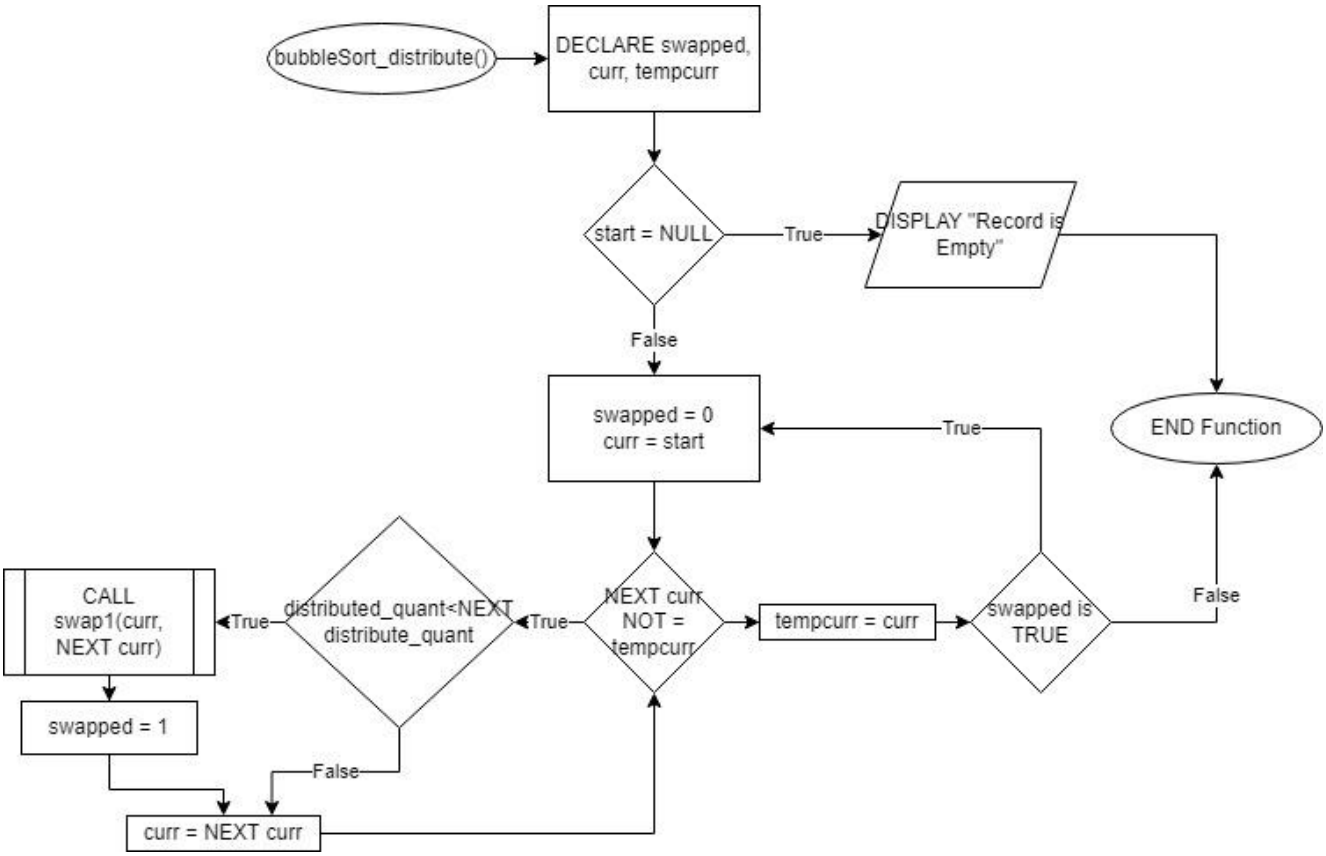

Figure 2.2 bubbleSort_distribute Function

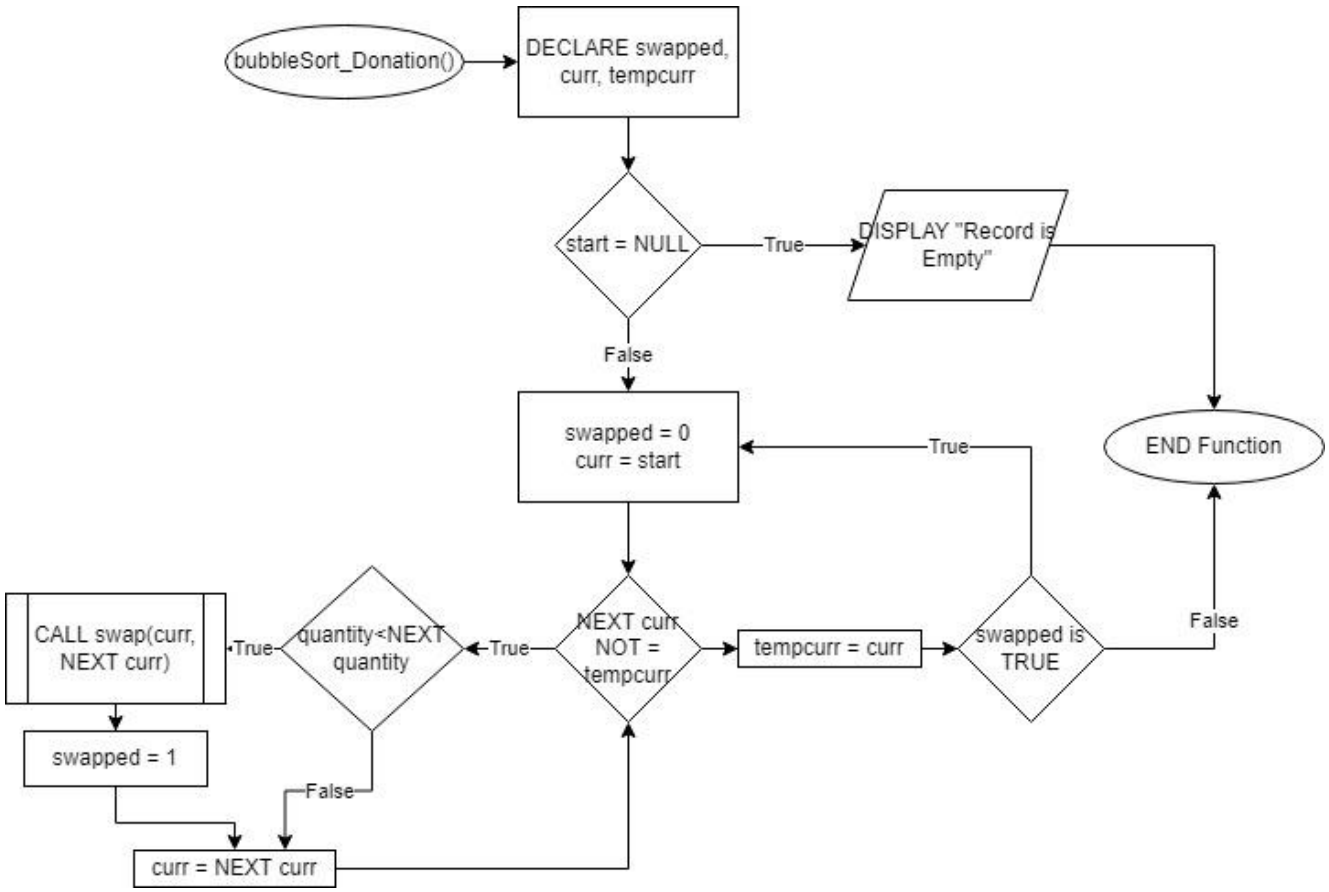

Figure 2.3 bubbleSort_Donation Function

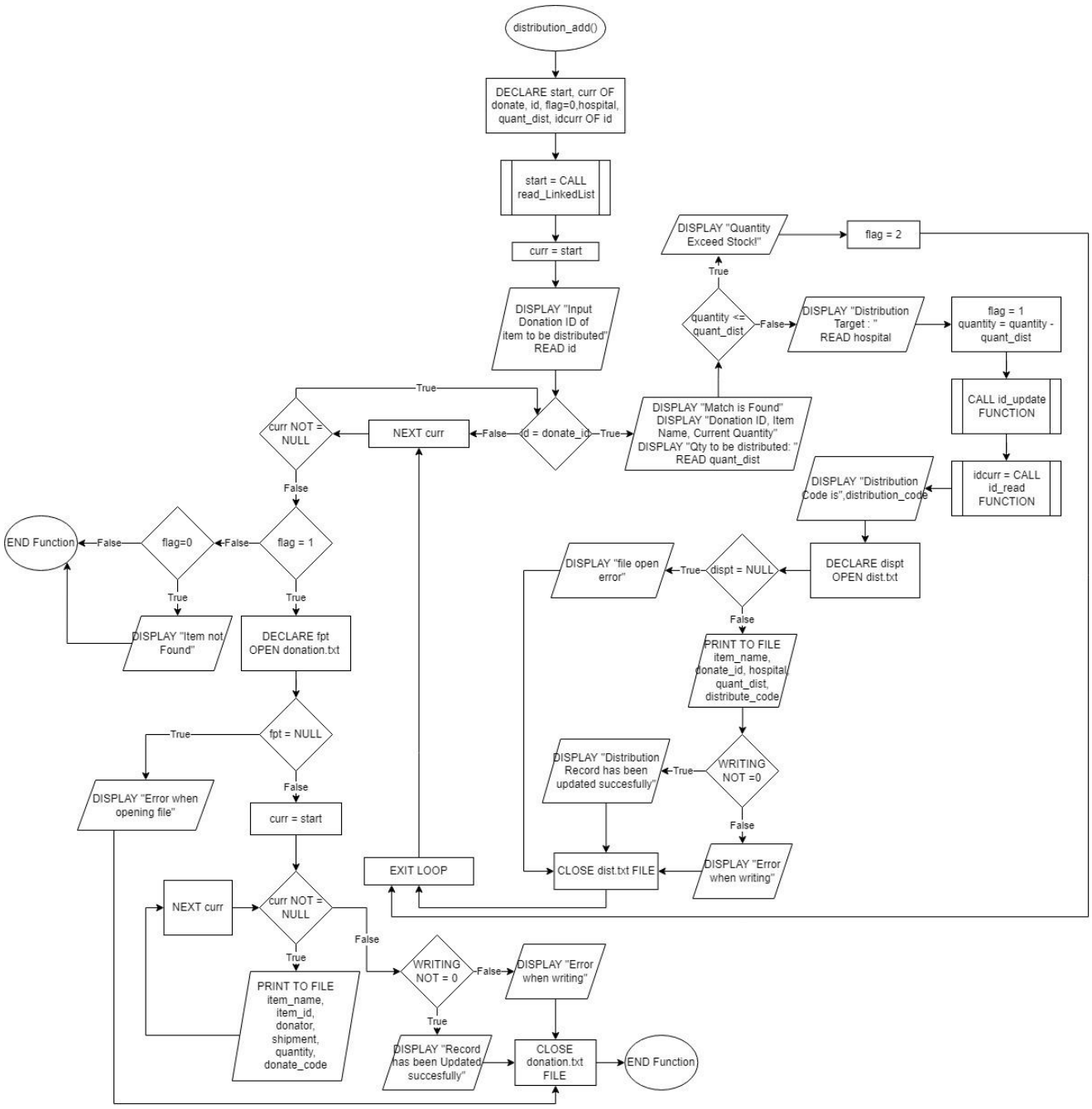

Figure 2.4 distribution_add Function

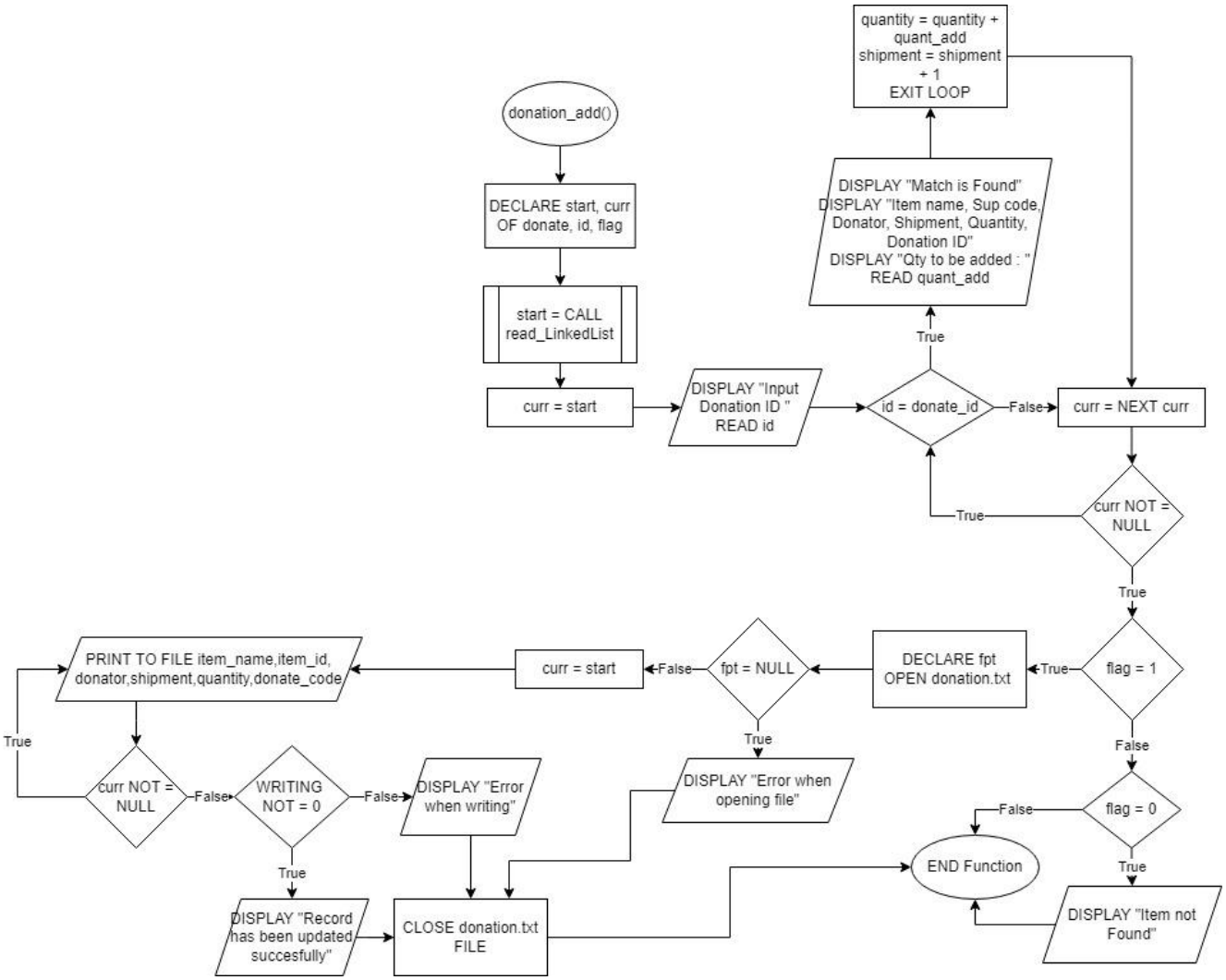

Figure 2.5 donation_add Function

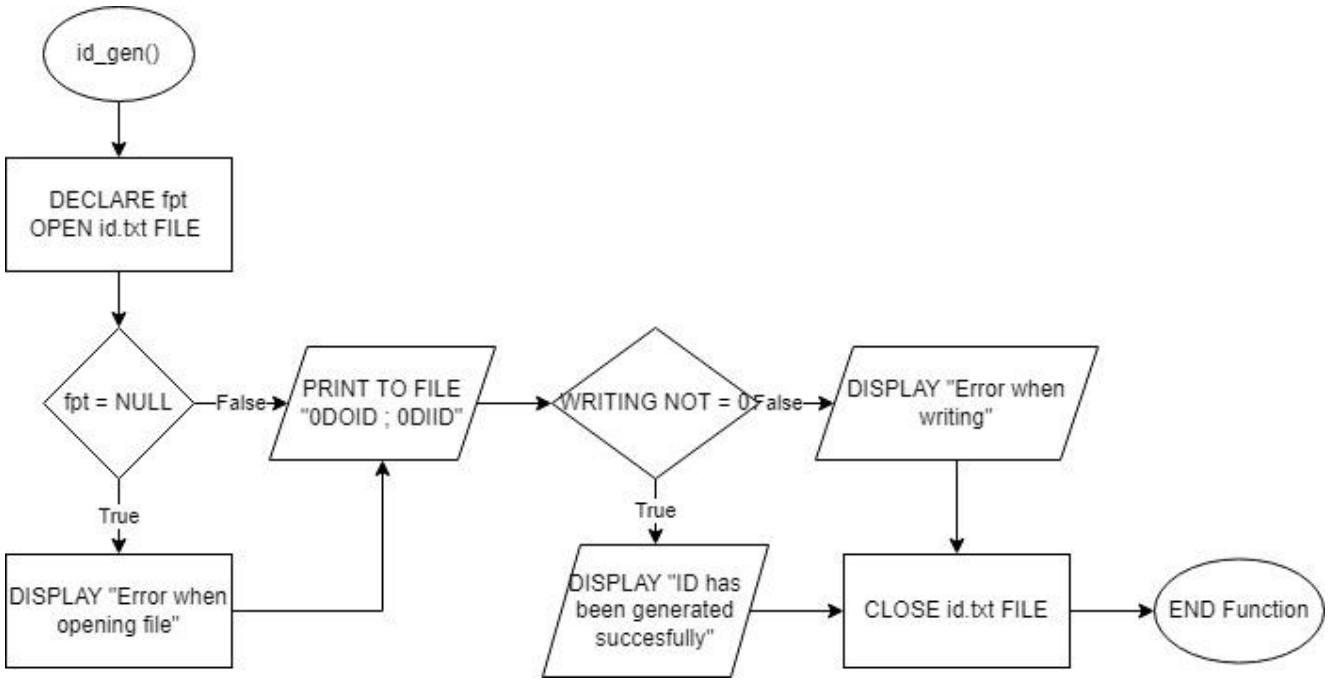

Figure 2.6 id_gen Function

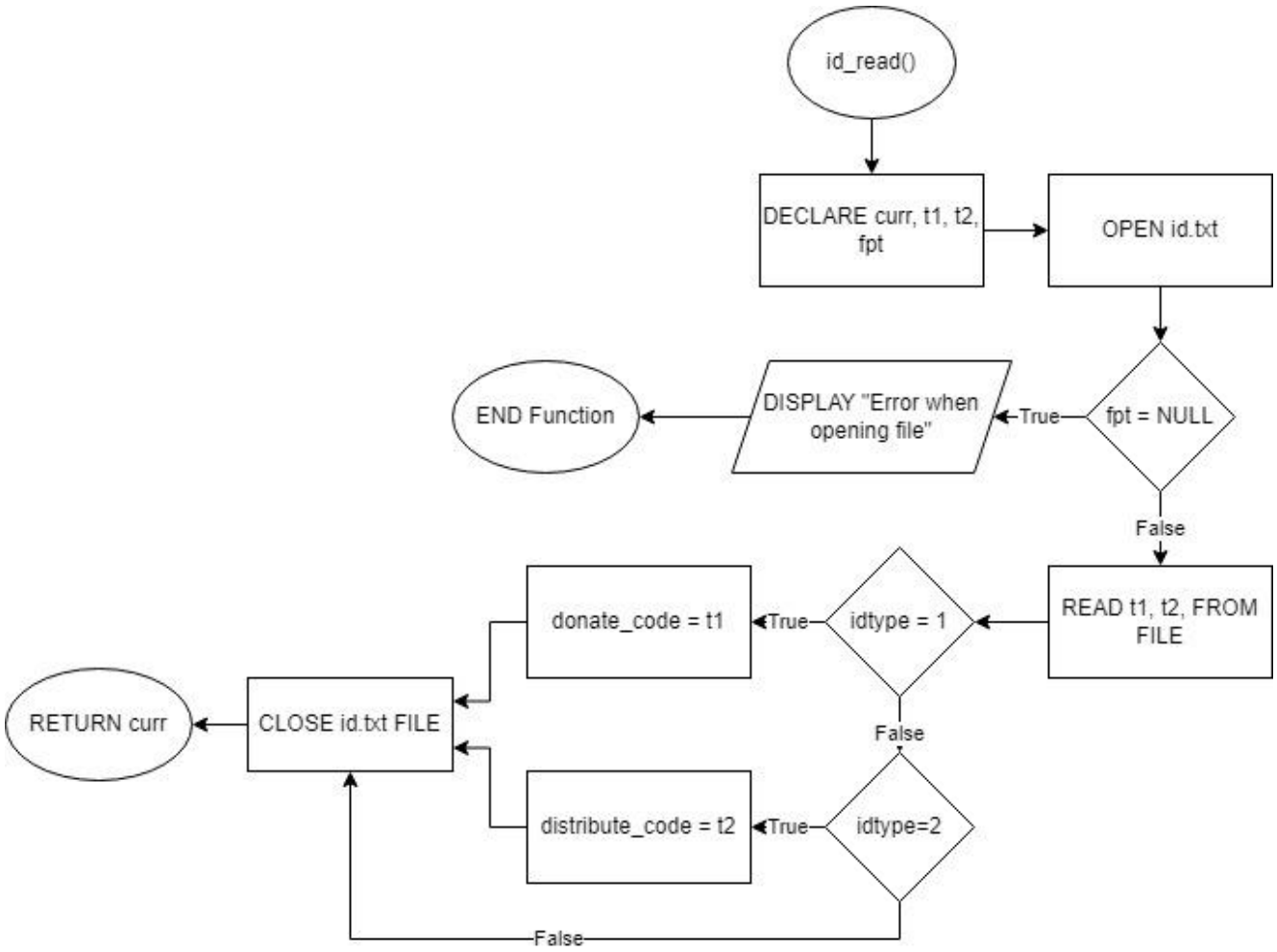

Figure 2.7 id_read Function

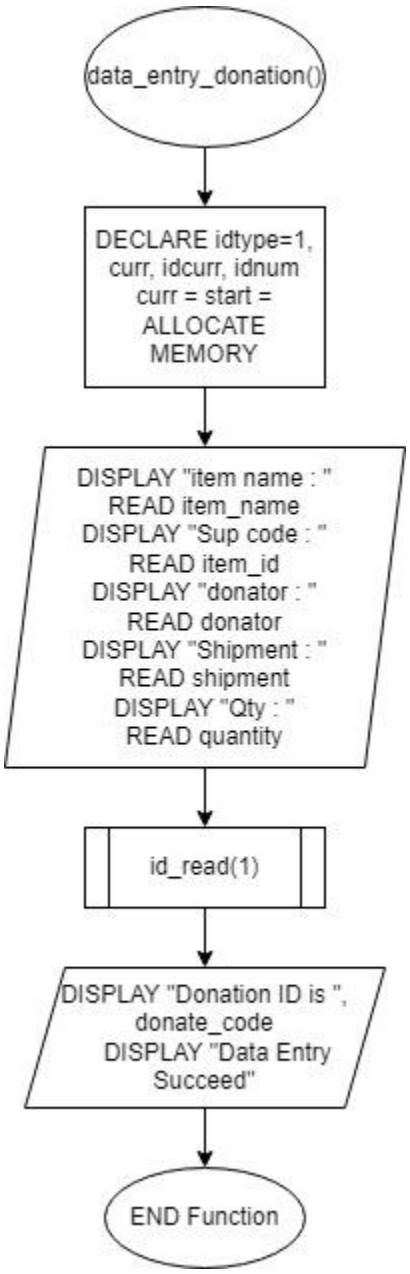

Figure 2.8 data_entry_donation Function

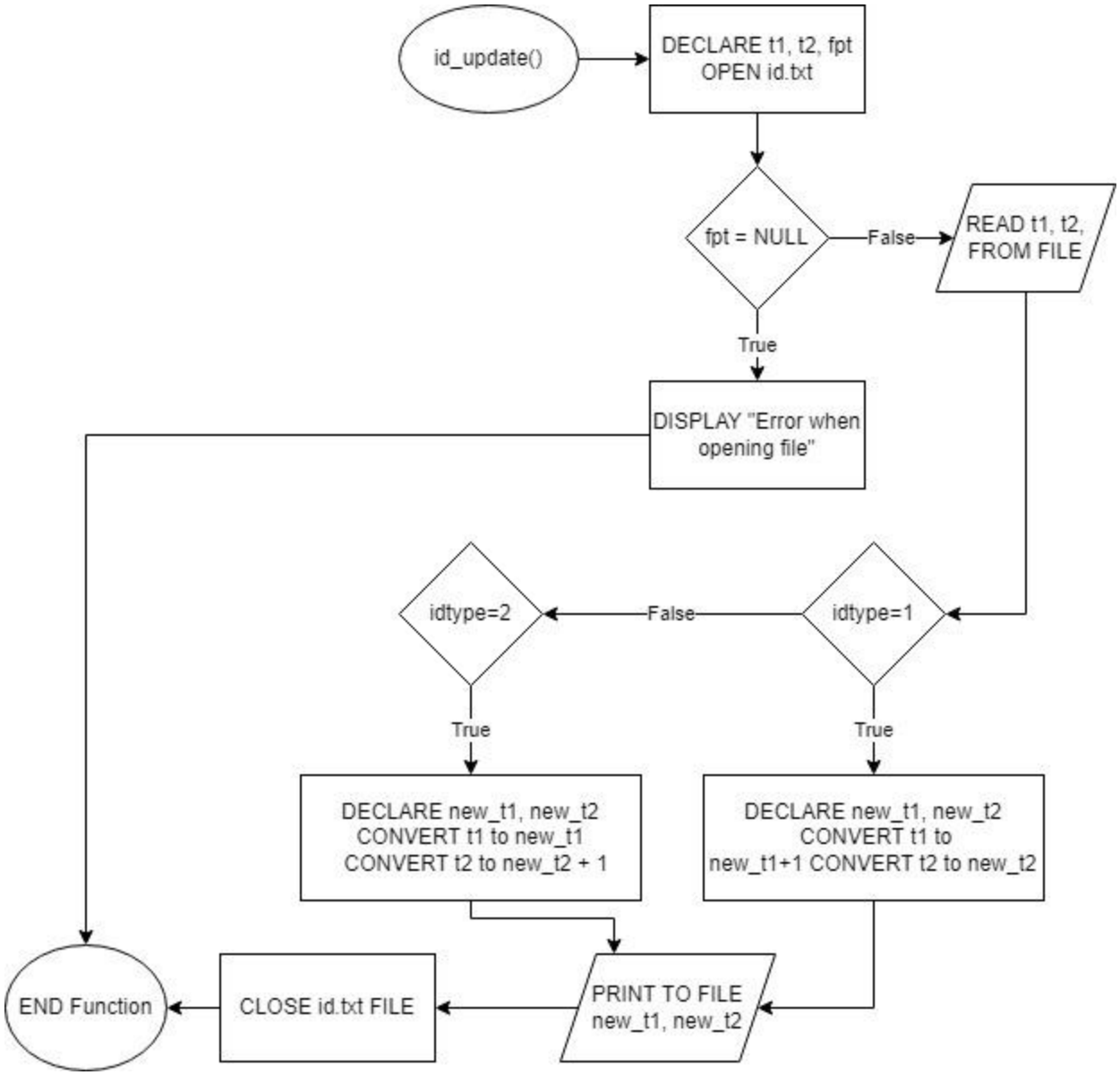

Figure 2.9 id_update Function

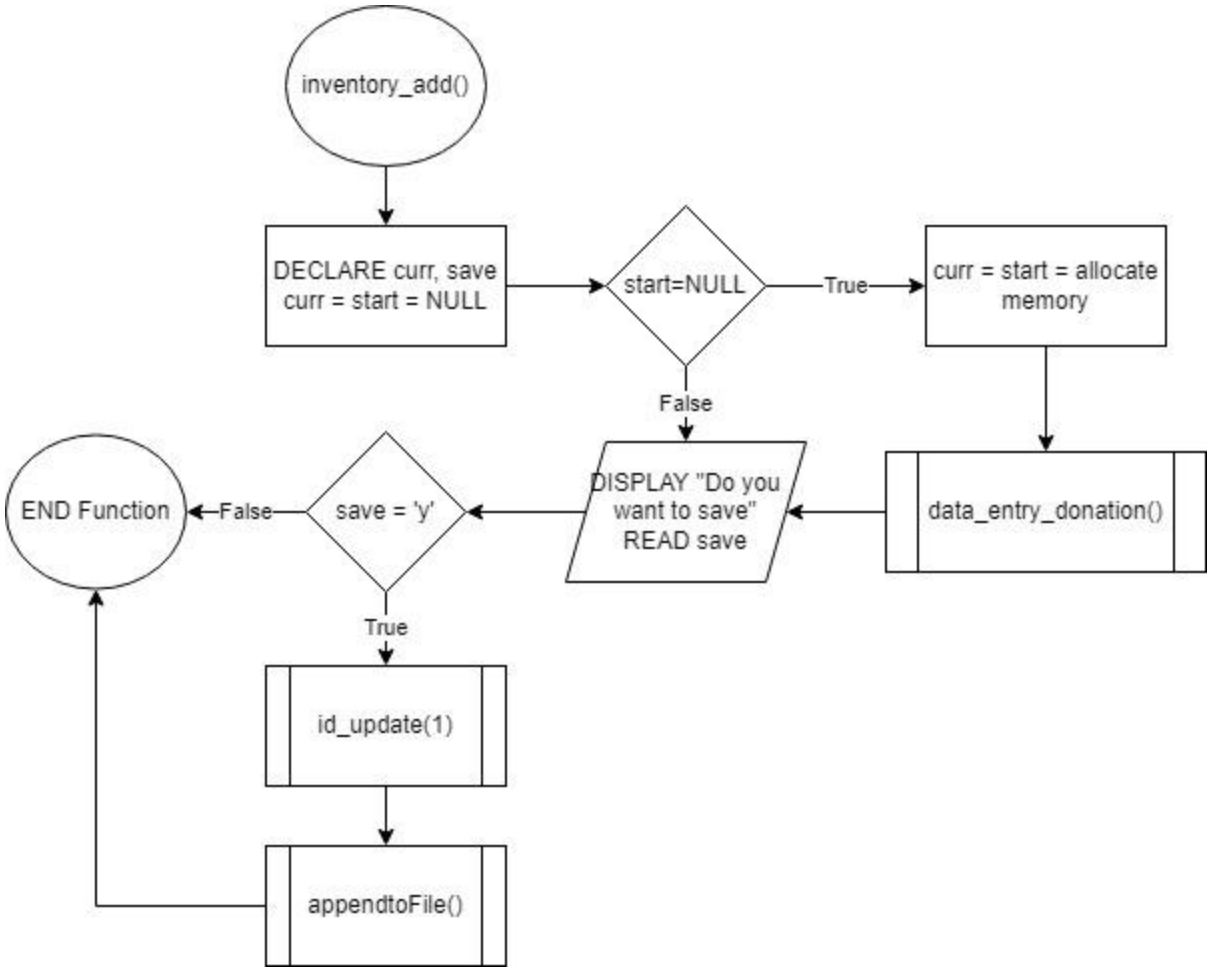

Figure 2.10 inventory_add Function

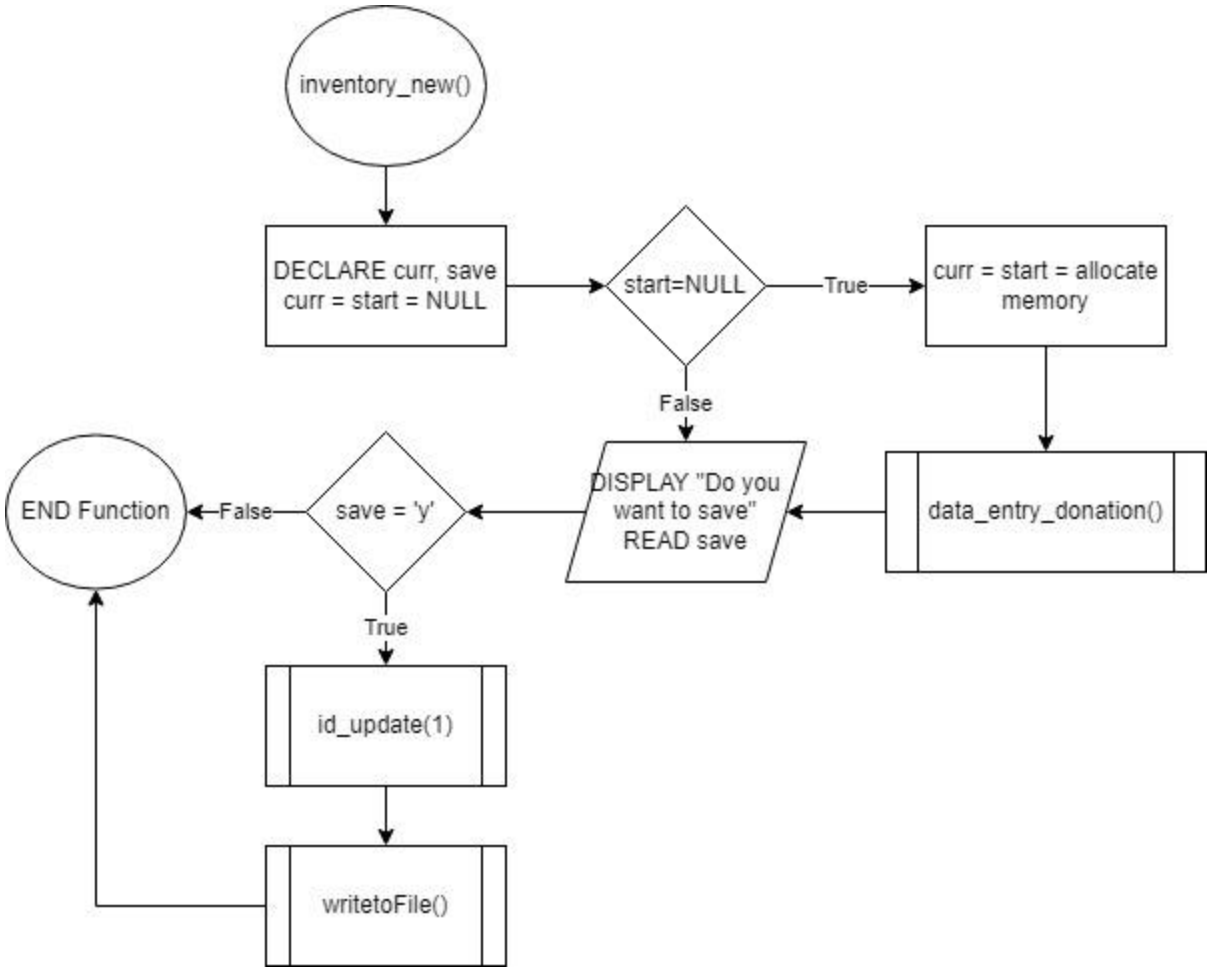

Figure 2.11 inventory_new Function

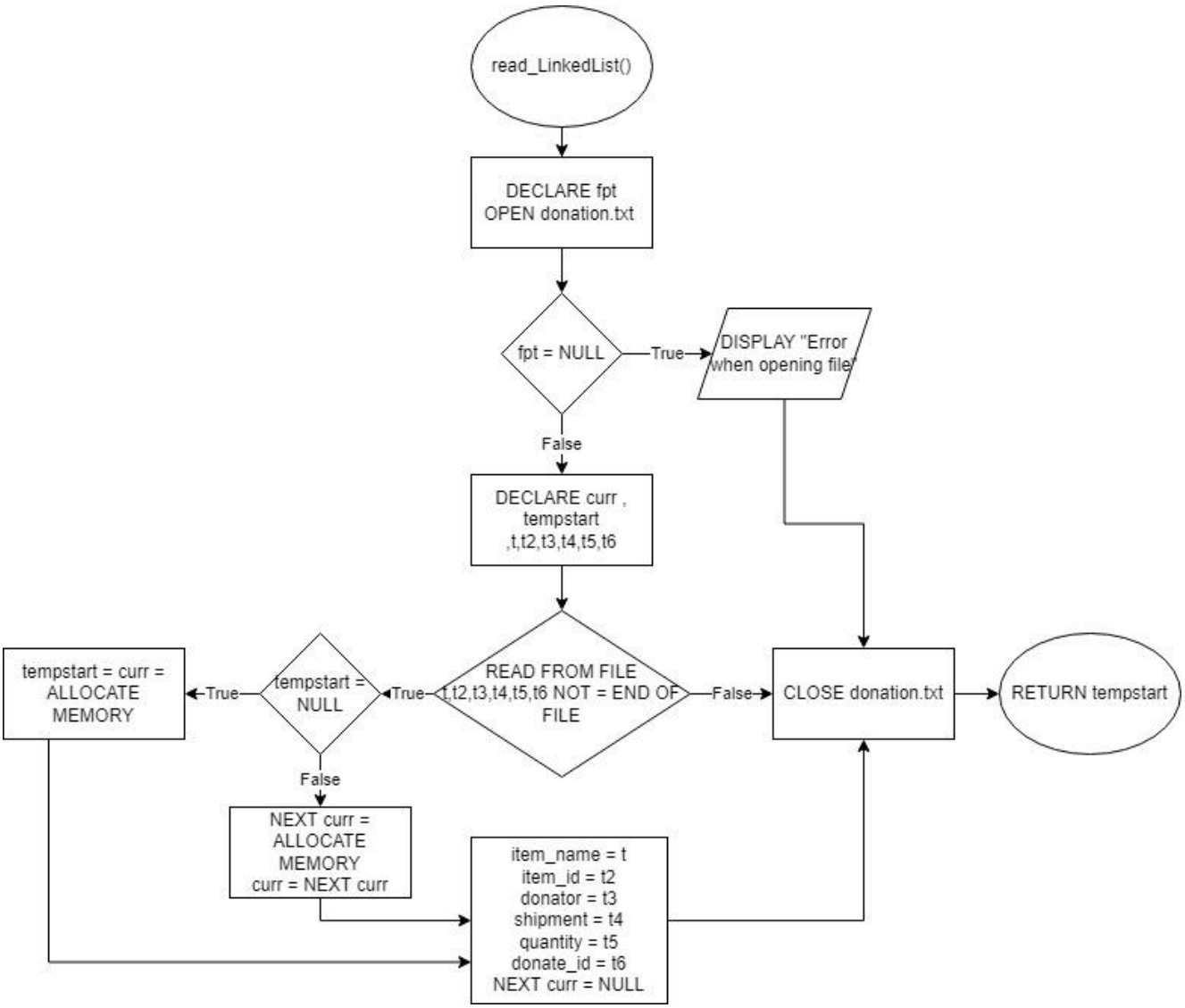

Figure 2.12 read_LinkedList Function

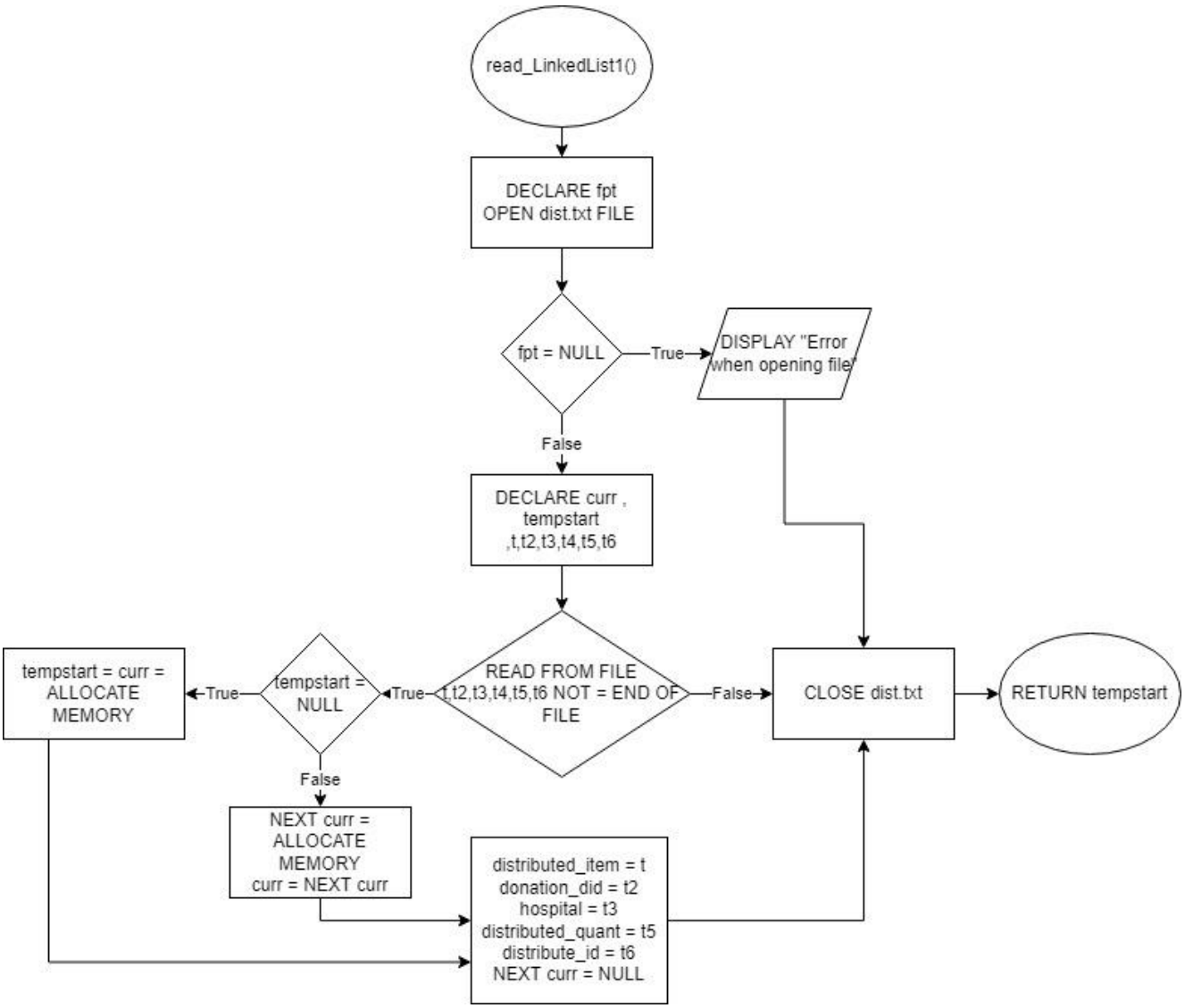

Figure 2.13 read_LinkedList1 Function

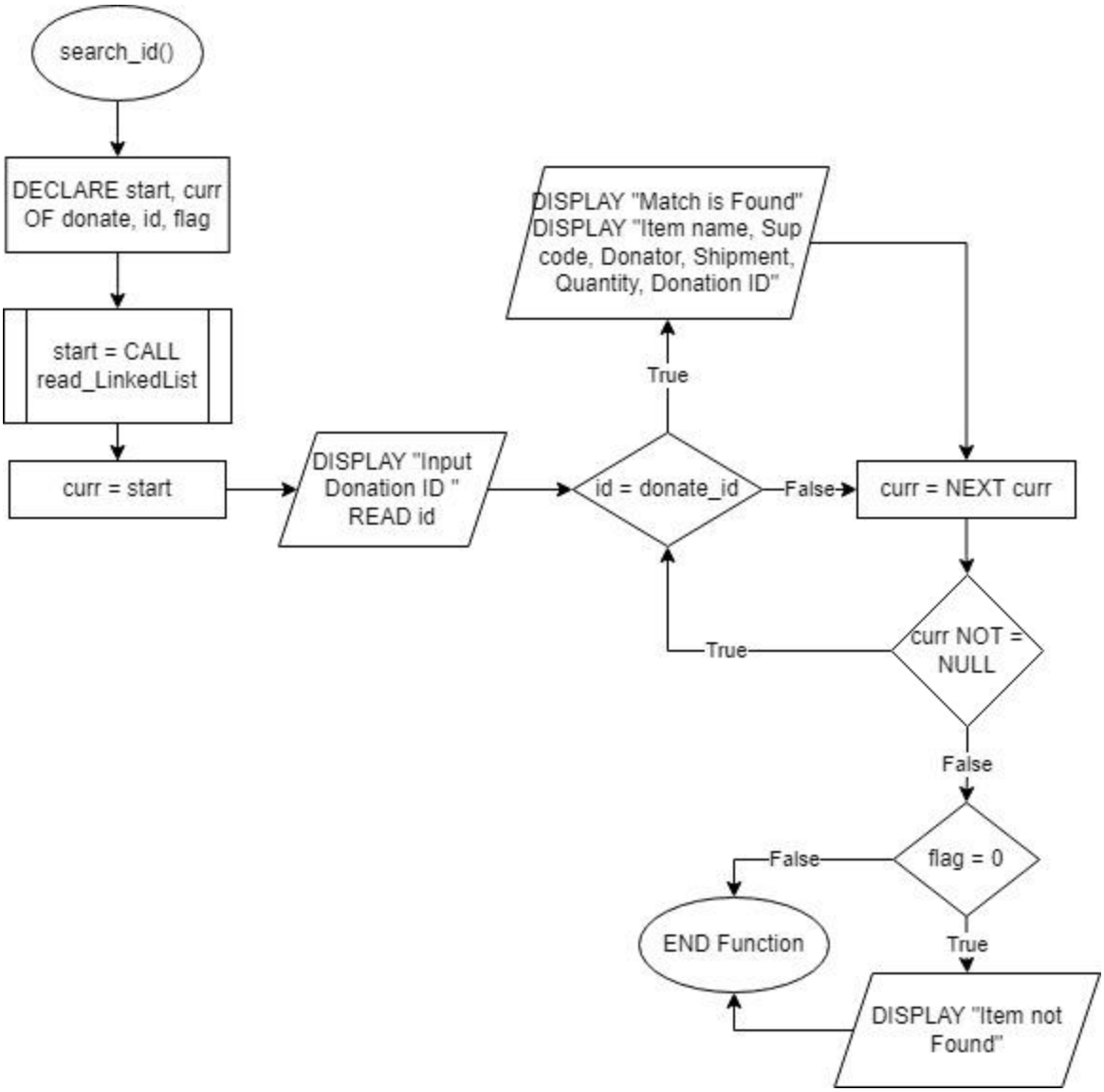

Figure 2.14 search_id Function

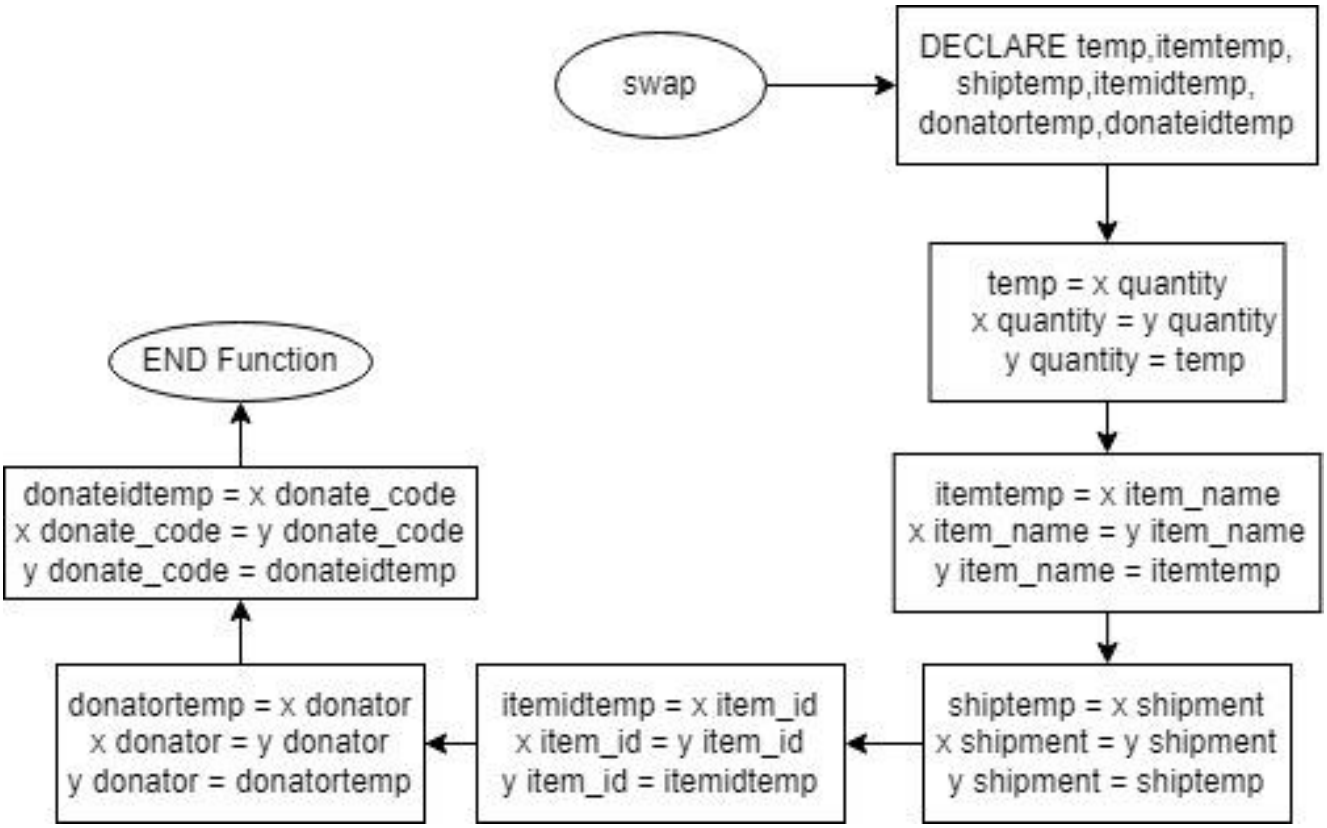

Figure 2.15 swap Function

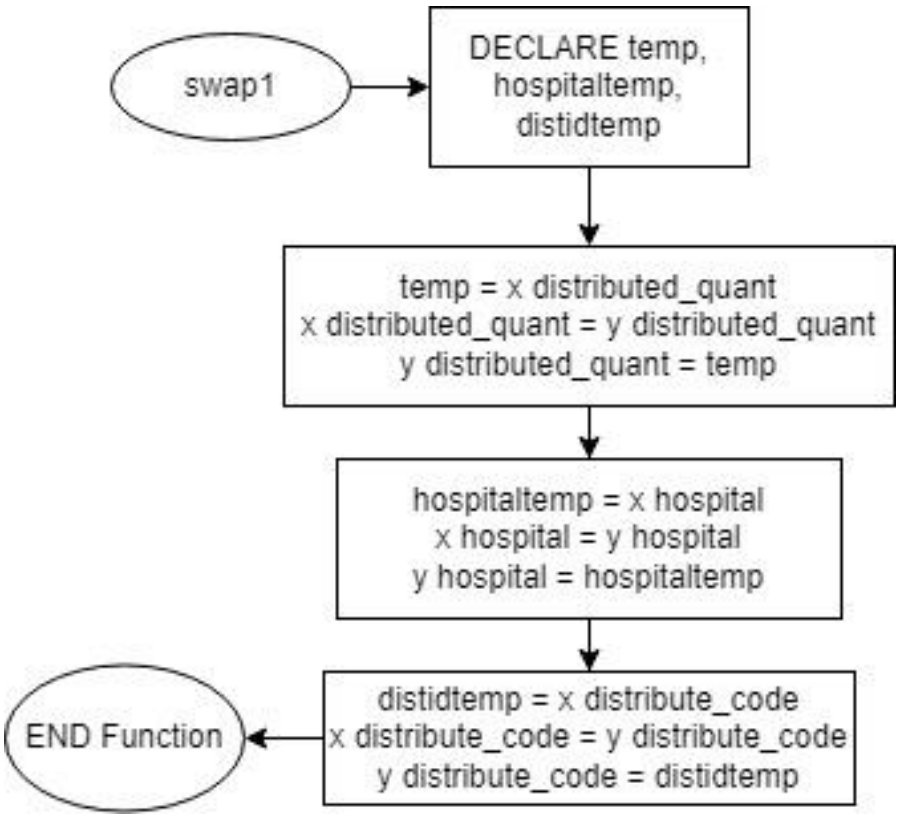

Figure 2.16 swap1 Function

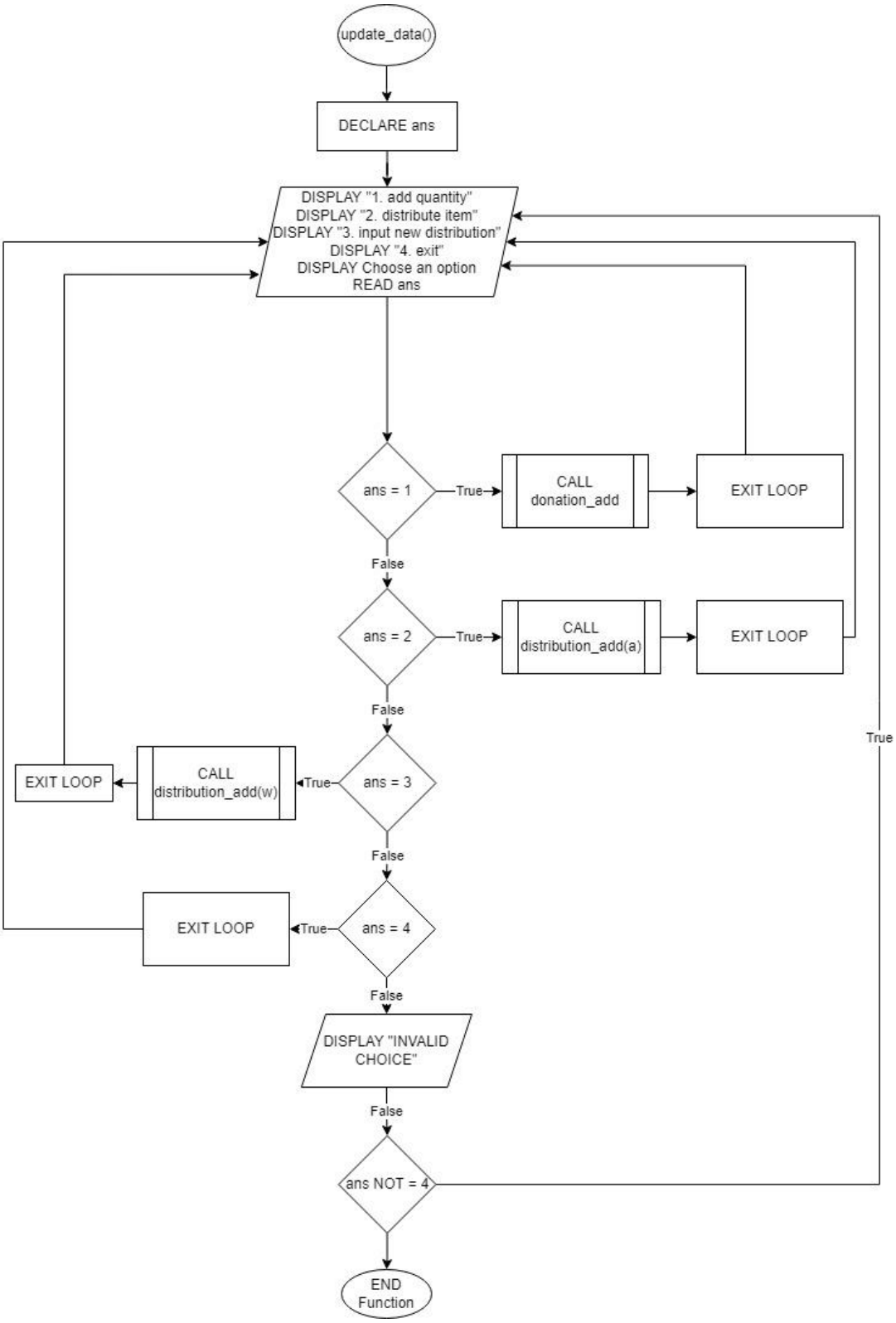

Figure 2.17 update_data Function

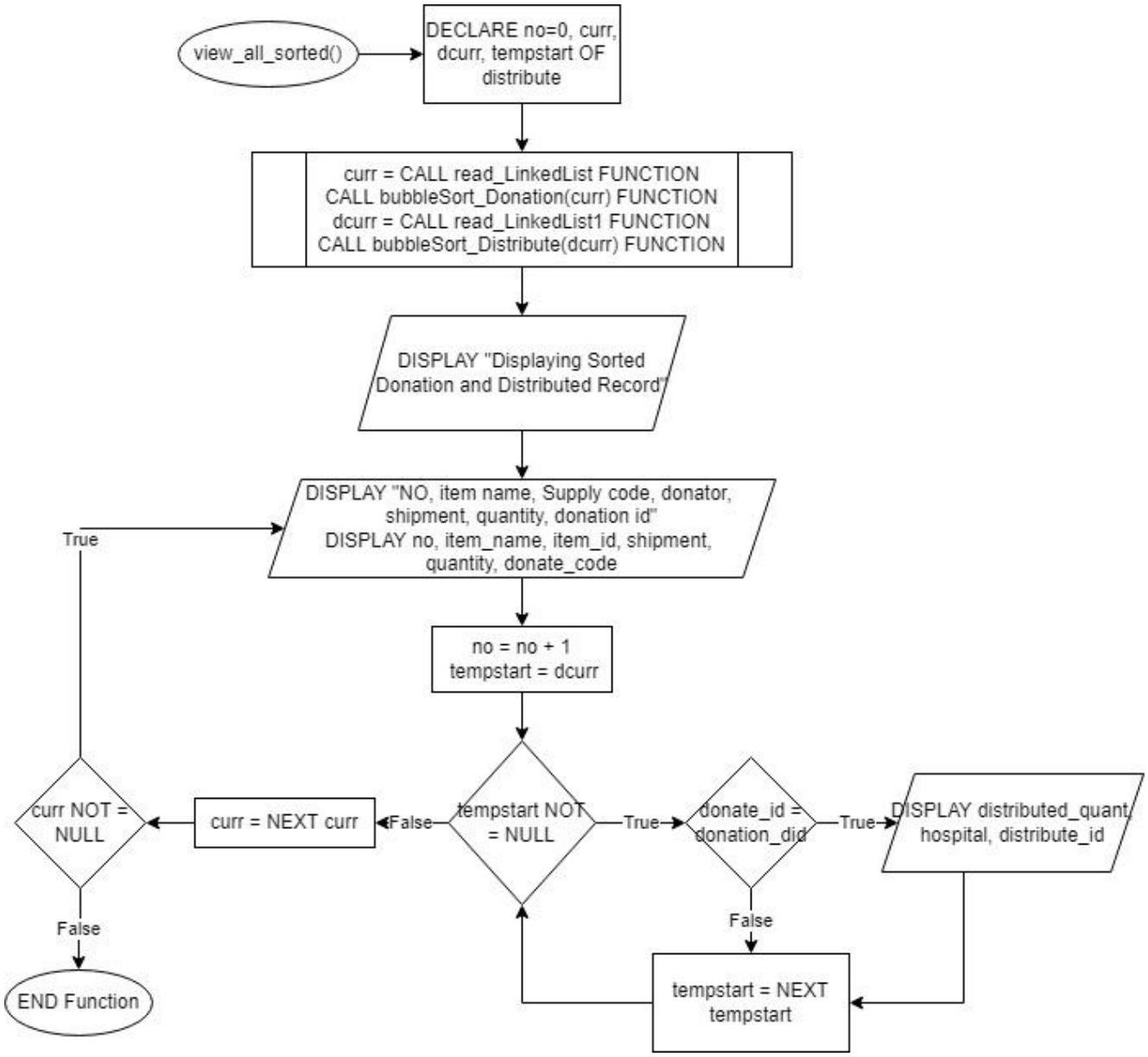

Figure 2.18 view_all_sorted Function

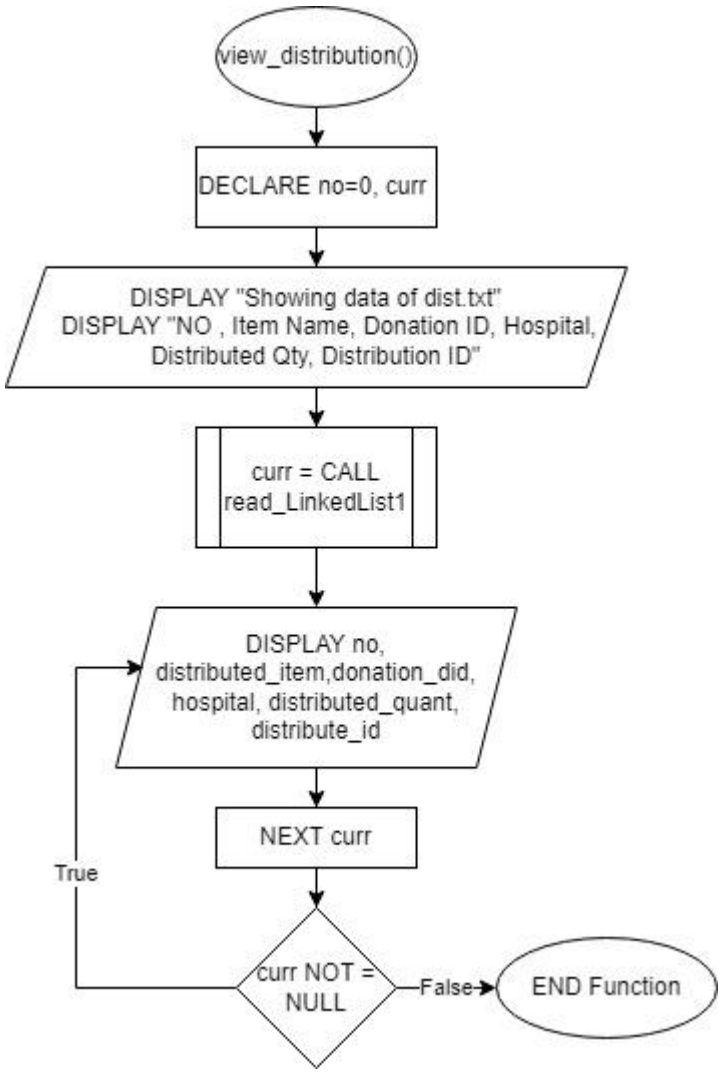

Figure 2.19 view_distribution Function

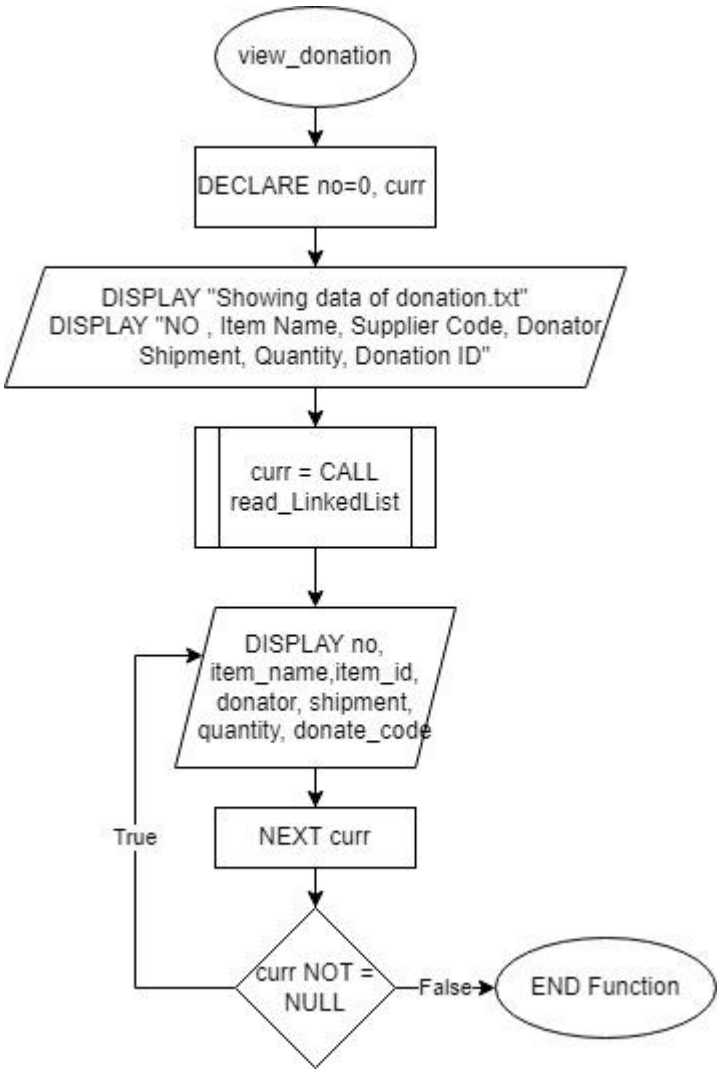

Figure 2.20 view_donation Function

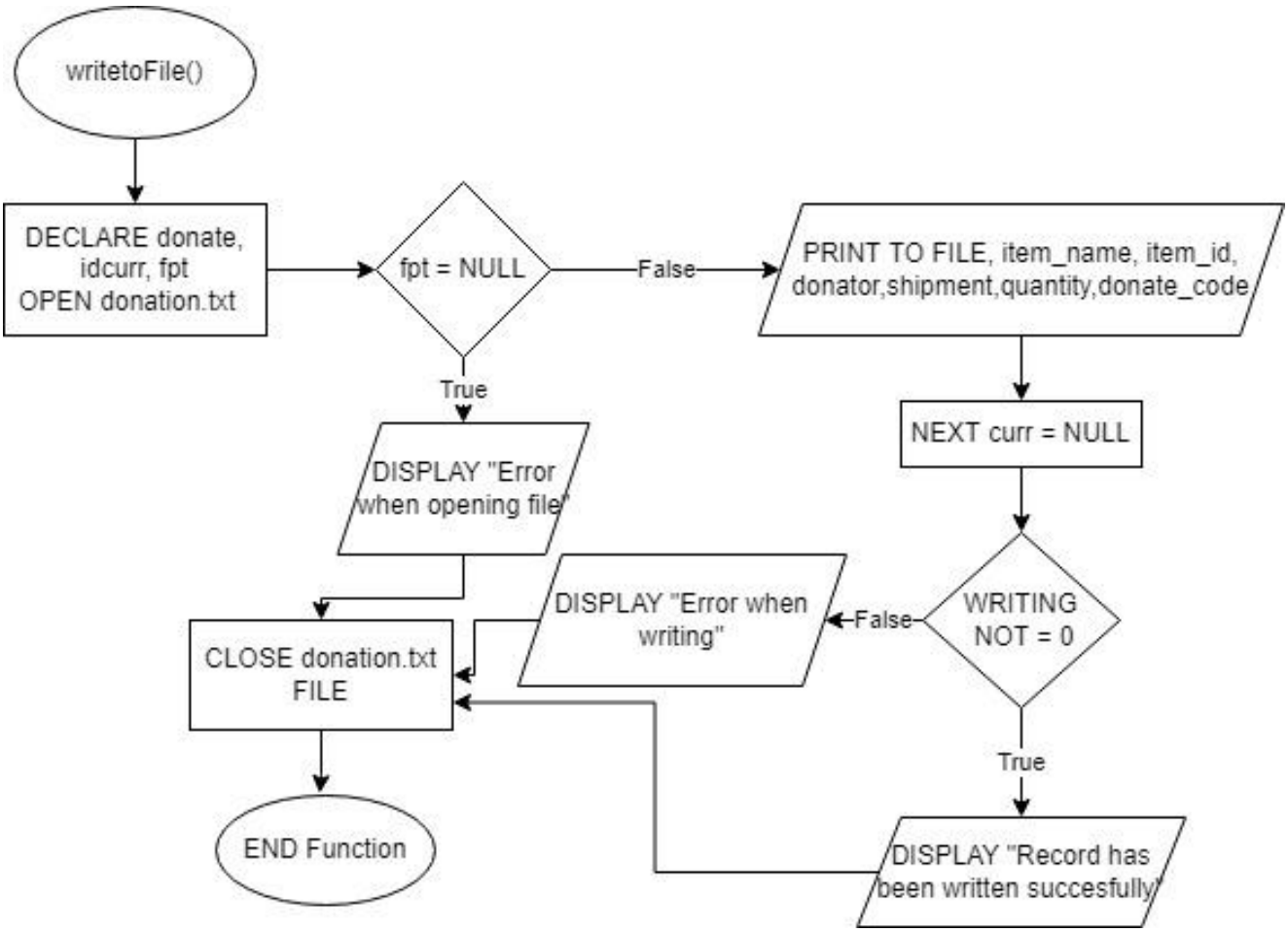

Figure 2.21 writetoFile Function

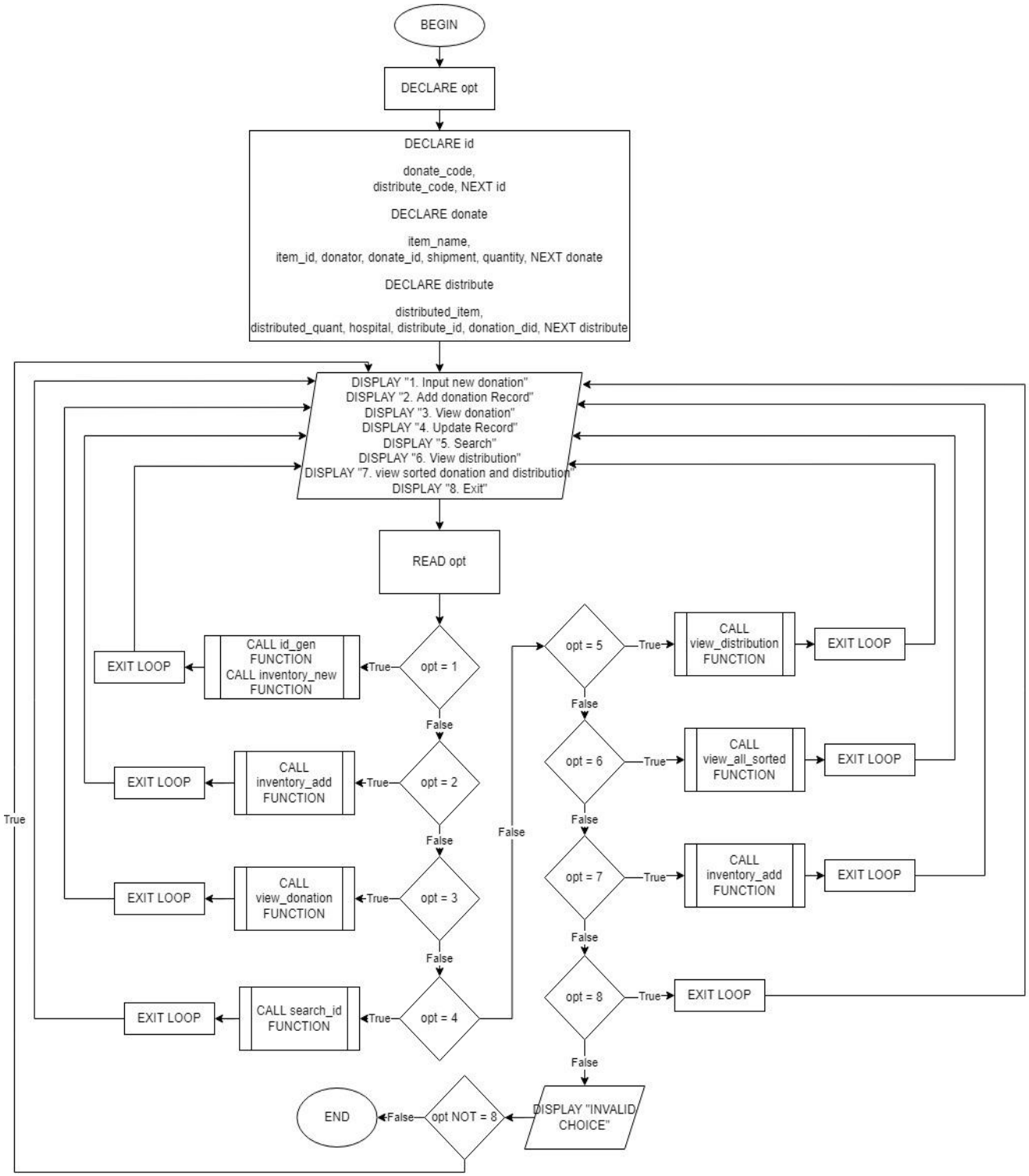

Figure 2.22 MAIN Function

3.)Additional Features

Aside from the main features of the donation management system, there is an additional feature added to ease the use of the system. This feature is called ID generation and update which is done by the id_gen, id_update, and id_read function. Id_gen Function will automatically generate a new ID for donation and distribution whenever users wish to start a new record. Whenever user add a new record , id_update function will cause the ID to change by adding 1 to the value of the ID and return it to the user by id_read to read the value.

4.)Sample Outputs

4.1 Main Menu

```
1. Input New Donation Record
2. Add Donation Record
3. View Donation Record
4. Update Donation Record
5. Search By Donation Code
6. View Distribution Record
7. View List of Donations and Distributions (Bubble Sorted by Quantity)
8. EXIT
Choose an option :
```

Figure 4.1.1) View of Main Menu

When the program is started for the first time, the program will show the main menu which is the way user may access the features of the program. Each of these features will be discussed within the next sample outputs.

4.2 Input new donation record and Add donation Record

<table>
<tr><th>SAMPLE INPUT</th><th>OUTPUT</th></tr>
<tr><td>

```
Choose an option : 1

 ID has been Generated succesfully!

Enter Item Name:
Contactless Thermometer
Supplier Code:
CT
Donate by:
Japan
No of Shipment:
1
Qty:
1.2
Donation ID : 1DOID
          Data Entry Succeed!
Do you want to save the file ? 'y' to continue
y

 Record has been written succesfully!
```

</td><td>

```
e code.cpp    donation.txt    dist.txt    id.txt
1   Contactless Thermometer ; CT ; Japan ; 1 ; 1.200000 ; 1DOID
2
```

</td></tr>
<tr><td>

```
Choose an option : 2

Enter Item Name:
Hand Sanitizers
Supplier Code:
HS
Donate by:
USA
No of Shipment:
1
Qty:
3.5
Donation ID : 2DOID
          Data Entry Succeed!
Do you want to save the file ? 'y' to continue
y

 Records has been saved succesfully!
```

</td><td>

```
e code.cpp    donation.txt    dist.txt    id.txt
1   Contactless Thermometer ; CT ; Japan ; 1 ; 1.200000 ; 1DOID
2   Hand Sanitizers ; HS ; USA ; 1 ; 3.500000 ; 2DOID
3
```

</td></tr>
</table>

When user choose the option 1, the program will prompt user for Item Name, Supplier Code, Donator, Number of Shipment, and Quantity to be stored in the file. When user want to save the file by inputting ‘y’, The data will be written into a file called donation.txt. By choosing option 1 user will delete the previous record and rewrite the new record into the file. To add a new record without deleting the previous record can be done by using the option 2 of the menu which is add Donation Record. Option 2 input and output are both similar to option 1 however it doesn’t rewrite the data in the file, instead it only adds new data to the file in the new line.

4.3 View donation record

SAMPLE INPUT

```
1. Input New Donation Record
2. Add Donation Record
3. View Donation Record
4. Update Donation Record
5. Search By Donation Code
6. View Distribution Record
7. View List of Donations and Distributions (Bubble Sorted by Quantity)
8. EXIT
Choose an option : 3
```

OUTPUT

```
Displaying All Data in donation.txt


==================================================================================================================================
NO      Item Name                       Supplier Code          Donator               Shipment              No.Quantity(Millions)          Donation ID
==================================================================================================================================
1       Contactless Thermometer         CT                     Japan                 1                     1.2                             1DOID
2       Hand Sanitizers                 HS                     USA                   1                     3.5                             2DOID
3       Face Mask                       FM                     China                 2                     120.0                           3DOID
4       Surgical Mask                   SM                     China                 2                     38.0                            4DOID
5       Oxygen Mask                     OM                     Saudi Arabia          2                     9.0                             5DOID
```

When user input 3 to the main menu, the terminal output will read and display all the data saved in the donation.txt file. The program will automatically show the records line by line in the order of they are saved in the file.

4.4 Update donation record

SAMPLE INPUT	OUTPUT
1. Input New Donation Record 2. Add Donation Record 3. View Donation Record 4. Update Donation Record 5. Search By Donation Code 6. View Distribution Record 7. View List of Donations and Distributions (Bubble Sorted by Quantity) 8. EXIT Choose an option : 4	1. Add to the quantity by Donation ID 2. Distribute the item by Donation ID 3. Input New Distribution List 4. Back To MENU Choice -->

The option 4 of the menu will show another menu which is required for the user to update the quantities in the inventory. The update donation menu allows users to add quantity to donation record, deduct quantity from donation record and create distribution records.

4.4.1.) Add quantity of donation

SAMPLE INPUT/OUTPUT	
TRUE INPUT/OUTPUT	**FALSE INPUT/OUTPUT**
Input Donation ID = 1DOID Match is found 1.Item Name : Contactless Thermometer 2.Supplier Code : CT 3.Donator : Japan 4.Shipment : 1 5.Quantity : 1.2 Millions 6.Donation ID : 1DOID Quantity to be added : 30 New Quantity is : 31.2 Millions Record has been updated succesfully!	1. Add to the quantity by Donation ID 2. Distribute the item by Donation ID 3. Input New Distribution List 4. Back To MENU Choice --> 1 Input Donation ID = 6DOID Item not found

Option 1 of this menu will allow user to add extra quantities to the item by the donation id generated by the program. If donation id is not found, program will produce a text that say that item not found.

4.4.2.) Distribute Item & New Distribution List

SAMPLE INPUT/OUTPUT	
INPUT	**OUTPUT**
1. Add to the quantity by Donation ID 2. Distribute the item by Donation ID 3. Input New Distribution List 4. Back To MENU Choice --> 3 Input Donation ID of Item to be Distributed = 3DOID	Match is found Donation ID : 3DOID Item Name : Face Mask Current Quantity : 120.0 Millions Quantity to be distributed : 20 Distribution/Hospital Target : Hospital ABC New Quantity is : 100.0 Millions Distribution ID is 2DIID Distribution Record has been updated succesfully! Donation Record has been updated succesfully!
1. Add to the quantity by Donation ID 2. Distribute the item by Donation ID 3. Input New Distribution List 4. Back To MENU Choice --> 2 Input Donation ID of Item to be Distributed = 10DOID	Choice --> 2 Input Donation ID of Item to be Distributed = 10DOID Item not found

Distribution happens in the option 2 and 3 of this menu. Option 3 will delete and rewritten a new list of distribution while option 2 will add a new record of distribution without deleting the previous record. In case of random donation id were inputted, the program will return a text that say item not found.

4.5 Search by donation code

INPUT	OUTPUT
1. Input New Donation Record 2. Add Donation Record 3. View Donation Record 4. Update Donation Record 5. Search By Donation Code 6. View Distribution Record 7. View List of Donations anc 8. EXIT Choose an option : 5 Input Donation ID = 1DOID	Choose an option : 5 Input Donation ID = 1DOID Match is found 1.Item Name : Hand Sanitizer 2.Supplier Code : HS 3.Donator : China 4.Shipment : 1 5.Quantity : 9.0 Millions 6.Donation ID : 1DOID
3. View Donation Record 4. Update Donation Record 5. Search By Donation Code 6. View Distribution Record 7. View List of Donations an 8. EXIT Choose an option : 9DOID	4. Update Donation Record 5. Search By Donation Code 6. View Distribution Record 7. View List of Donations and 8. EXIT Choose an option : 5 Input Donation ID = 9DOID Item not found

The option 5 show the search feature on this program. The search function will compare the ID given by users and the ID saved on the records. If match is found, then all the details of the particular ID will be shown on the other is not found then nothing is shown.

4.6 View distribution record

INPUT

```
5. Search By Donation Code
6. View Distribution Record
7. View List of Donations an
8. EXIT
Choose an option : 6
```

OUTPUT

```
Choose an option : 6

 Displaying All Data in dist.txt


 ==========================================================================================================================
 NO     Item Name                      Donation ID          Hospital             Distributed Quantity(Millions)  Distribution ID
 ==========================================================================================================================
 1      Hand Sanitizer                 1DOID                Hospital C           2.0                             2DIID
 2      Surgical Mask                  3DOID                Hospital C           4.0                             3DIID
```

The view distribution will show the list of all distribution in the order of they are created. This function read the data saved in the file and show it to the user on the output.

4.7 View sorted donation and distribution

INPUT

```
6. View Distribution Record
7. View List of Donations and Distributions (Bubble Sorted by Quantity)
8. EXIT
Choose an option : 7
```

OUTPUT

```
==========================================================================================================================================
NO     Item Name                      Supplier Code         Donator               Shipment              Quantity Stored(Millions)      Donation ID
==========================================================================================================================================
1      Contactless Thermometer        CT                    Japan                 3                     37.0                           2DOID

==========================================================================================
Qty Distributed       Hospital/Destination           Distribution ID
==========================================================================================



==========================================================================================================================================
NO     Item Name                      Supplier Code         Donator               Shipment              Quantity Stored(Millions)      Donation ID
==========================================================================================================================================
2      Surgical Mask                  SM                    Italy                 4                     30.5                           3DOID

==========================================================================================
Qty Distributed       Hospital/Destination           Distribution ID
==========================================================================================
2.0                   Hospital C                     2DIID



==========================================================================================================================================
NO     Item Name                      Supplier Code         Donator               Shipment              Quantity Stored(Millions)      Donation ID
==========================================================================================================================================
3      Hand Sanitizer                 HS                    China                 1                     9.0                            1DOID

==========================================================================================
Qty Distributed       Hospital/Destination           Distribution ID
==========================================================================================
4.0                   Hospital C                     3DIID
```

The option 7 will show the sorted list of donation and distribution. The list will be sort by the quantities start from the highest quantity to the lowest quantity. Distribution also sorted by the highest quantity distributed first and then the lowest quantity distributed last.

4.8 Exit program

INPUT

```
6. View Distribution Record
7. View List of Donations and
8. EXIT
Choose an option : 8
```

OUTPUT

```
5. Search By Donation Code
6. View Distribution Record
7. View List of Donations and Distributions (Bubble Sorted by Quantity)
8. EXIT
Choose an option : 8

C:\Users\MSI-INDO\OneDrive - Asia Pacific University\Desktop\ICP Assignment DEADLINE 10_8_2022\ICP A\ICP INDIVIDUAL ASSIGNMENT NICHOLAS SUN TP06
NICHOLAS SUN TP062907.exe (process 22620) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```

When user decided to exit the program, user may enter option 8 which will end the program.

5.)Conclusion

In conclusion, creating the inventory management system with C language has been a splendid example of understanding deeper on learning linked list data structure technique and on its language itself. This system has showed the features of C language that may adapt in some situations although may not be as efficient as other language, but it still helps to improve my algorithm thinking.

This inventory management system may not be perfect for the best use in business situation, but it still able to operate like the best inventory management system example that is made by the professional. Lastly, this system can still improve to gain better features and design for the user best experience while using it.

6.)References

Flowchart symbols - a complete guide. Zen Flowchart. (n.d.). Retrieved August 2, 2022, from https://www.zenflowchart.com/flowchart-symbols#:~:text=Off%2Dpage%20Connector%3A%20An%20off,target%20is%20on%20another%20page.&text=11.,process%20block%20is%20usually%20dashed.

Bose, S. (2021, February 2). *Coding standards and best practices to follow*. BrowserStack. Retrieved August 1, 2022, from https://www.browserstack.com/guide/coding-standards-best-practices