INDIVIDUAL ASSIGNMENT

TECHNOLOGY PARK MALAYSIA

CT010-3-1-PYP

PYTHON PROGRAMMING

LIANG HAN SHENG

TP067242

APD1F2111/APU1F2111 – CS(IS)

HAND OUT DATE: **10TH JANUARY 2022**

HAND IN DATE: **7TH MARCH 2022**

WEIGHTAGE: **100%**

INSTRUCTIONS TO CANDIDATES:

1. Submit your assignment online in MS Teams 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

Contents

Introduction

The Department oof Health needs a computer program to manage the inventory of PPEs that it receives from multiple suppliers and distribute them to the hospitals that it manages. With this computer program, user able to keep track on what item assigned to which supplier, remaining stock for every supplier with each item, and the distribution record to target hospital. User can register suppliers, hospitals, assign items to suppliers and save distribution record to txt file.

Assumption

- Assume the system can register suppliers.
- Assume the system can register hospitals.
- Assume the system can assign items to suppliers.
- Assume the system can register new item and assign to supplier based on supplier code.
- Assume the system can display item stocks based on supplier code.
- Assume the system can display item stock which less than 25 based on supplier code.
- Assiuem the system can update supplier details based on supplier code.
- Assume the system can update item stock based on supplier code.

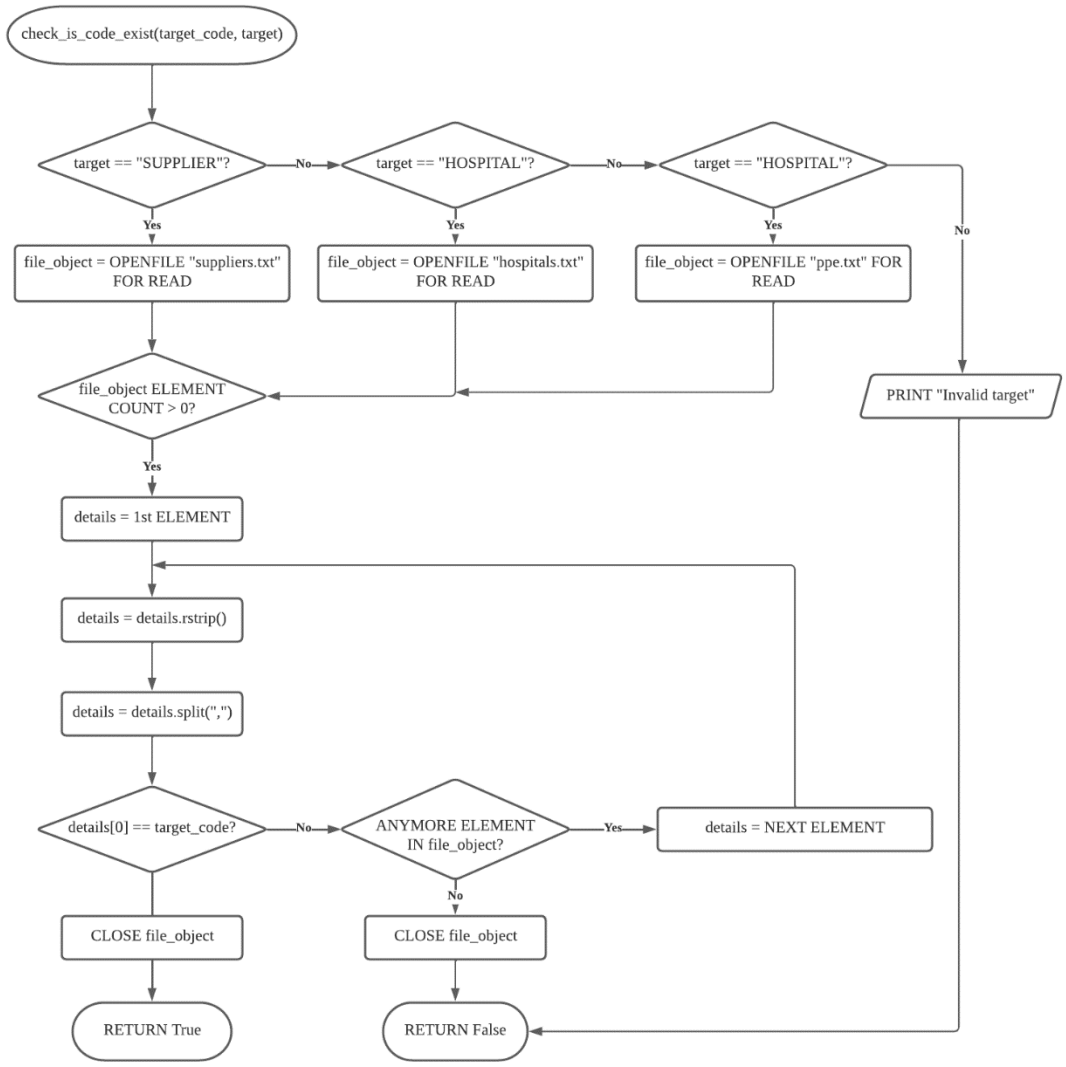
Design of the Program

check_is_code_exist

Pseudocode

```
FUNCTION check_is_code_exist(target_code, target)
  DECLARE String Filename
  DECLARE String Line
  DECLARE File file_object
  IF target == "SUPPLIER" THEN
    Filename = "supplier.txt"
  ELSE
    IF target == "HOSPITAL" THEN
      Filename = "hospital.txt"
    ELSE
      IF target == "ITEM" THEN
        Filename = "ppe.txt"
      ELSE
        PRINT "Invalid target"
        RETURN False
      ENDIF
    ENDIF
  ENDIF
  file_object = OPENFILE Filename FOR READ
  FOREACH details IN file_object
    details = details.split(",")
    IF details[0] = target_code THEN
      CLOSE file_object
      RETURN True
  ENDFOR
  CLOSE file_object
  RETURN False
END
```

Flow Chart

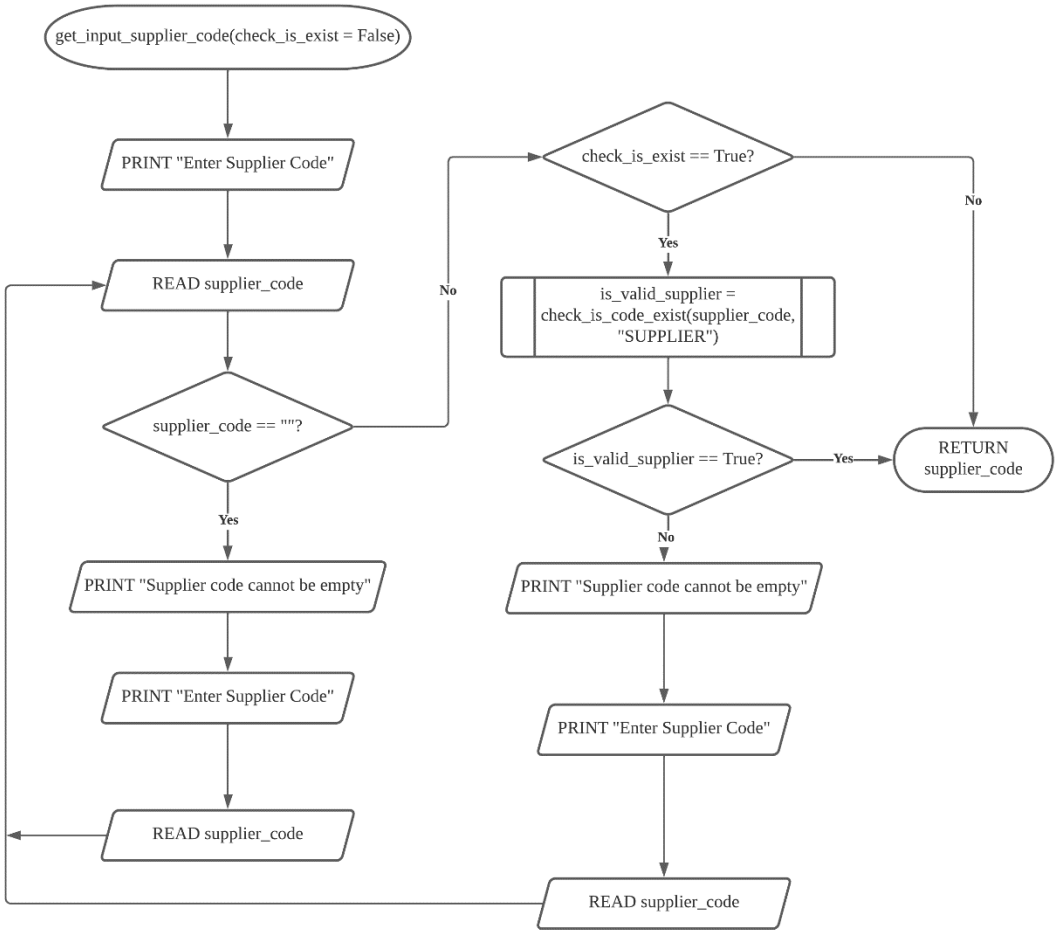

check_is_code_exist(target_code, target)
target == "SUPPLIER"?
No
target == "HOSPITAL"?
No
target == "HOSPITAL"?
No
Yes
Yes
Yes
file_object = OPENFILE "suppliers.txt" FOR READ
file_object = OPENFILE "hospitals.txt" FOR READ
file_object = OPENFILE "ppe.txt" FOR READ
file_object ELEMENT COUNT > 0?
PRINT "Invalid target"
Yes
details = 1st ELEMENT
details = details.rstrip()
details = details.split(",")
details[0] == target_code?
No
ANYMORE ELEMENT IN file_object?
Yes
details = NEXT ELEMENT
No
CLOSE file_object
CLOSE file_object
RETURN True
RETURN False

get_input_supplier_code

Pseudocode

```
FUNCTION get_input_supplier_code(check_is_exist=False)
  PRINT "Enter supplier code"
  READ supplier_code
  DOWHILE True
    IF supplier_code == "" THEN
      PRINT "Supplier Code cannot be empty"
      PRINT "Enter supplier code"
      READ supplier_code
    ELSE
      IF check_is_exist == True THEN
        is_valid_supplier = CALL check_is_code_exist(supplier_code, "SUPPLIER")
        IF is_valid_supplier THEN
          RETURN supplier_code
        ELSE
          PRINT "Supplier not found"
          PRINT "Enter supplier code"
          READ supplier_code
        ENDIF
      ELSE
        RETURN supplier_code
      ENDIF
    ENDIF
  ENDWHILE
ENDFUNCTION
```

Flow Chart

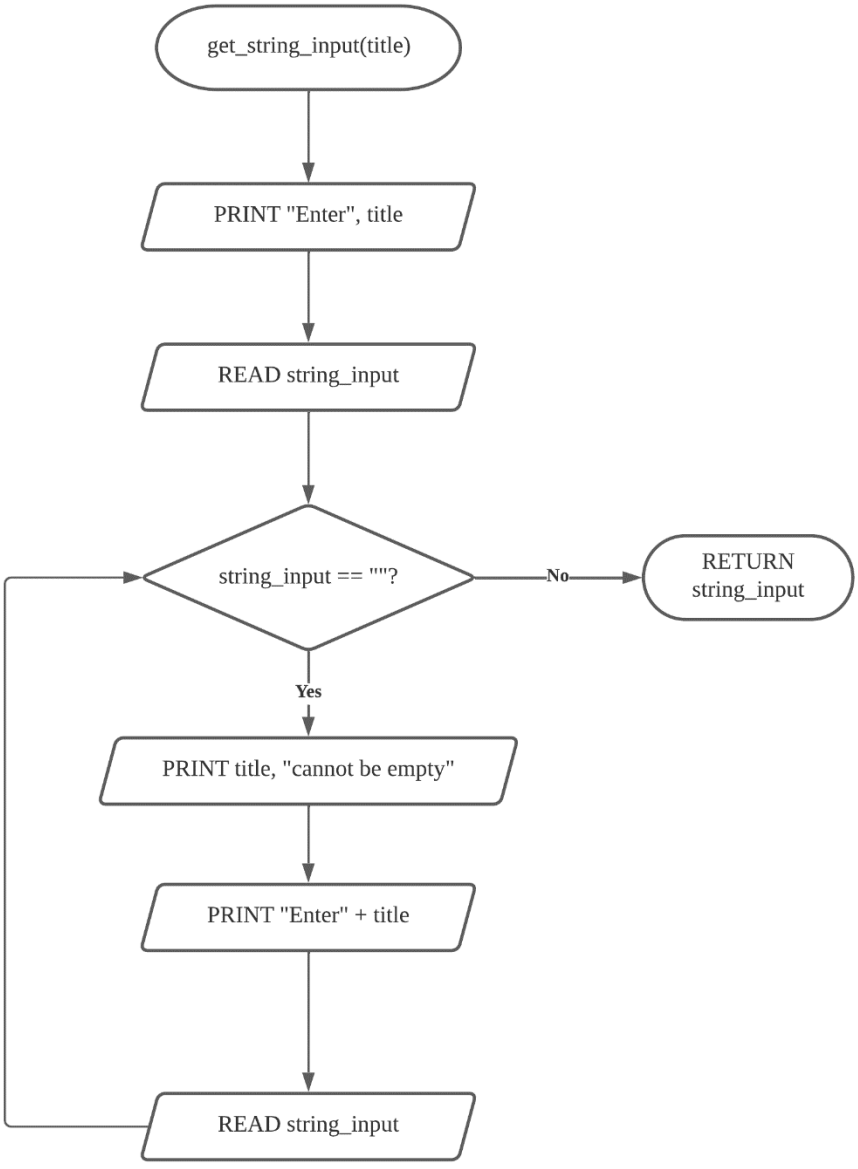

get_input_supplier_code(check_is_exist = False)
PRINT "Enter Supplier Code"
READ supplier_code
supplier_code == ""?
No
Yes
PRINT "Supplier code cannot be empty"
PRINT "Enter Supplier Code"
READ supplier_code
check_is_exist == True?
No
Yes
is_valid_supplier = check_is_code_exist(supplier_code, "SUPPLIER")
is_valid_supplier == True?
Yes
No
RETURN supplier_code
PRINT "Supplier code cannot be empty"
PRINT "Enter Supplier Code"
READ supplier_code

get_string_input

Pseudocode

```
FUNCTION get_string_input(title)
  PRINT "Enter" + title
  READ string_input
  DOWHILE True
    IF string_input = "" THEN
      PRINT title + "cannot be empty"
      PRINT "Enter" + title
      READ string_input
    ELSE
      RETURN string_input
    ENDIF
  ENDWHILE
ENDFUNCTION
```

Flow Chart

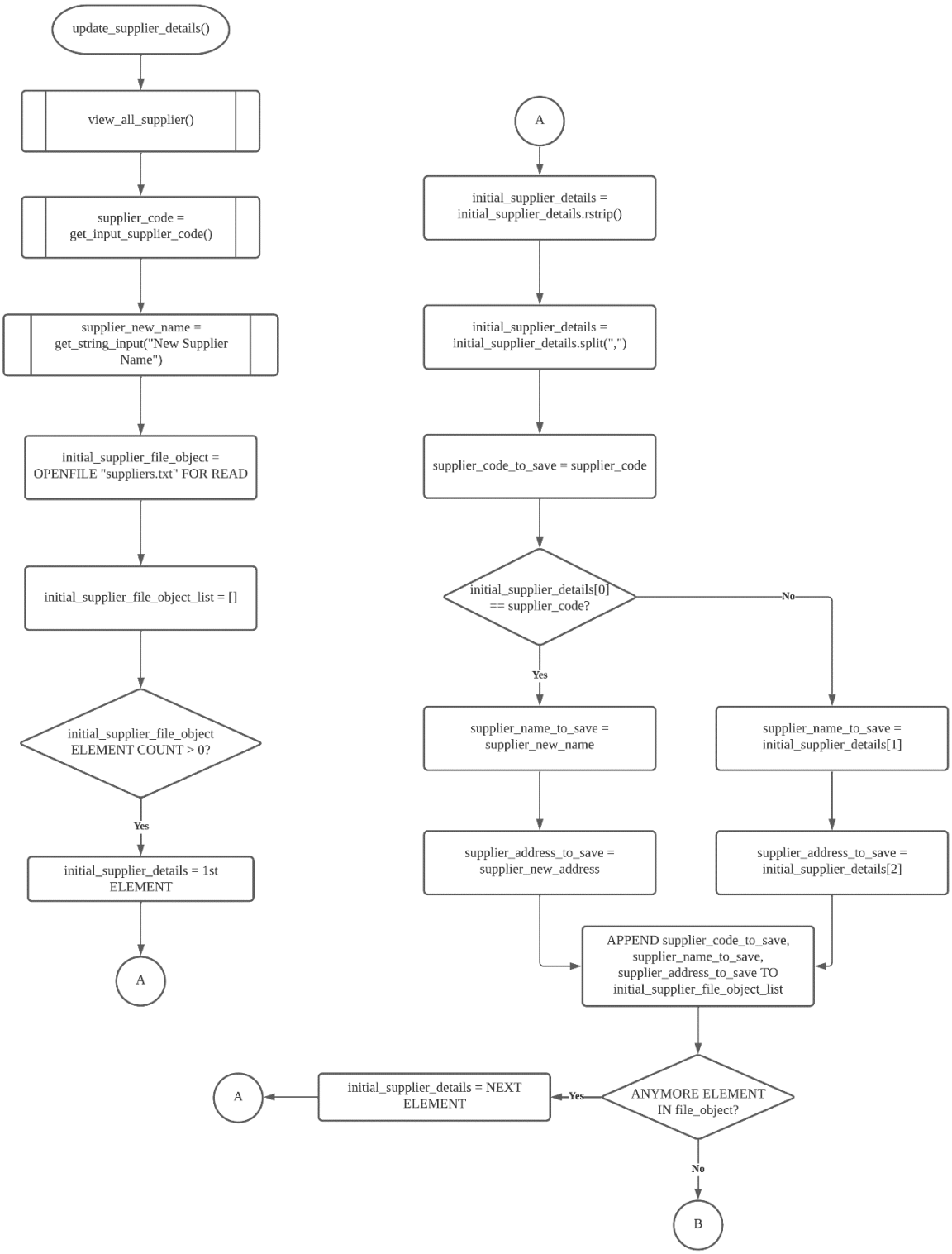


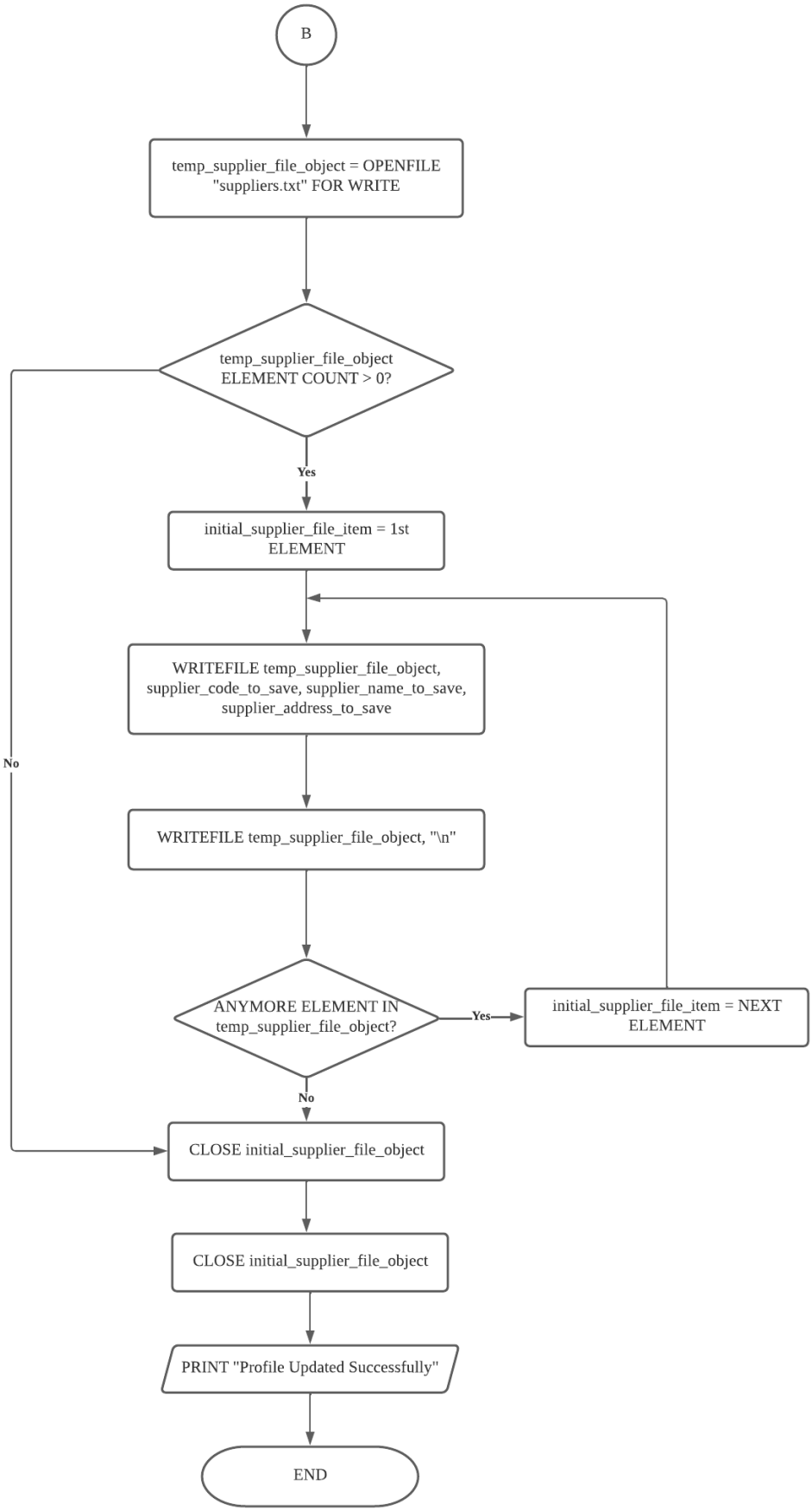
update_supplier_details

Pseudocode

```
FUNCTION update_supplier_details()
  DECLARE File initial_supplier_file_object
  DECLARE String supplier_code_to_save
  CALL view_all_supplier()
  supplier_code = CALL get_input_supplier_code()
  supplier_new_name = CALL get_string_input("New Supplier Name")
  supplier_new_address = CALL get_string_input("New Supplier Address")
  initial_supplier_file_object = OPENFILE "suppliers.txt" FOR READ
  DECLARE List initial_supplier_file_object_list
  FOREACH initial_supplier_details IN initial_supplier_file_object
    initial_supplier_details = initial_supplier_details.rstrip()
    initial_supplier_details = initial_supplier_details.split(",")
    supplier_code_to_save = supplier_code
    IF initial_supplier_details[0] = supplier_code THEN
      supplier_name_to_save = supplier_new_name
      supplier_address_to_save = supplier_new_address
    ELSE
      supplier_name_to_save = initial_supplier_details[1]
      supplier_address_to_save = initial_supplier_details[2]
    ENDIF
    APPEND supplier_code_to_save, supplier_name_to_save, supplier_address_to_save
TO initial_supplier_file_object_list
  ENDFOR
  temp_supplier_file_object = OPENFILE "suppliers.txt" FOR WRITE
  FOREACH initial_supplier_file_item IN initial_supplier_file_object_list
    WRITEFILE temp_supplier_file_object,
    supplier_code_to_save, supplier_name_to_save, supplier_address_to_save
    WRITEFILE temp_supplier_file_object, "\n"
  ENDFOR
  CLOSE initial_supplier_file_object
  CLOSE temp_supplier_file_object
  PRINT "Profile Updated Successfully"
ENDFUNCTION
```

Flow Chart


update_supplier_details()
view_all_supplier()
supplier_code = get_input_supplier_code()
supplier_new_name = get_string_input("New Supplier Name")
initial_supplier_file_object = OPENFILE "suppliers.txt" FOR READ
initial_supplier_file_object_list = []
initial_supplier_file_object ELEMENT COUNT > 0?
Yes
initial_supplier_details = 1st ELEMENT
A
A
initial_supplier_details = initial_supplier_details.rstrip()
initial_supplier_details = initial_supplier_details.split(",")
supplier_code_to_save = supplier_code
initial_supplier_details[0] == supplier_code?
No
Yes
supplier_name_to_save = supplier_new_name
supplier_name_to_save = initial_supplier_details[1]
supplier_address_to_save = supplier_new_address
supplier_address_to_save = initial_supplier_details[2]
APPEND supplier_code_to_save, supplier_name_to_save, supplier_address_to_save TO initial_supplier_file_object_list
ANYMORE ELEMENT IN file_object?
Yes
initial_supplier_details = NEXT ELEMENT
A
No
B

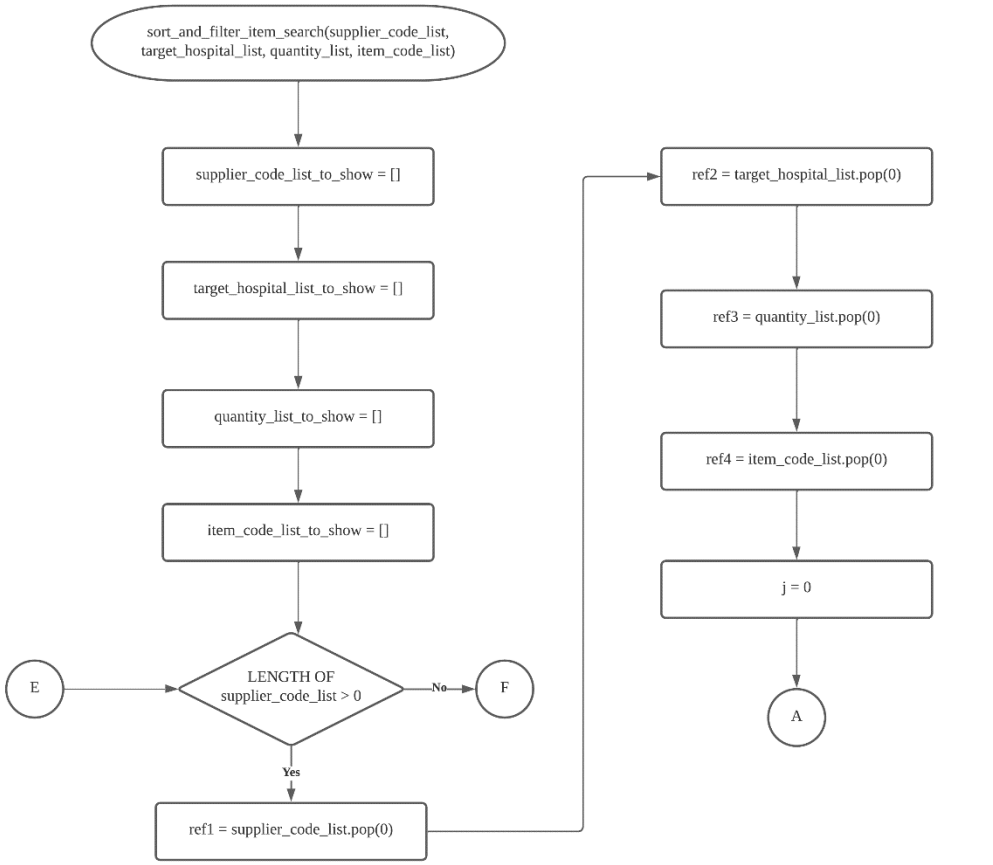

B
temp_supplier_file_object = OPENFILE "suppliers.txt" FOR WRITE
temp_supplier_file_object ELEMENT COUNT > 0?
Yes
No
initial_supplier_file_item = 1st ELEMENT
WRITEFILE temp_supplier_file_object, supplier_code_to_save, supplier_name_to_save, supplier_address_to_save
WRITEFILE temp_supplier_file_object, "\n"
ANYMORE ELEMENT IN temp_supplier_file_object?
Yes
initial_supplier_file_item = NEXT ELEMENT
No
CLOSE initial_supplier_file_object
CLOSE initial_supplier_file_object
PRINT "Profile Updated Successfully"
END

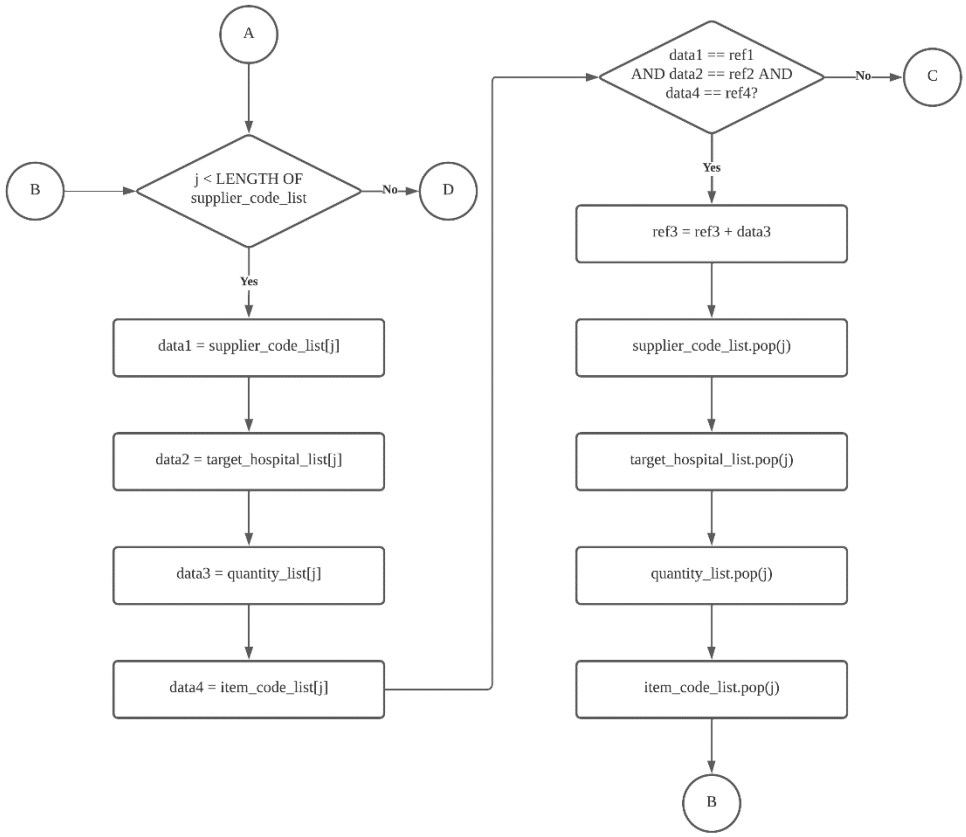
sort_and_filter_item_search

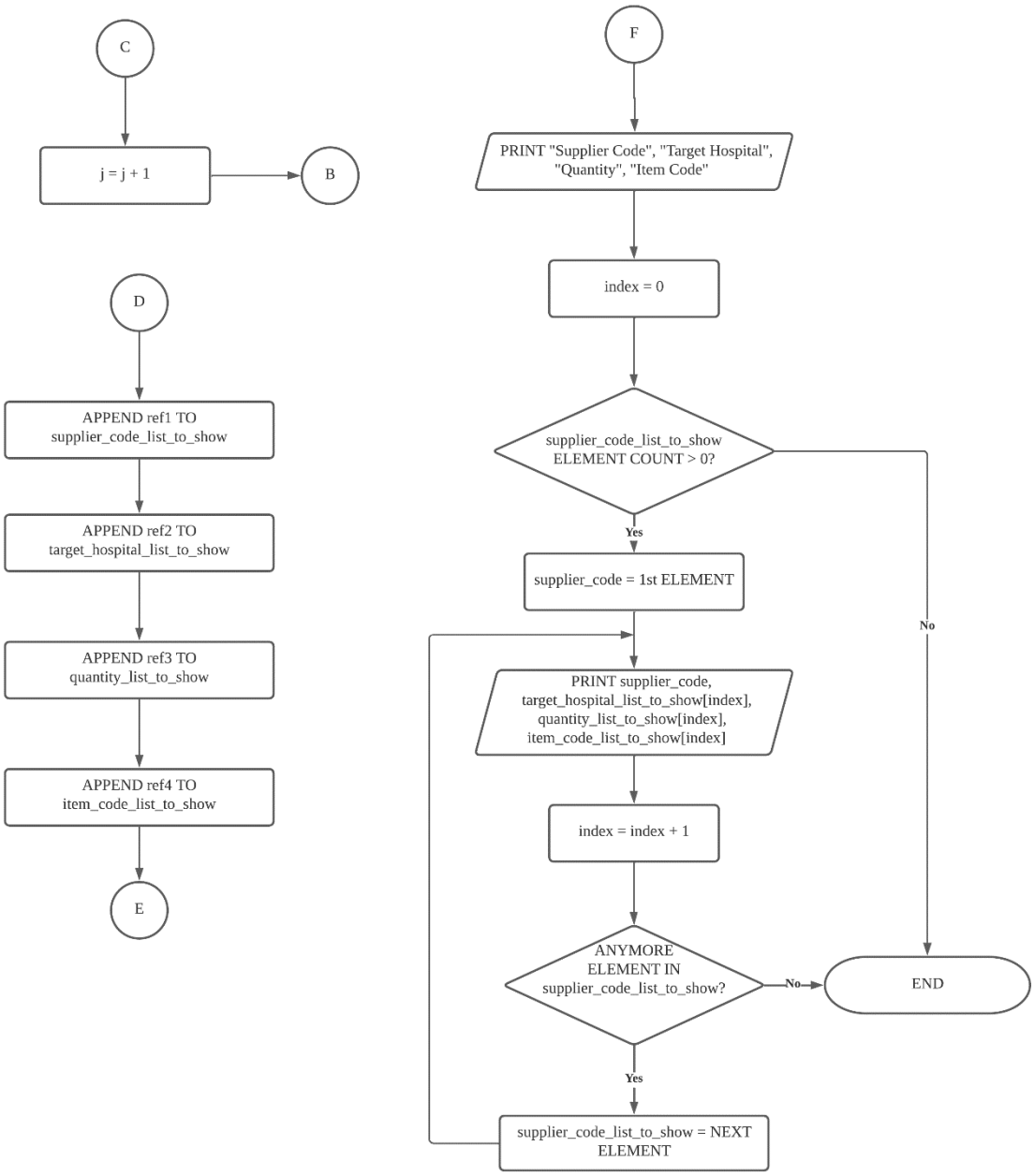
Pseudocode

```
FUNCTION sort_and_filter_item_search(supplier_code_list, target_hospital_list,
quantity_list, item_code_list)
  DECLARE List supplier_code_list_to_show
  DECLARE List target_hospital_list_to_show
  DECLARE List quantity_list_to_show
  DECLARE List item_code_list_to_show
  DOWHILE LENGTH OF supplier_code_list > 0
    ref1 = supplier_code_list.pop(0)
    ref2 = target_hospital_list.pop(0)
    ref3 = quantity_list.pop(0)
    ref4 = item_code_list.pop(0)
    j = 0
    DOWHILE LENGTH OF supplier_code_list > 0
      data1 = supplier_code_list[j]
      data2 = target_hospital_list[j]
      data3 = quantity_list[j]
      data4 = item_code_list[j]
      IF data1 == ref1 AND data2 == ref2 AND data4 == ref4 THEN
        ref3 = ref3 + data3
        supplier_code_list.pop(j)
        target_hospital_list.pop(j)
        quantity_list.pop(j)
        item_code_list.pop(j)
        CONTINUE
      ENDIF
      j += 1
    ENDDO
    APPEND ref1 TO supplier_code_list_to_show
    APPEND ref2 TO target_hospital_list_to_show
    APPEND ref3 TO quantity_list_to_show
    APPEND ref4 TO item_code_list_to_show
  ENDDO
  index = 0
  FOREACH supplier_code IN supplier_code_list_to_show
    PRINT supplier_code,
    target_hospital_list_to_show[index], quantity_list_to_show[index],
    item_code_list_to_show[index]
    index = index + 1
  ENDFOR
ENDFUNCTION
```

Flow Chart


sort_and_filter_item_search(supplier_code_list, target_hospital_list, quantity_list, item_code_list)
supplier_code_list_to_show = []
target_hospital_list_to_show = []
quantity_list_to_show = []
item_code_list_to_show = []
E
LENGTH OF supplier_code_list > 0
No
F
Yes
ref1 = supplier_code_list.pop(0)
ref2 = target_hospital_list.pop(0)
ref3 = quantity_list.pop(0)
ref4 = item_code_list.pop(0)
j = 0
A


A
B
j < LENGTH OF supplier_code_list
No
D
Yes
data1 = supplier_code_list[j]
data2 = target_hospital_list[j]
data3 = quantity_list[j]
data4 = item_code_list[j]
data1 == ref1 AND data2 == ref2 AND data4 == ref4?
No
C
Yes
ref3 = ref3 + data3
supplier_code_list.pop(j)
target_hospital_list.pop(j)
quantity_list.pop(j)
item_code_list.pop(j)
B

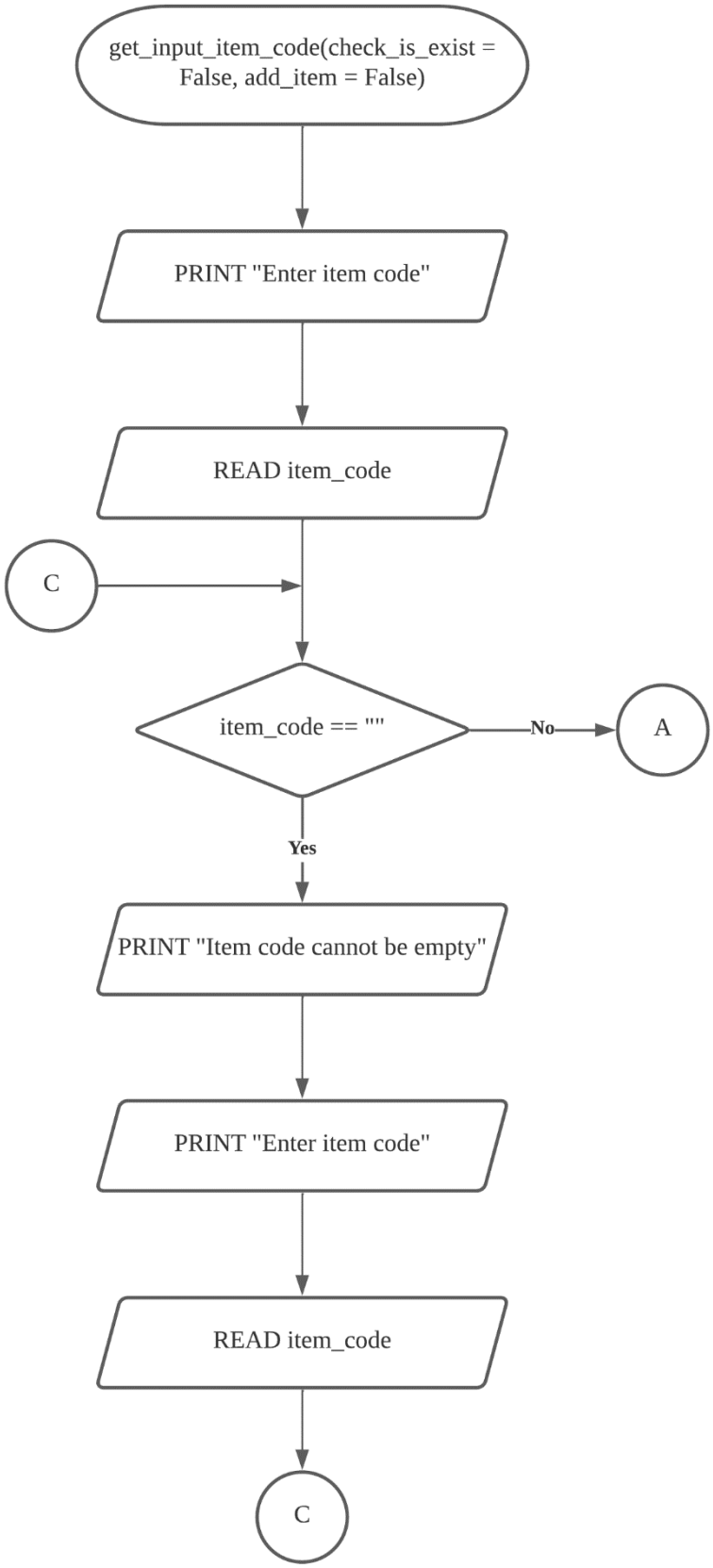

C
j = j + 1
B
D
APPEND ref1 TO supplier_code_list_to_show
APPEND ref2 TO target_hospital_list_to_show
APPEND ref3 TO quantity_list_to_show
APPEND ref4 TO item_code_list_to_show
E
F
PRINT "Supplier Code", "Target Hospital", "Quantity", "Item Code"
index = 0
supplier_code_list_to_show ELEMENT COUNT > 0?
Yes
No
supplier_code = 1st ELEMENT
PRINT supplier_code, target_hospital_list_to_show[index], quantity_list_to_show[index], item_code_list_to_show[index]
index = index + 1
ANYMORE ELEMENT IN supplier_code_list_to_show?
No
END
Yes
supplier_code_list_to_show = NEXT ELEMENT

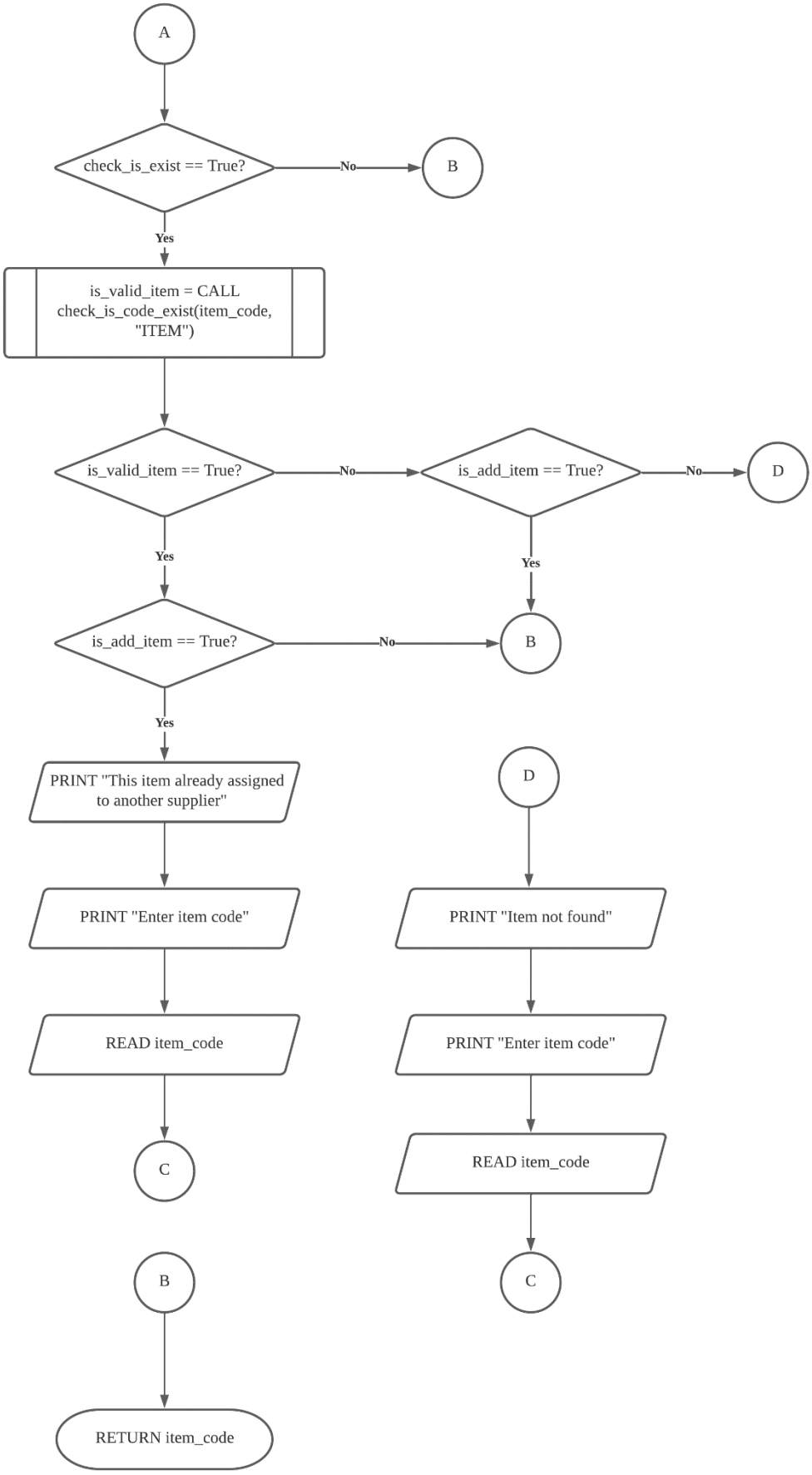
get_input_item_code

Pseudocode

```
FUNCTION get_input_item_code(check_is_exist = False, add_item = False)
  PRINT "Enter item code"
  READ item_code
  DOWHILE True
    IF item_code == "" THEN
      PRINT "Item code cannot be empty"
      PRINT "Enter item code"
      READ item_code
    ELSE
      IF check_is_exist == True THEN
        is_valid_item = CALL check_is_code_exist(supplier_code, "ITEM")
        IF is_valid_item THEN
          IF add_item THEN
            PRINT "This item already assigned to another supplier"
            PRINT "Enter item code"
            READ item_code
          ELSE
            READ item_code
          ENDIF
        ELSE
          IF add_item THEN
            RETURN item_code
          PRINT "Item not found"
          PRINT "Enter item code"
          READ item_code
        ENDIF
      ELSE
        RETURN item_code
      ENDIF
    ENDIF
  ENDWHILE
ENDFUNCTION
```

Flow Chart


get_input_item_code(check_is_exist = False, add_item = False)
PRINT "Enter item code"
READ item_code
C
item_code == ""
No
A
Yes
PRINT "Item code cannot be empty"
PRINT "Enter item code"
READ item_code
C

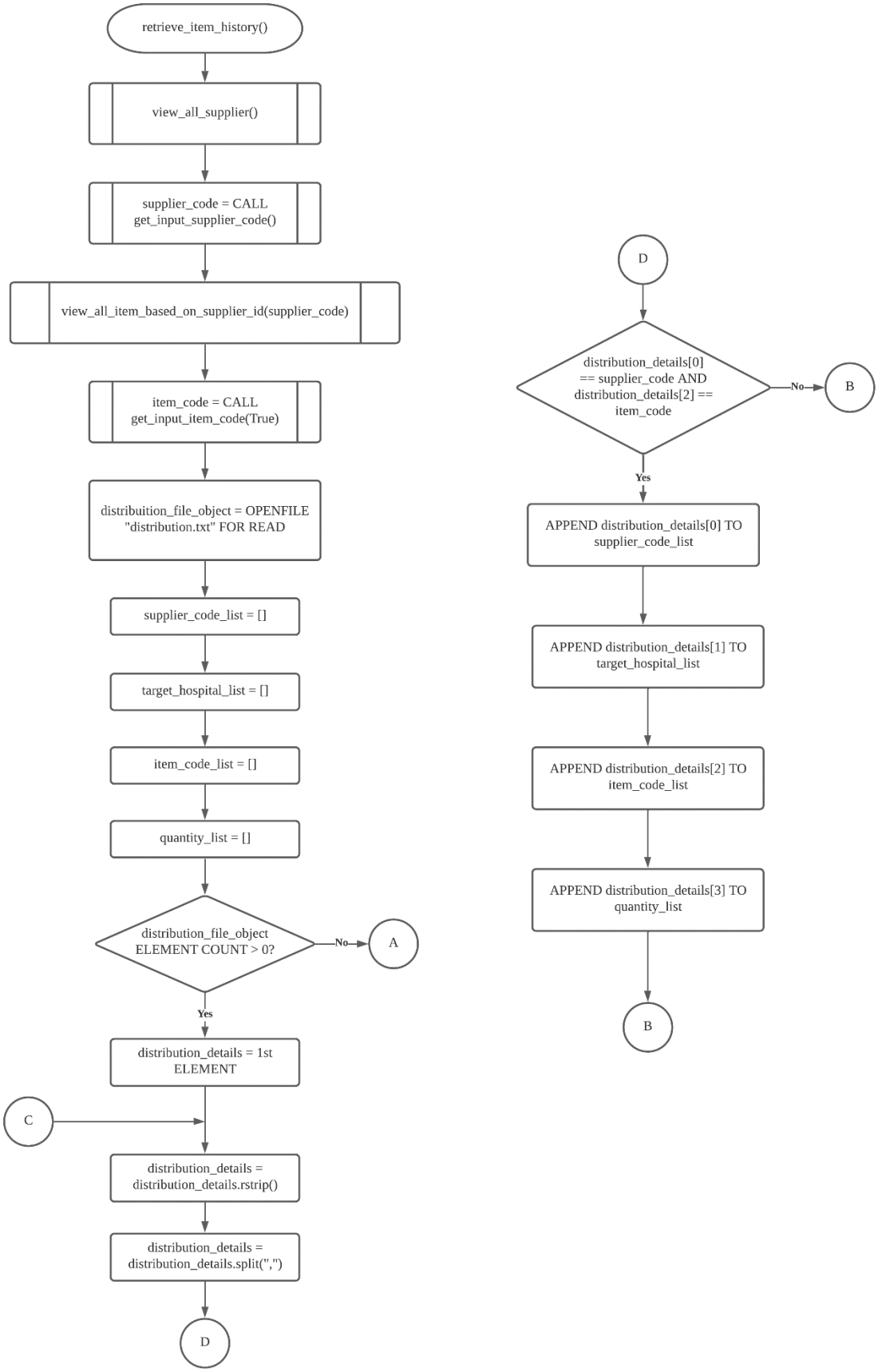

A
check_is_exist == True?
No
B
Yes
is_valid_item = CALL check_is_code_exist(item_code, "ITEM")
is_valid_item == True?
No
is_add_item == True?
No
D
Yes
Yes
is_add_item == True?
No
B
Yes
PRINT "This item already assigned to another supplier"
PRINT "Enter item code"
READ item_code
C
B
RETURN item_code
D
PRINT "Item not found"
PRINT "Enter item code"
READ item_code
C

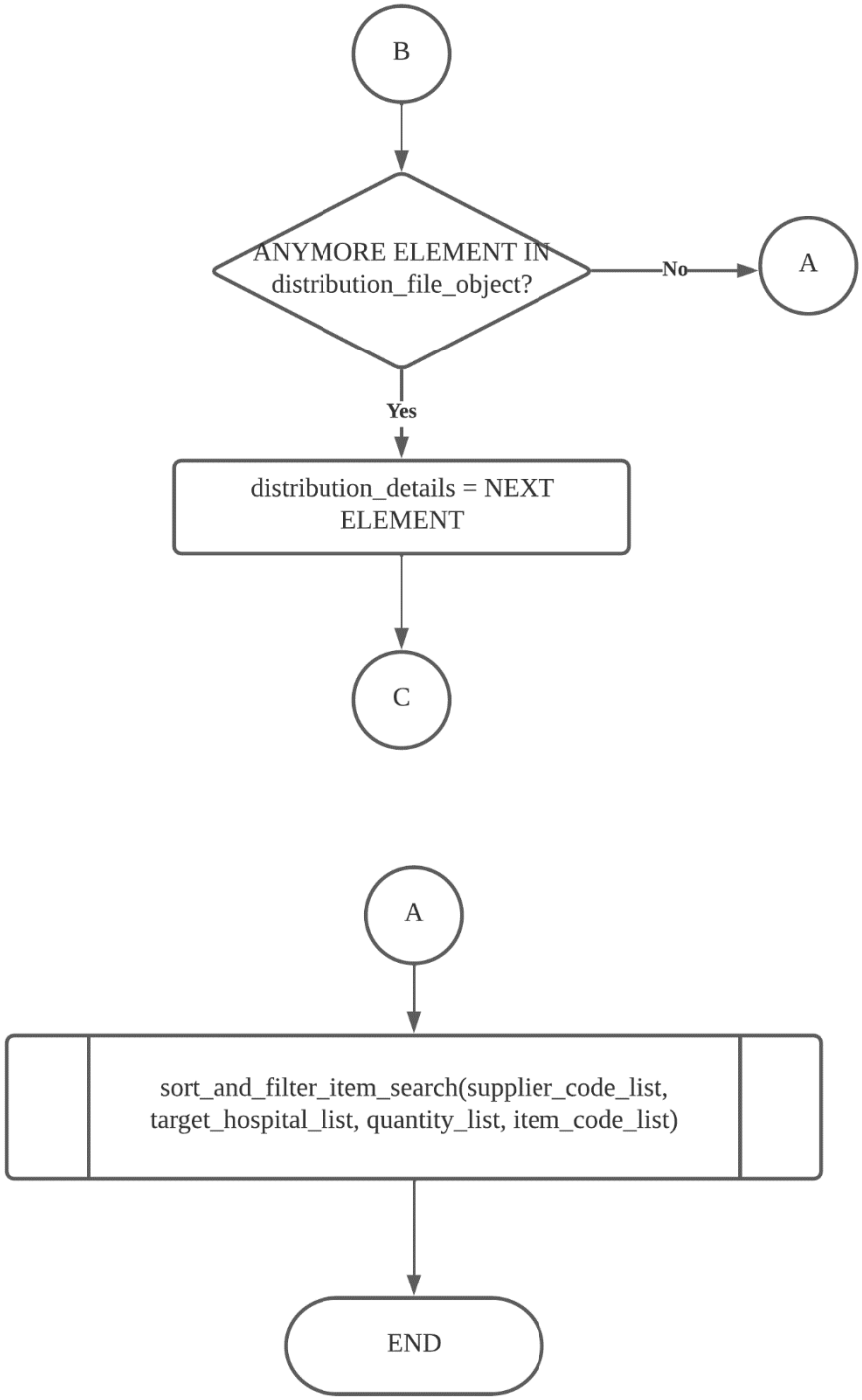
retrieve_item_history

Pseudocode

```
FUNCTION retrieve_item_history()
  CALL view_all_supplier()
  supplier_code = CALL get_input_supplier_code()
  CALL view_all_item_based_on_supplier_id(supplier_code)
  item_code = CALL get_input_item_code(True)
  distribution_file_object = OPENFILE "distribution.txt" FOR READ
  DECLARE List supplier_code_list
  DECLARE List target_hospital_list
  DECLARE List item_code_list
  DECLARE List quantity_list
  FOREACH distribution_details IN distribution_file_object
    distribution_details = distribution_details.rstrip()
    distribution_details = distribution_details.split(",")
    IF distribution_details[0] == supplier_code AND distribution_details[2] ==
item_code
      APPEND distribution_details[0] TO supplier_code_list
      APPEND distribution_details[1] TO target_hospital_list
      APPEND distribution_details[2] TO item_code_list
      APPEND distribution_details[3] TO quantity_list
    ENDIF
  ENDFOR
  CALL sort_and_filter_item_search(supplier_code_list, target_hospital_list,
quantity_list, item_code_list)
ENDFUNCTION
```

Flow Chart


retrieve_item_history()
view_all_supplier()
supplier_code = CALL get_input_supplier_code()
view_all_item_based_on_supplier_id(supplier_code)
item_code = CALL get_input_item_code(True)
distribuition_file_object = OPENFILE "distribution.txt" FOR READ
supplier_code_list = []
target_hospital_list = []
item_code_list = []
quantity_list = []
distribution_file_object ELEMENT COUNT > 0?
No
A
Yes
distribution_details = 1st ELEMENT
C
distribution_details = distribution_details.rstrip()
distribution_details = distribution_details.split(",")
D
D
distribution_details[0] == supplier_code AND distribution_details[2] == item_code
No
B
Yes
APPEND distribution_details[0] TO supplier_code_list
APPEND distribution_details[1] TO target_hospital_list
APPEND distribution_details[2] TO item_code_list
APPEND distribution_details[3] TO quantity_list
B

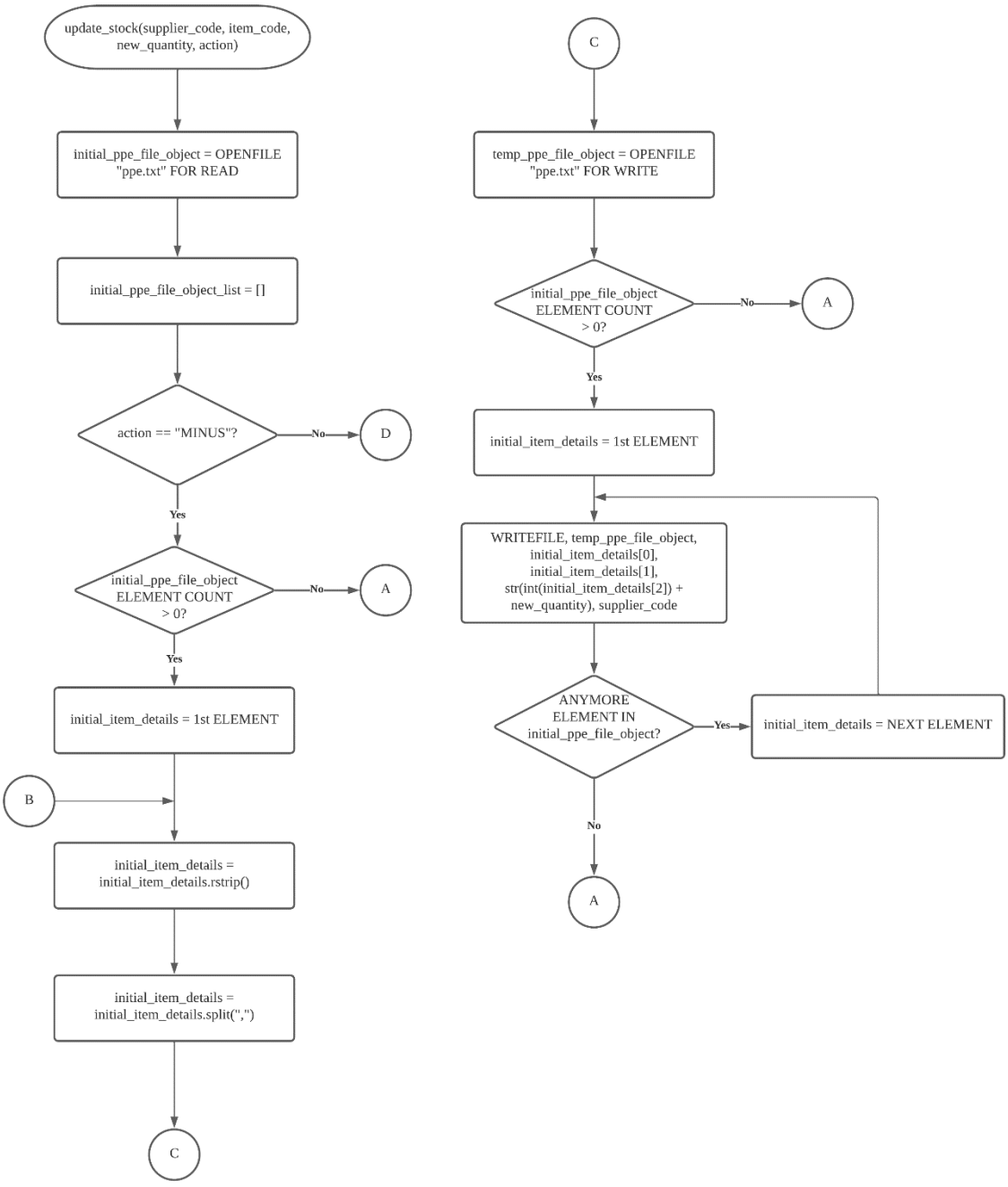

B
ANYMORE ELEMENT IN distribution_file_object?
No
A
Yes
distribution_details = NEXT ELEMENT
C
A
sort_and_filter_item_search(supplier_code_list, target_hospital_list, quantity_list, item_code_list)
END

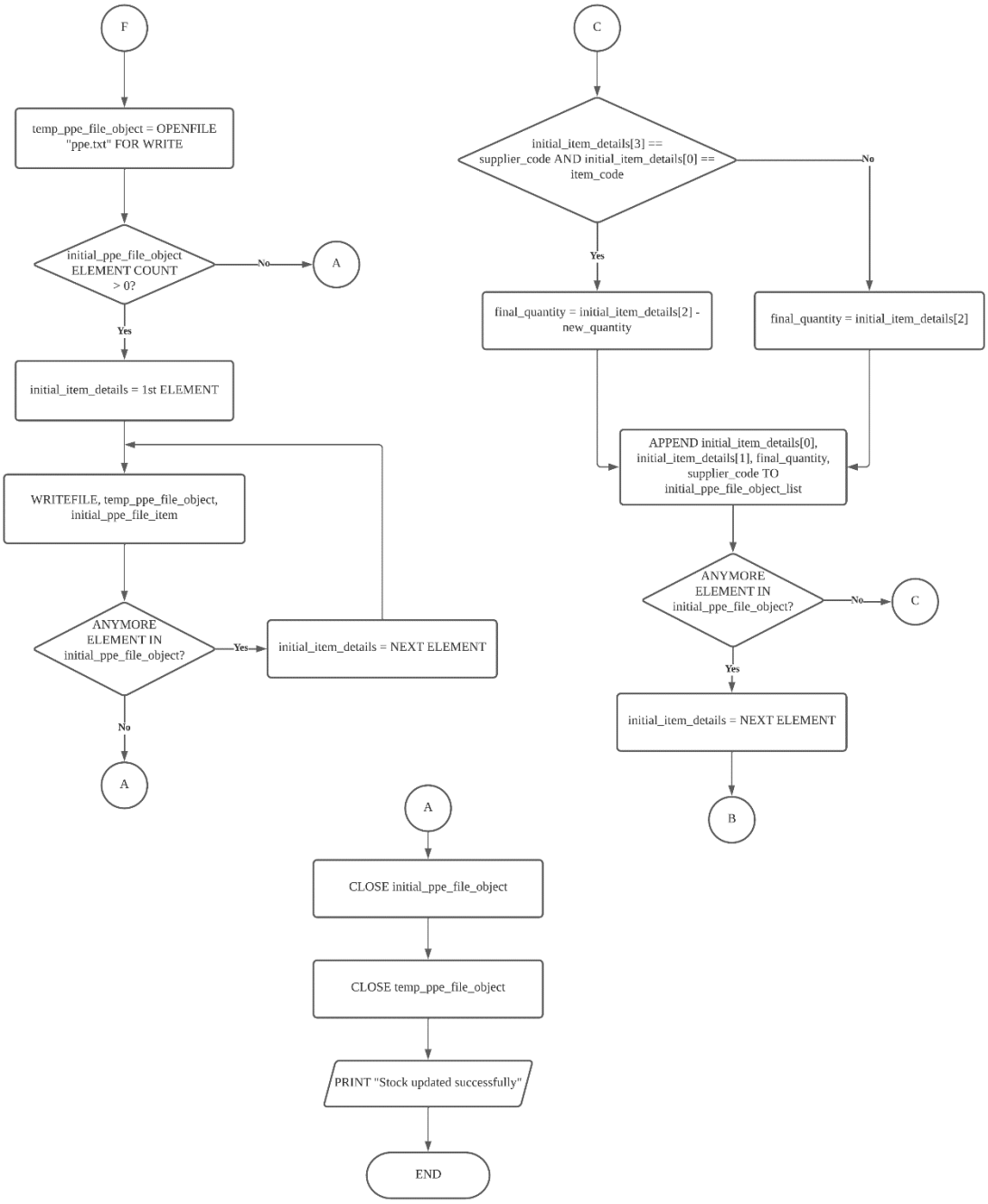
update_stock

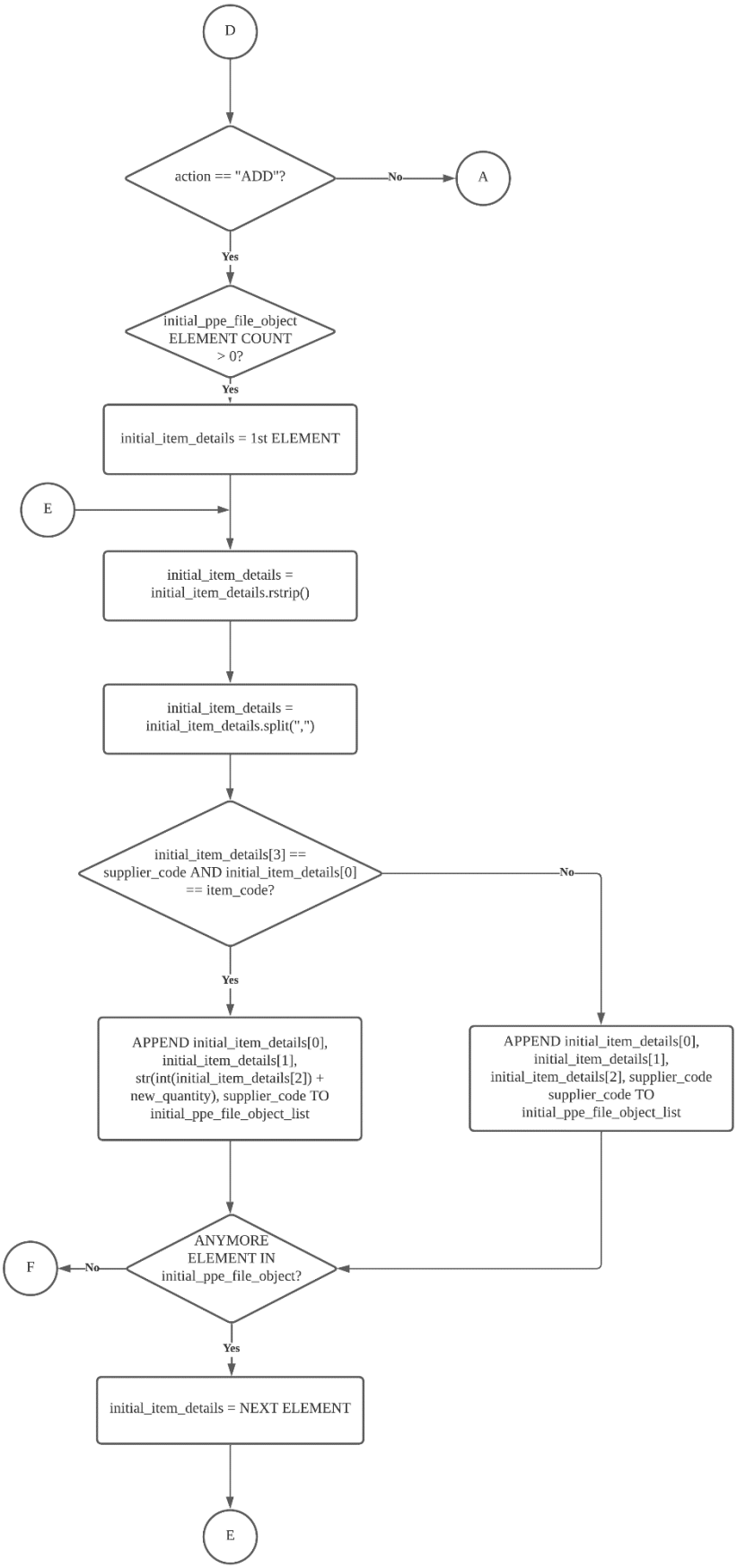
Pseudocode

```
FUNCTION update_stock(supplier_code, item_code, new_quantity, action)
  initial_ppe_file_object = OPENFILE "ppe.txt" FOR READ
  DECLARE List initial_ppe_file_object_list
  IF action == "MINUS"
    FOREACH initial_item_details IN initial_ppe_file_object
      initial_item_details = initial_item_details.rstrip()
      initial_item_details = initial_item_details.split(",")
      IF initial_item_details[3] == supplier_code AND
      initial_item_details[0] == item_code
        final_quantity = initial_item_details[2] - new_quantity
      ELSE
        final_quantity = initial_item_details[2]
      ENDIF
      WRITEFILE temp_supplier_file_object,
      initial_item_details[0],  initial_item_details[1],
      final_quantity, supplier_code
      WRITEFILE "\n"
    ENDFOR
    temp_ppe_file_object = OPENFILE "suppliers.txt" FOR WRITE
    FOREACH initial_item_details IN initial_ppe_file_object
      WRITEFILE temp_supplier_file_object, initial_item_details
      WRITEFILE "\n"
    ENDFOR
  ELSE
    IF action == "ADD"
      FOREACH initial_item_details IN initial_ppe_file_object
        initial_item_details = initial_item_details.rstrip()
        initial_item_details = initial_item_details.split(",")
        IF initial_item_details[3] == supplier_code AND
        initial_item_details[0] == item_code
          APPEND initial_item_details[0],
         initial_item_details[1], str(int(initial_item_details[2])
         + new_quantity),
         supplier_code TO initial_ppe_file_object_list
        ELSE
          APPEND initial_item_details[0],
          initial_item_details[1], initial_item_details[2],
          supplier_code TO initial_ppe_file_object_list
        ENDIF
      ENDFOR
      temp_ppe_file_object = OPENFILE "ppe.txt" FOR WRITE
      FOREACH initial_ppe_file_item IN temp_ppe_file_object
        WRITEFILE temp_supplier_file_object, initial_item_details
        WRITEFILE "\n"
      ENDFOR
    ENDIF
  ENDIF
  CLOSE initial_ppe_file_object
  CLOSE temp_ppe_file_object
  PRINT "Stock updated successfully"
ENDFUNCTION
```

Flow Chart


update_stock(supplier_code, item_code, new_quantity, action)
initial_ppe_file_object = OPENFILE "ppe.txt" FOR READ
initial_ppe_file_object_list = []
action == "MINUS"?
No
D
Yes
initial_ppe_file_object ELEMENT COUNT > 0?
No
A
Yes
initial_item_details = 1st ELEMENT
B
initial_item_details = initial_item_details.rstrip()
initial_item_details = initial_item_details.split(",")
C
C
temp_ppe_file_object = OPENFILE "ppe.txt" FOR WRITE
initial_ppe_file_object ELEMENT COUNT > 0?
No
A
Yes
initial_item_details = 1st ELEMENT
WRITEFILE, temp_ppe_file_object, initial_item_details[0], initial_item_details[1], str(int(initial_item_details[2]) + new_quantity), supplier_code
ANYMORE ELEMENT IN initial_ppe_file_object?
Yes
initial_item_details = NEXT ELEMENT
No
A


F
temp_ppe_file_object = OPENFILE "ppe.txt" FOR WRITE
initial_ppe_file_object ELEMENT COUNT > 0?
No
A
Yes
initial_item_details = 1st ELEMENT
WRITEFILE, temp_ppe_file_object, initial_ppe_file_item
ANYMORE ELEMENT IN initial_ppe_file_object?
Yes
initial_item_details = NEXT ELEMENT
No
A
C
initial_item_details[3] == supplier_code AND initial_item_details[0] == item_code
No
Yes
final_quantity = initial_item_details[2] - new_quantity
final_quantity = initial_item_details[2]
APPEND initial_item_details[0], initial_item_details[1], final_quantity, supplier_code TO initial_ppe_file_object_list
ANYMORE ELEMENT IN initial_ppe_file_object?
No
C
Yes
initial_item_details = NEXT ELEMENT
B
A
CLOSE initial_ppe_file_object
CLOSE temp_ppe_file_object
PRINT "Stock updated successfully"
END

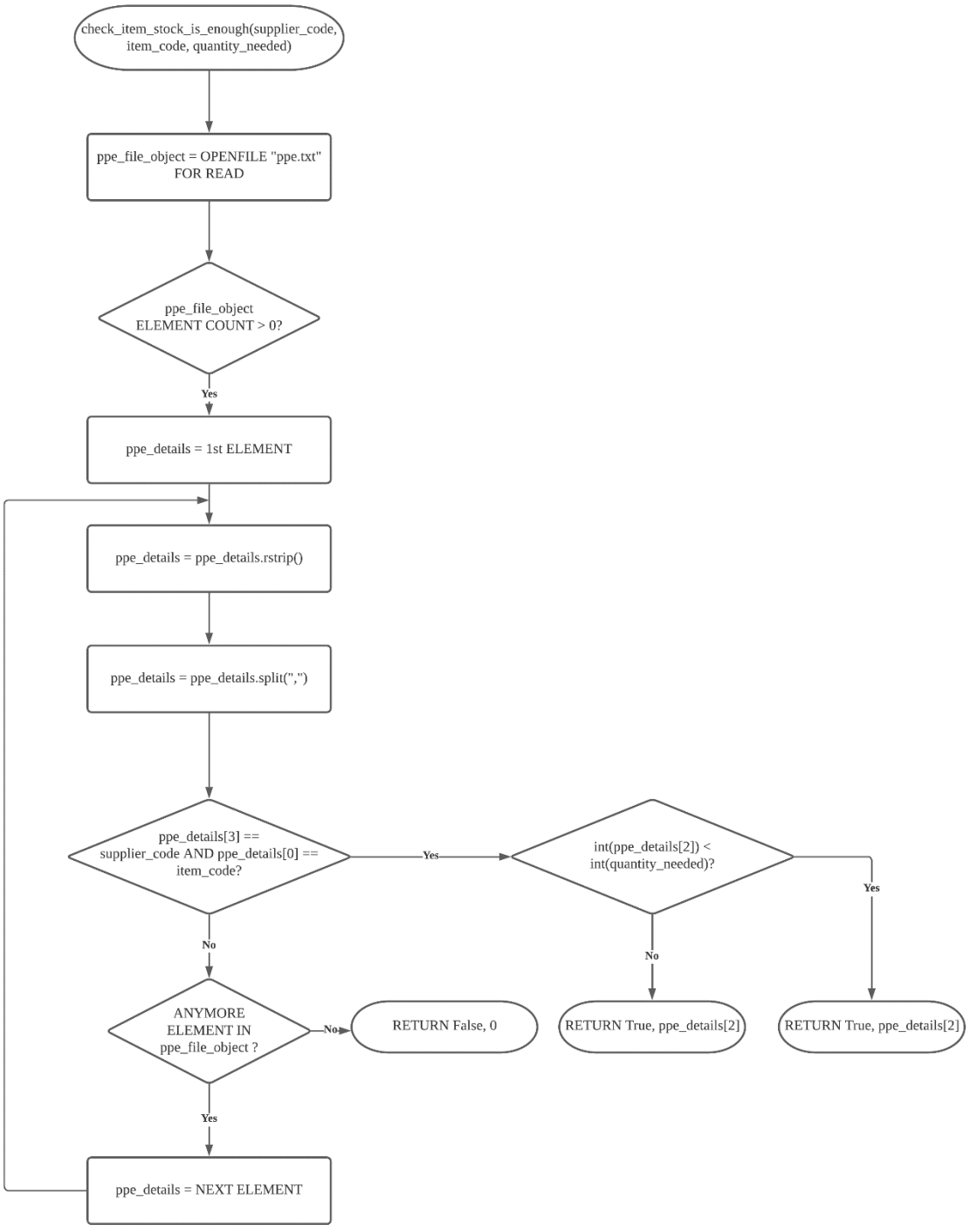

D
action == "ADD"?
No
A
Yes
initial_ppe_file_object ELEMENT COUNT > 0?
Yes
initial_item_details = 1st ELEMENT
E
initial_item_details = initial_item_details.rstrip()
initial_item_details = initial_item_details.split(",")
initial_item_details[3] == supplier_code AND initial_item_details[0] == item_code?
No
Yes
APPEND initial_item_details[0], initial_item_details[1], str(int(initial_item_details[2]) + new_quantity), supplier_code TO initial_ppe_file_object_list
APPEND initial_item_details[0], initial_item_details[1], initial_item_details[2], supplier_code supplier_code TO initial_ppe_file_object_list
ANYMORE ELEMENT IN initial_ppe_file_object?
F
No
Yes
initial_item_details = NEXT ELEMENT
E

check_item_stock_is_enough

Pseudocode

```
FUNCTION check_item_stock_is_enough(supplier_code, item_code, quantity_needed)
  ppe_file_object = OPENFILE "ppe.txt" FOR READ
  FOREACH ppe_details IN ppe_file_object
    ppe_details = ppe_details.rstrip()
    ppe_details = ppe_details.split(",")
    IF ppe_details[3] == supplier_code AND ppe_details[0] == item_code
      IF ippe_details[2] < quantity_needed
        RETURN False, ppe_details[2]
      ELSE
        RETURN True, ppe_details[2]
    ENDIF
  ENDFOR
  RETURN False, 0
ENDFUNCTION
```

Flow Chart

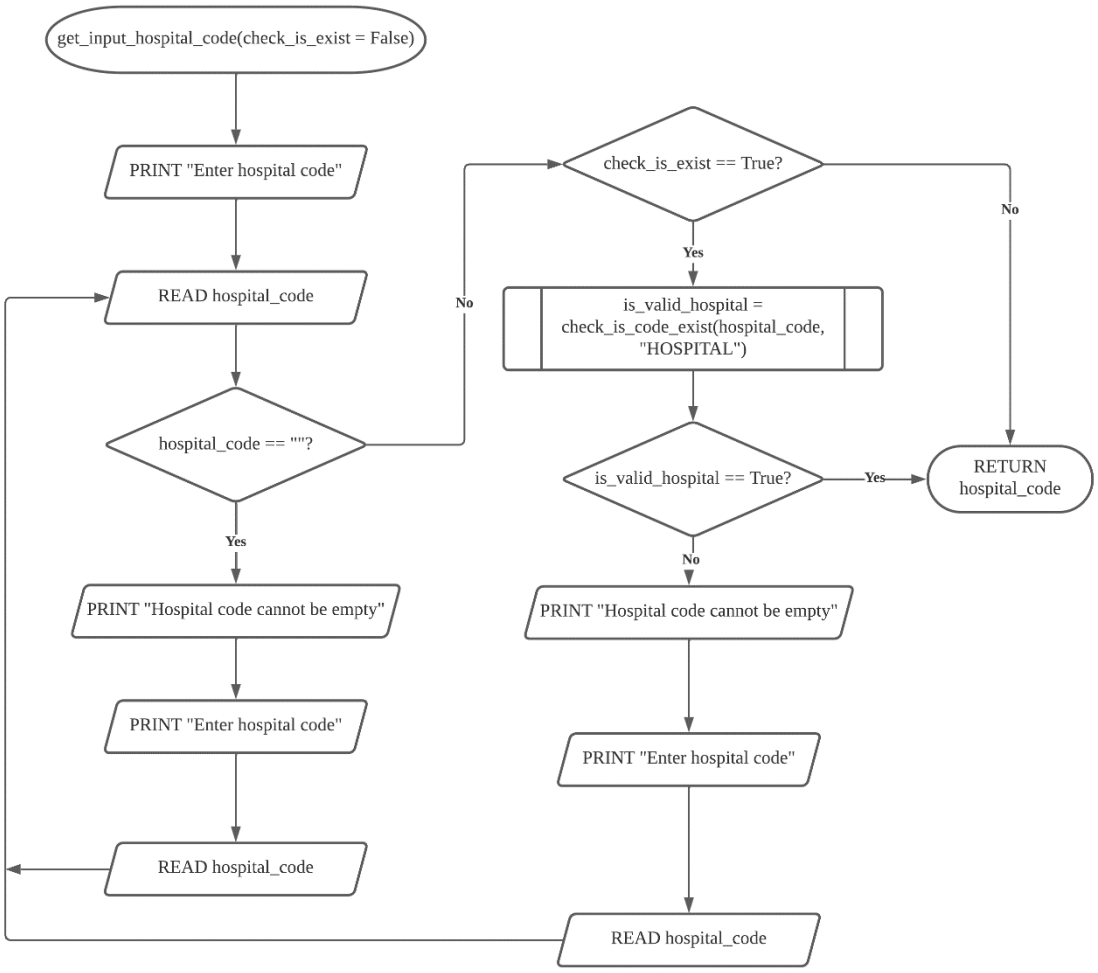

check_item_stock_is_enough(supplier_code, item_code, quantity_needed)
ppe_file_object = OPENFILE "ppe.txt" FOR READ
ppe_file_object ELEMENT COUNT > 0?
Yes
ppe_details = 1st ELEMENT
ppe_details = ppe_details.rstrip()
ppe_details = ppe_details.split(",")
ppe_details[3] == supplier_code AND ppe_details[0] == item_code?
Yes
int(ppe_details[2]) < int(quantity_needed)?
Yes
No
RETURN True, ppe_details[2]
RETURN True, ppe_details[2]
No
ANYMORE ELEMENT IN ppe_file_object ?
No
RETURN False, 0
Yes
ppe_details = NEXT ELEMENT

get_input_hospital_code

Pseudocode

```
FUNCTION get_input_hospital_code(check_is_exist=False)
  PRINT "Enter hospital code"
  READ hospital_code
  DOWHILE True
    IF hospital_code == "" THEN
      PRINT "Hospital code cannot be empty"
      PRINT "Enter hospital code"
      READ hospital_code
    ELSE
      IF check_is_exist == True THEN
        is_valid_hospital = CALL check_is_code_exist(hospital_code, "HOSPITAL")
        IF is_valid_hospital THEN
          RETURN hospital_code
        ELSE
          PRINT "Hospital not found"
          PRINT "Enter hospital code"
          READ hospital_code
        ENDIF
      ELSE
        RETURN hospital_code
      ENDIF
    ENDIF
  ENDWHILE
ENDFUNCTION
```

Flow Chart

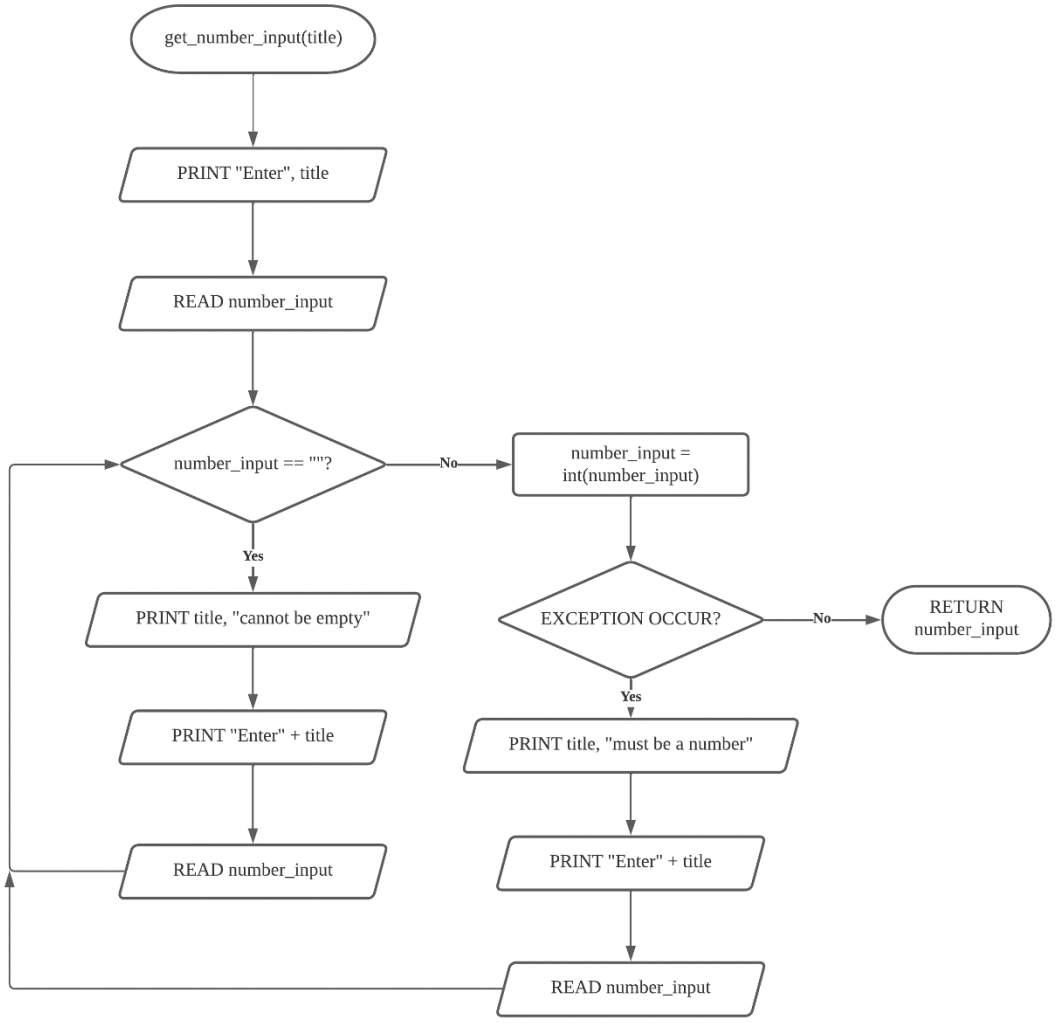

get_input_hospital_code(check_is_exist = False)
PRINT "Enter hospital code"
READ hospital_code
hospital_code == ""?
No
Yes
PRINT "Hospital code cannot be empty"
PRINT "Enter hospital code"
READ hospital_code
check_is_exist == True?
No
Yes
is_valid_hospital = check_is_code_exist(hospital_code, "HOSPITAL")
is_valid_hospital == True?
Yes
RETURN hospital_code
No
PRINT "Hospital code cannot be empty"
PRINT "Enter hospital code"
READ hospital_code

get_number_input

Pseudocode

```
FUNCTION get_number_input(title)
  PRINT "Enter", title
  READ number_input
  DOWHILE True
    IF number_input == "" THEN
      PRINT title, "cannot be empty"
      PRINT "Enter", title
    ELSE
      TRY
        number_input = int(number_input)
        RETURN number_input
      EXCEPT
        PRINT title, "must be a number"
        PRINT "Enter", title
    ENDIF
  ENDDO
ENDFUNCTION
```

Flow Chart

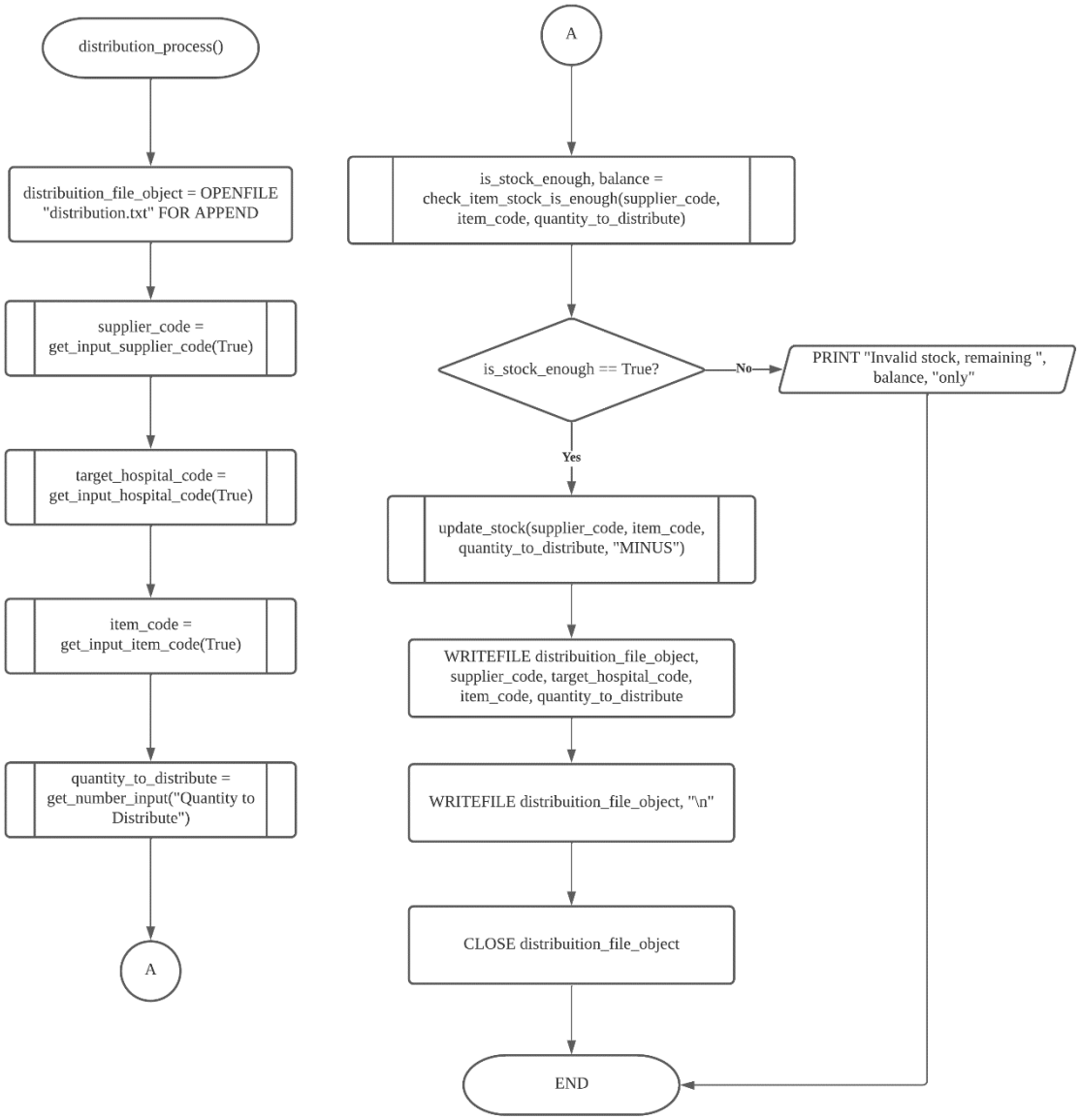


distribution_process

Pseudocode

```
FUNCTION distribution_process()
  distribution_file_object = OPENFILE "distribution.txt" FOR APPEND
  CALL view_all_supplier()
  supplier_code = CALL get_input_supplier_code(True)
  CALL view_all_item_based_on_supplier_id(supplier_code)
  target_hospital_code = CALL get_input_hospital_code(True)
  item_code = get_input_item_code(True)
  quantity_to_distribute = get_number_input("Quantity to distribute")
  is_stock_enough, balance = CALL check_item_stock_is_enough(supplier_code,
item_code, quantity_to_distribute)
  IF is_stock_enough THEN
    CALL update_stock(supplier_code, item_code, quantity_to_distribute, "MINUS")
    WRITEFILE distribution_file_object, supplier_code, target_hospital_code,
item_code, quantity_to_distribute
    WRITEFILE "\n"
  ELSE
    PRINT "Invalid stock, remaining", balance, "only"
  ENDIF
  CLOSE distribution_file_object
ENDFUNCTION
```

Flow Chart

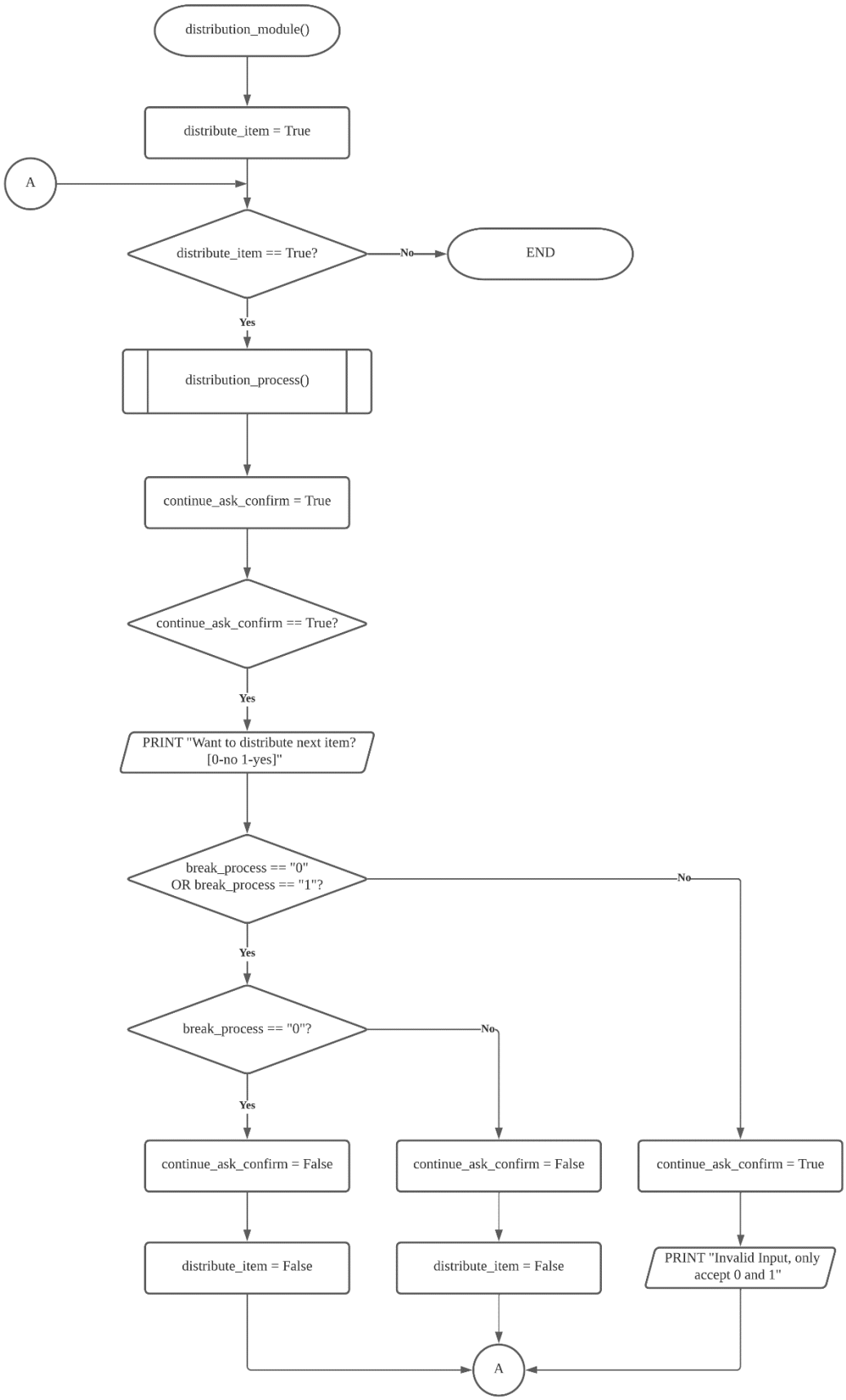

distribution_process()
distribuition_file_object = OPENFILE "distribution.txt" FOR APPEND
supplier_code = get_input_supplier_code(True)
target_hospital_code = get_input_hospital_code(True)
item_code = get_input_item_code(True)
quantity_to_distribute = get_number_input("Quantity to Distribute")
A
A
is_stock_enough, balance = check_item_stock_is_enough(supplier_code, item_code, quantity_to_distribute)
is_stock_enough == True?
No
PRINT "Invalid stock, remaining ", balance, "only"
Yes
update_stock(supplier_code, item_code, quantity_to_distribute, "MINUS")
WRITEFILE distribuition_file_object, supplier_code, target_hospital_code, item_code, quantity_to_distribute
WRITEFILE distribuition_file_object, "\n"
CLOSE distribuition_file_object
END

distribution_module

Pseudocode

```
FUNCTION distribution_module()
  distribute_item = True
  DOWHILE distribute_item
    CALL distribution_process()
    continue_ask_confirm = True
    DOWHILE continue_ask_confirm
      PRINT "Want to distribute next item? [0-no 1-yes]"
      IF break_process == "0" OR break_process == "1" THEN
        IF break_process == "0" THEN
          continue_ask_confirm = False
          distribute_item = False
        ELSE
          continue_ask_confirm = False
          distribute_item = True
        ENDIF
      ELSE
        continue_ask_confirm = True
        PRINT "Invalid Input, only accept 0 and 1"
      ENDIF
    ENDDO
  ENDDO
ENDFUNCTION
```

Flow Chart

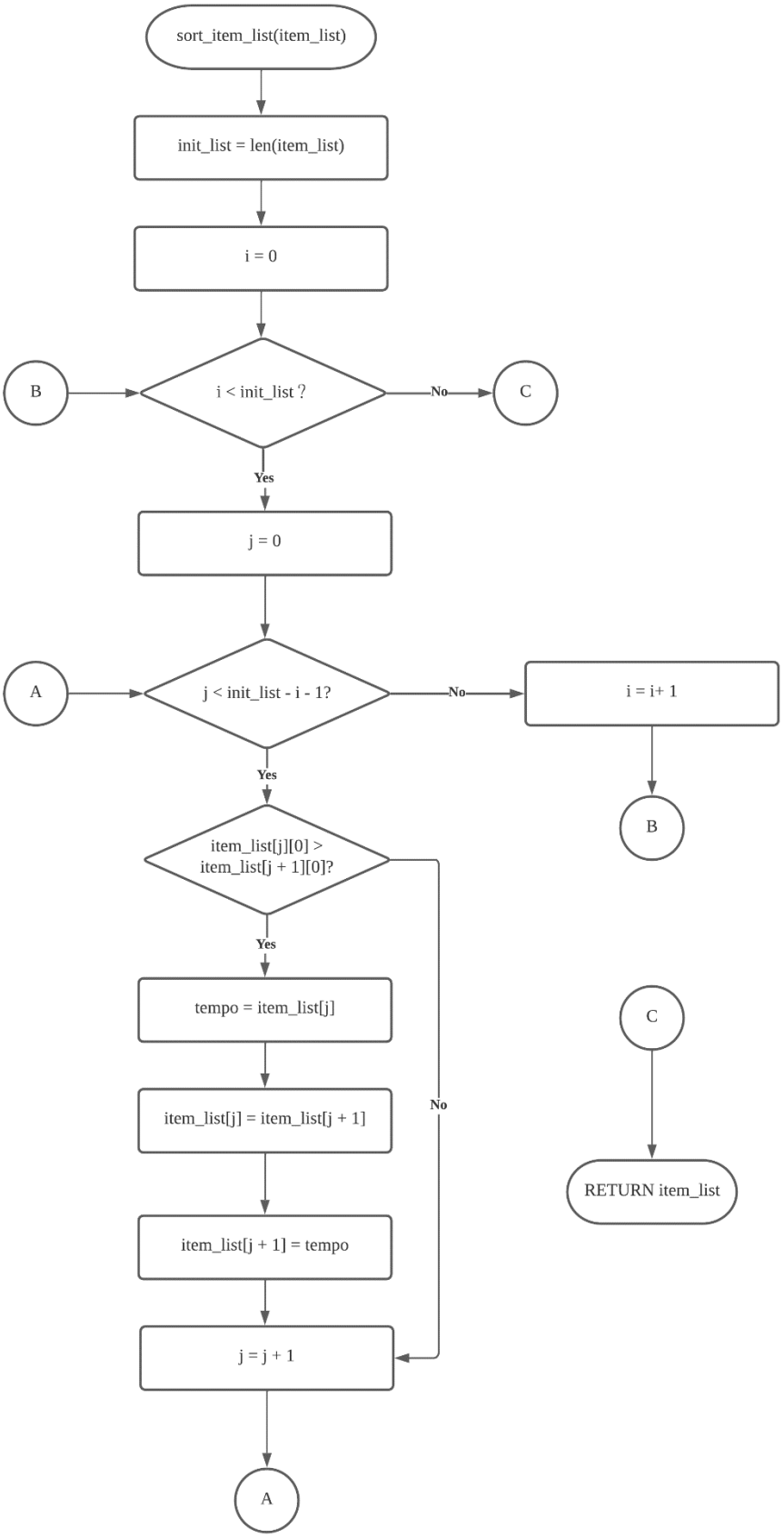

distribution_module()
distribute_item = True
A
distribute_item == True?
No
END
Yes
distribution_process()
continue_ask_confirm = True
continue_ask_confirm == True?
Yes
PRINT "Want to distribute next item? [0-no 1-yes]"
break_process == "0" OR break_process == "1"?
No
Yes
break_process == "0"?
No
Yes
continue_ask_confirm = False
continue_ask_confirm = False
continue_ask_confirm = True
distribute_item = False
distribute_item = False
PRINT "Invalid Input, only accept 0 and 1"
A

sort_item_list

Pseudocode

```
FUNCTION sort_item_list(item_list)
  init_list = LENGTH OF item_list
  LOOP i FROM 0 TO init_list STEP 1
    LOOP j FROM 0 TO init_list - i - 1 STEP 1
      IF item_list[j][0] > item_list[j + 1][0] THEN
        tempo = item_list[j]
        item_list[j] = item_list[j + 1]
        item_list[j + 1] = tempo
      ENDIF
    ENDLOOP
  ENDLOOP
  RETURN item_list
ENDFUNCTION
```

Flow Chart

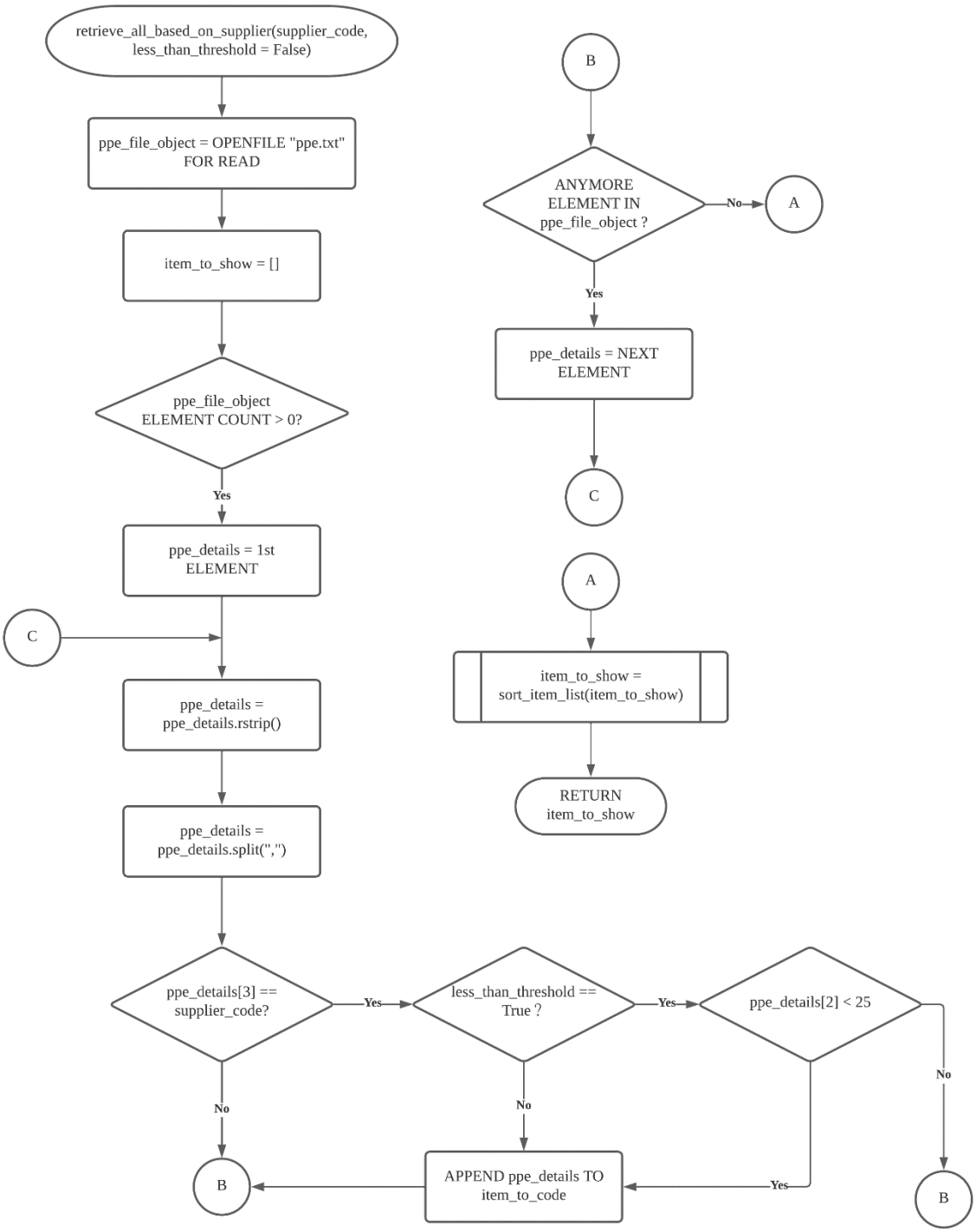

sort_item_list(item_list)
init_list = len(item_list)
i = 0
B
i < init_list ?
No
C
Yes
j = 0
A
j < init_list - i - 1?
No
i = i+ 1
B
Yes
item_list[j][0] >
item_list[j + 1][0]?
Yes
tempo = item_list[j]
item_list[j] = item_list[j + 1]
item_list[j + 1] = tempo
No
j = j + 1
A
C
RETURN item_list

retrieve_all_based_on_supplier

Pseudocode

```
FUNCTION retrieve_all_based_on_supplier(supplier_code, less_than_threshold =
False)
  ppe_file_object = OPENFILE "ppe.txt" FOR READ
  DECLARE List item_to_show
  FOREACH ppe_details IN ppe_file_object
    ppe_details = ppe_details.rstrip()
    ppe_details = ppe_details.split(",")
    IF ppe_details[3] == supplier_code THEN
      IF ppe_details[2]) < 25 THEN
        APPEND ppe_details TO item_to_show
      ENDIF
    ELSE
      APPEND ppe_details TO item_to_show
    ENDIF
  ENDFOR
  RETURN item_to_show
ENDFUNCTION
```

Flow Chart

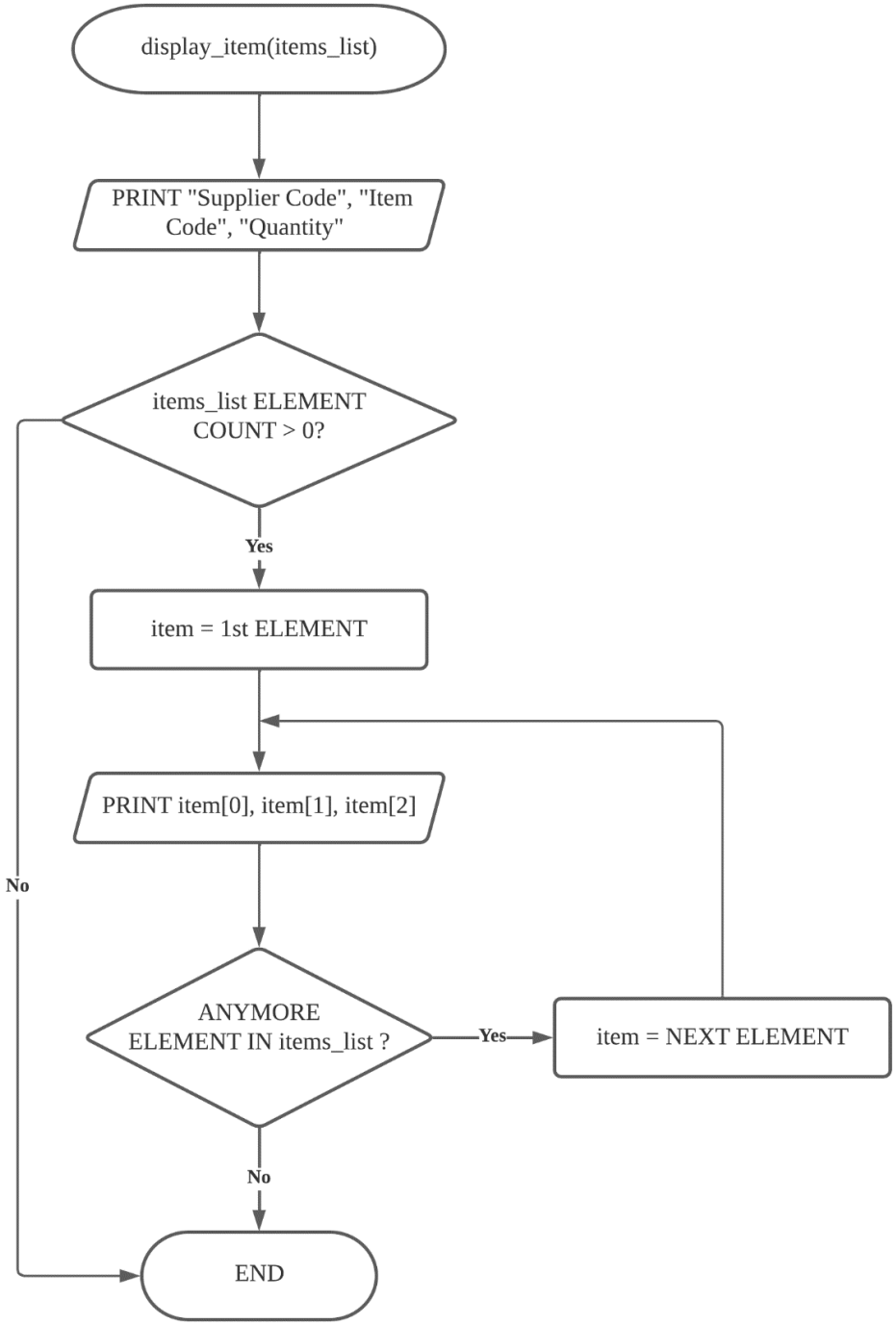

retrieve_all_based_on_supplier(supplier_code, less_than_threshold = False)
ppe_file_object = OPENFILE "ppe.txt" FOR READ
item_to_show = []
ppe_file_object ELEMENT COUNT > 0?
Yes
ppe_details = 1st ELEMENT
C
ppe_details = ppe_details.rstrip()
ppe_details = ppe_details.split(",")
ppe_details[3] == supplier_code?
Yes
less_than_threshold == True ?
Yes
ppe_details[2] < 25
No
No
No
B
APPEND ppe_details TO item_to_code
Yes
B
B
ANYMORE ELEMENT IN ppe_file_object ?
No
A
Yes
ppe_details = NEXT ELEMENT
C
A
item_to_show = sort_item_list(item_to_show)
RETURN item_to_show

display_item

Pseudocode

```
FUNCTION display_item(items_list)
  PRINT "Supplier Code", "Item Code", "Quantity"
  FOREACH item IN items_list
    PRINT item[0], item[1], item[2]
  ENDFOR
ENDFUNCTION
```

Flow Chart

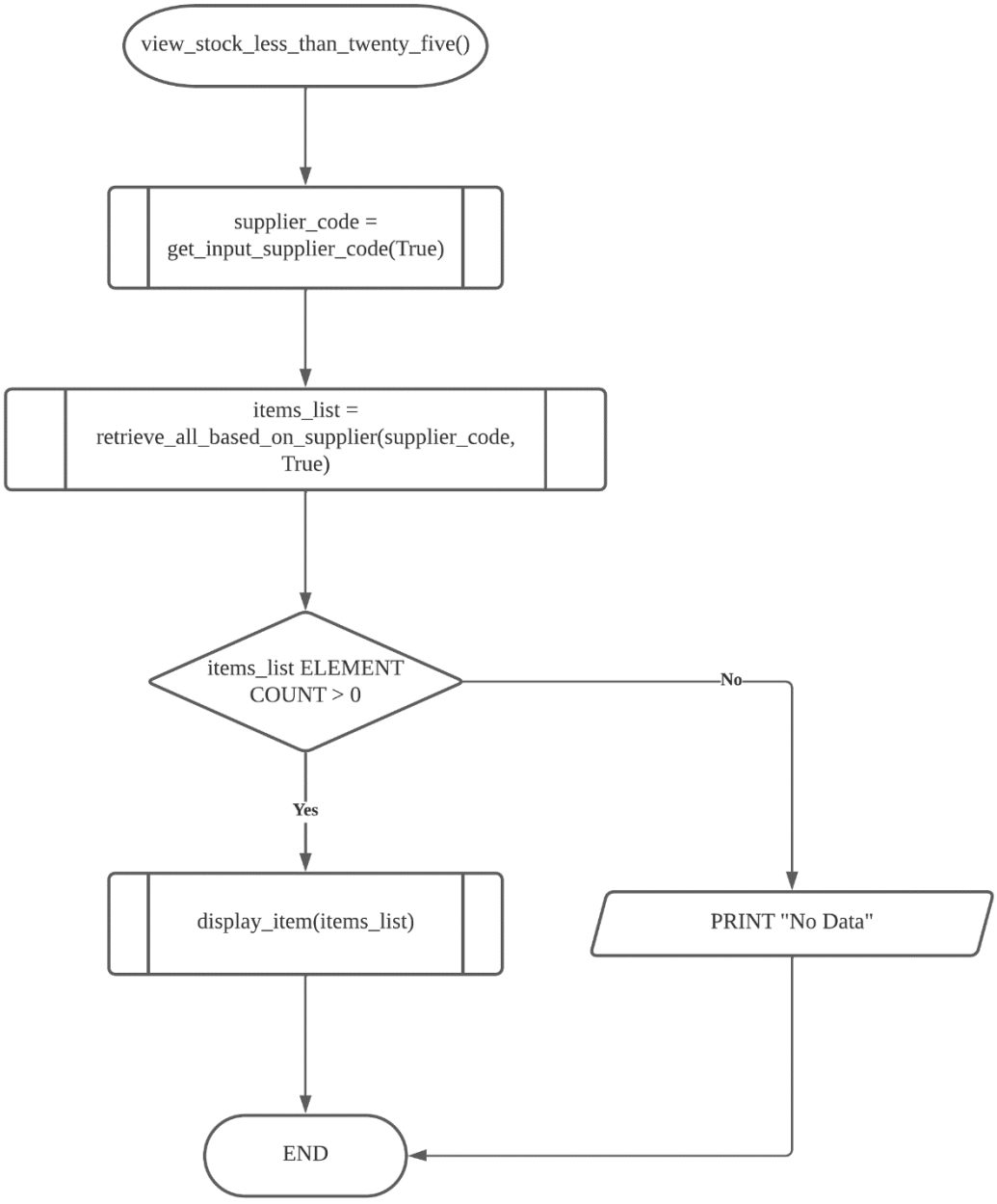


view_stock_less_than_twenty_five

Pseudocode

```
FUNCTION view_all_stock()
  supplier_code = CALL get_input_supplier_code()
  items_list = CALL retrieve_all_based_on_supplier(supplier_code, True)
  IF LENGTH OF items_list > 0 THEN
    display_item(items_list)
  ELSE
    PRINT "No Data"
  ENDIF
ENDFUNCTION
```

Flow Chart

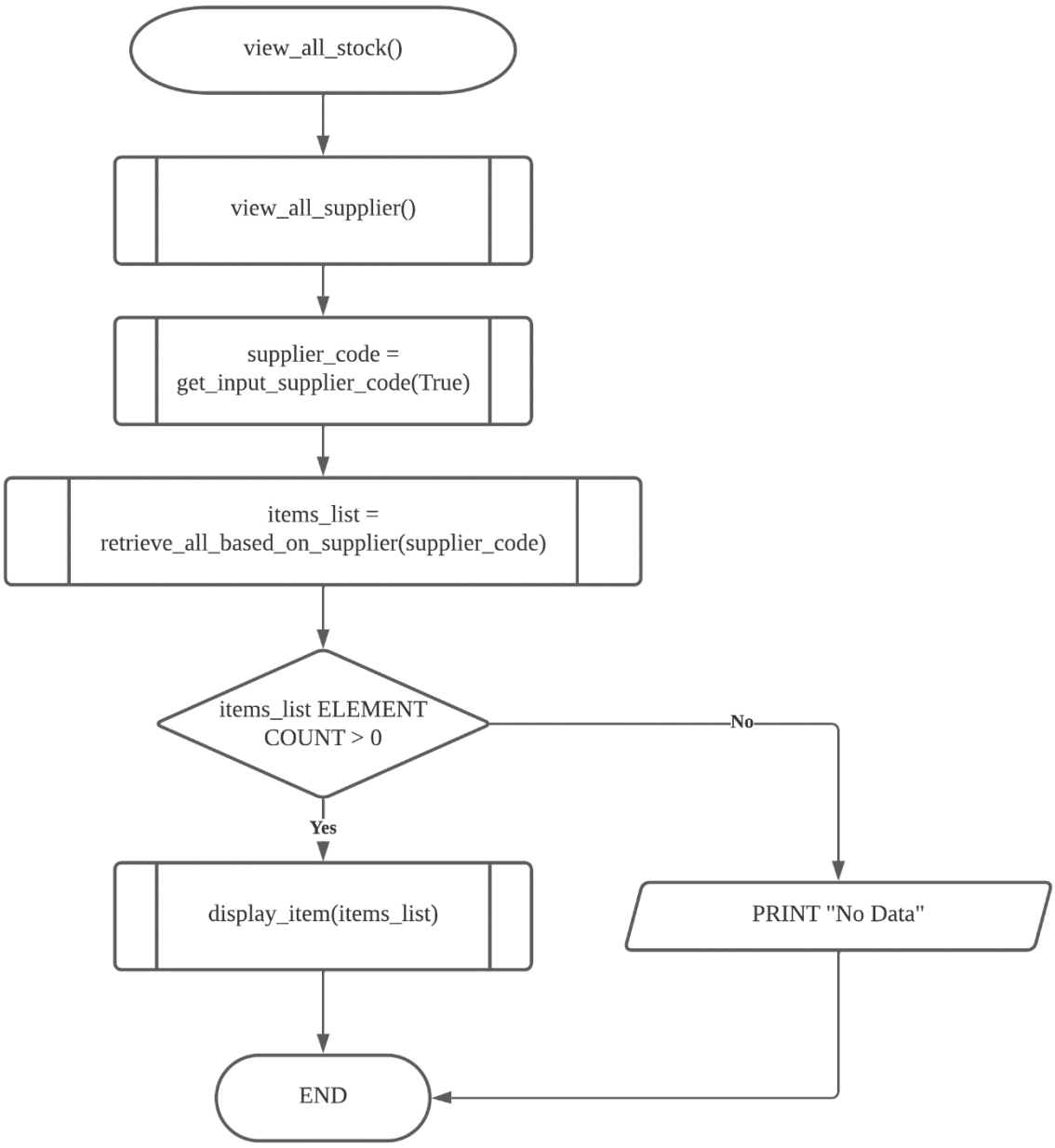


view_all_stock

Pseudocode

```
FUNCTION view_all_stock()
  CALL view_all_supplier()
  supplier_code = CALL get_input_supplier_code()
  items_list = CALL retrieve_all_based_on_supplier(supplier_code)
  IF LENGTH OF items_list > 0 THEN
    display_item(items_list)
  ELSE
    PRINT "No Data"
  ENDIF
ENDFUNCTION
```

Flow Chart

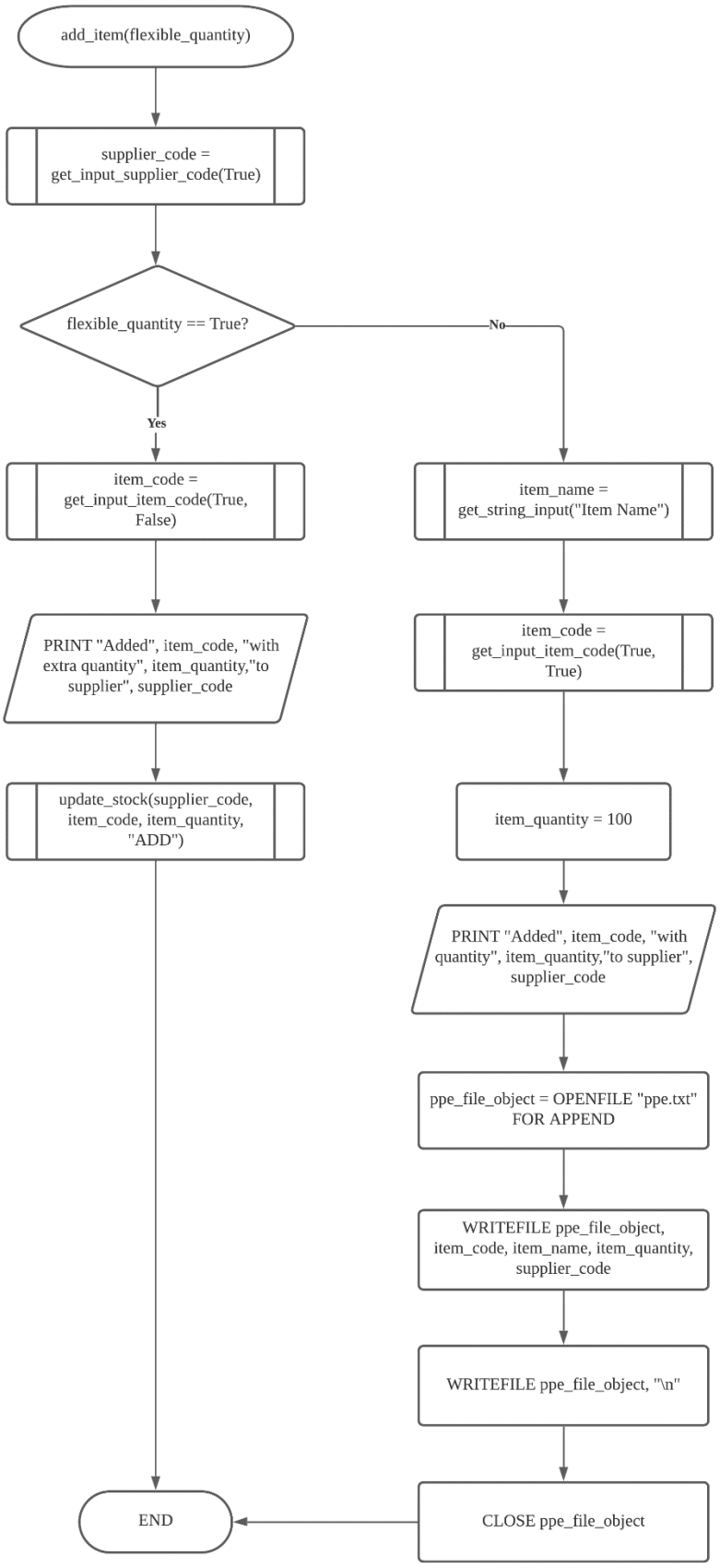


add_item

Pseudocode

```
FUNCTION add_item(flexible_quantity)
  supplier_code = CALL get_input_supplier_code(True)
  IF flexible_quantity THEN
    item_code = get_input_item_code(True, False)
    item_quantity = CALL get_number_input("Item Quantity")
    PRINT "Added", item_code, "with extra quantity",
    item_quantity, "to supplier", supplier_code
    update_stock(supplier_code, item_code, item_quantity, "ADD")
  ELSE
    item_name = get_string_input("Item Name")
    item_code = get_input_item_code(True, True)
    item_quantity = 100
    PRINT "Added", item_code, "with quantity", item_quantity,
    "to supplier", supplier_code
    ppe_file_object = OPENFILE "ppe.txt" FOR APPEND
    WRITEFILE ppe_file_object, item_code, item_name, item_quantity, supplier_code
    WRITEFILE ppe_file_object, "\n"
    CLOSE ppe_file_object
  ENDIF
ENDFUNCTION
```

Flow Chart

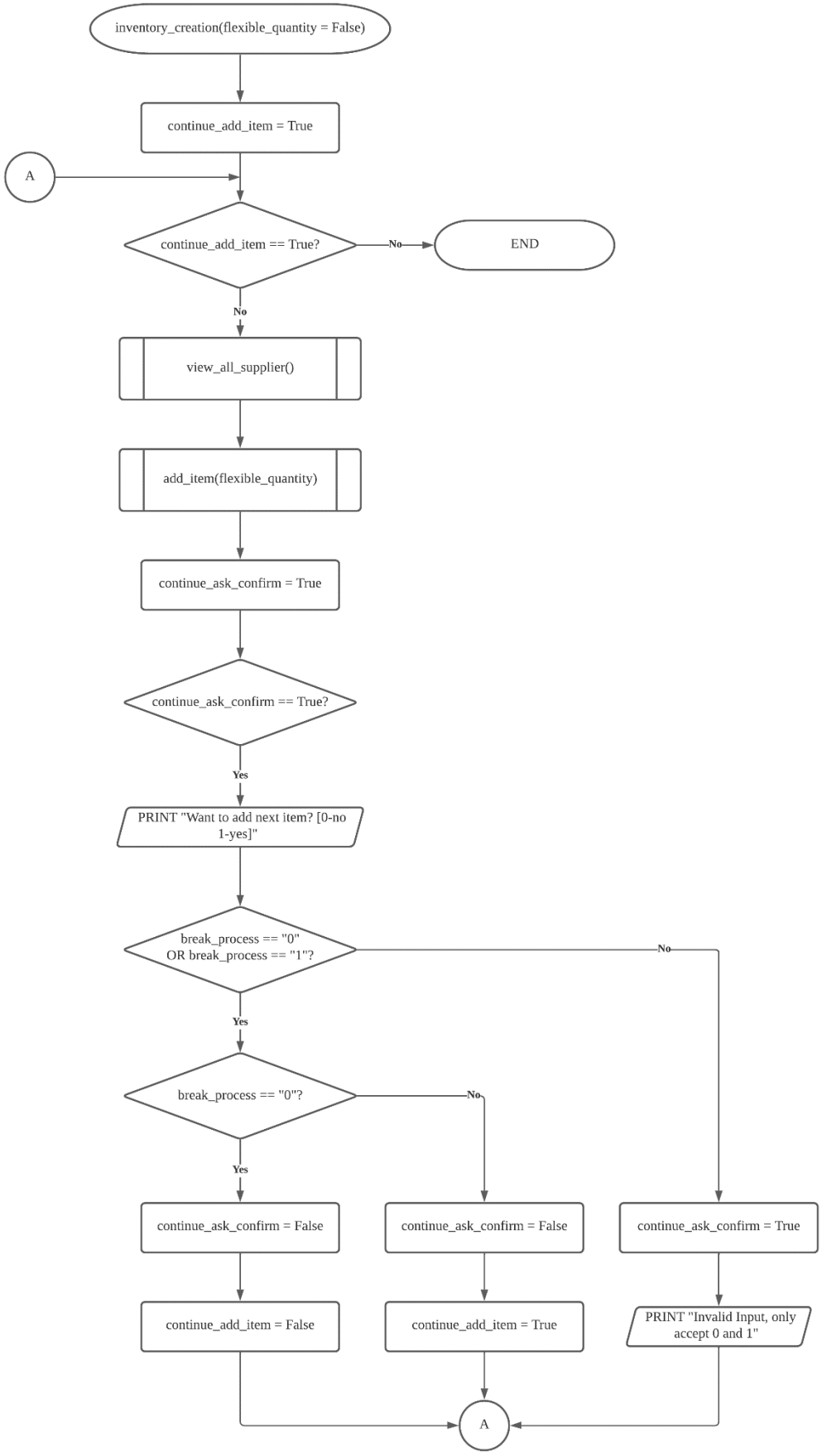

add_item(flexible_quantity)
supplier_code = get_input_supplier_code(True)
flexible_quantity == True?
No
Yes
item_code = get_input_item_code(True, False)
item_name = get_string_input("Item Name")
PRINT "Added", item_code, "with extra quantity", item_quantity,"to supplier", supplier_code
item_code = get_input_item_code(True, True)
update_stock(supplier_code, item_code, item_quantity, "ADD")
item_quantity = 100
PRINT "Added", item_code, "with quantity", item_quantity,"to supplier", supplier_code
ppe_file_object = OPENFILE "ppe.txt" FOR APPEND
WRITEFILE ppe_file_object, item_code, item_name, item_quantity, supplier_code
WRITEFILE ppe_file_object, "\n"
CLOSE ppe_file_object
END

inventory_creation

Pseudocode

```
FUNCTION inventory_creation(flexible_quantity = False)
  IF not flexible_quantity THEN
    file_object = OPENFILE “ppe.txt” FOR WRITE
    CLOSE file_object
  ENDIF
  continue_add_item = True
  DOWHILE continue_add_item
    CALL view_all_supplier()
    CALL add_item(flexible_quantity)
    continue_ask_confirm = True
    DOWHILE continue_ask_confirm
      PRINT "Want to distribute next item? [0-no 1-yes]"
      IF break_process == "0" OR break_process == "1" THEN
        IF break_process == "0" THEN
          continue_ask_confirm = False
          continue_add_item = False
        ELSE
          continue_ask_confirm = False
          continue_add_item = True
        ENDIF
      ELSE
        continue_ask_confirm = True
        PRINT "Invalid Input, only accept 0 and 1"
      ENDIF
    ENDDO
  ENDDO
ENDFUNCTION
```

Flow Chart

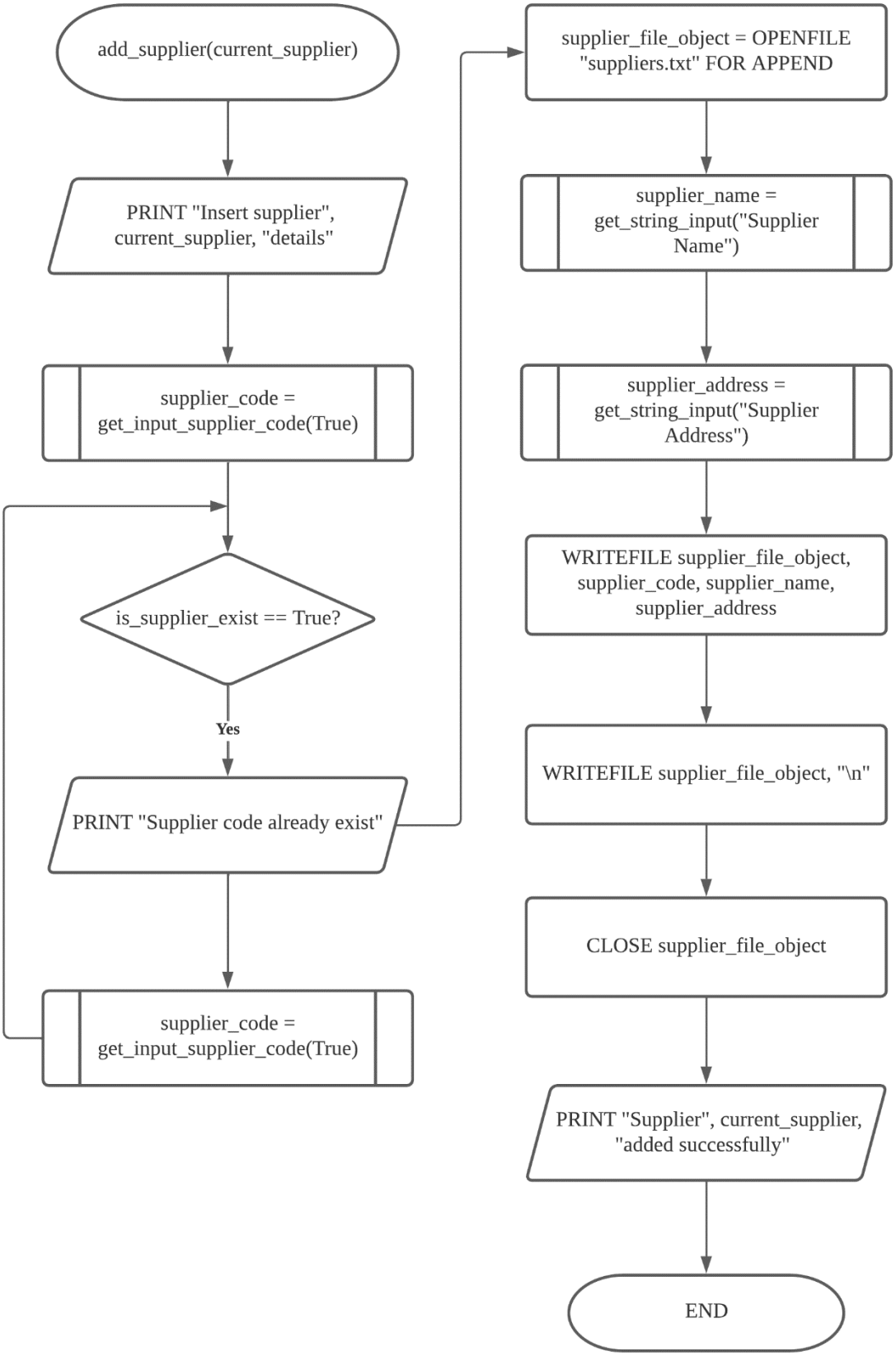

inventory_creation(flexible_quantity = False)
continue_add_item = True
A
continue_add_item == True?
No
END
No
view_all_supplier()
add_item(flexible_quantity)
continue_ask_confirm = True
continue_ask_confirm == True?
Yes
PRINT "Want to add next item? [0-no 1-yes]"
break_process == "0" OR break_process == "1"?
No
Yes
break_process == "0"?
No
Yes
continue_ask_confirm = False
continue_add_item = False
continue_ask_confirm = False
continue_add_item = True
continue_ask_confirm = True
PRINT "Invalid Input, only accept 0 and 1"
A

add_supplier

Pseudocode

```
FUNCTION add_supplier(current_supplier)
  PRINT "Insert supplier", current_supplier,"details"
  supplier_code = CALL get_input_supplier_code()
  DOWHILE True
    is_supplier_exist = CALL check_is_code_exist(supplier_code, "SUPPLIER")
    IF is_supplier_exist == True THEN
      PRINT "Supplier code already exist"
      supplier_code = CALL get_input_supplier_code()
    ELSE
      BREAK
    ENDIF
  ENDDO
  supplier_file_object = OPENFILE "suppliers.txt" FOR APPEND
  supplier_name = CALL get_string_input("Supplier Name")
  supplier_address = CALL get_string_input("Supplier Address")
  WRITEFILE supplier_file_object, supplier_code, supplier_name, supplier_address
  WRITEFILE supplier_file_object "\n"
  CLOSE supplier_file_object
  PRINT "Supplier", current_supplier,"added successfully\n"
ENDFUNCTION
```

Flow Chart

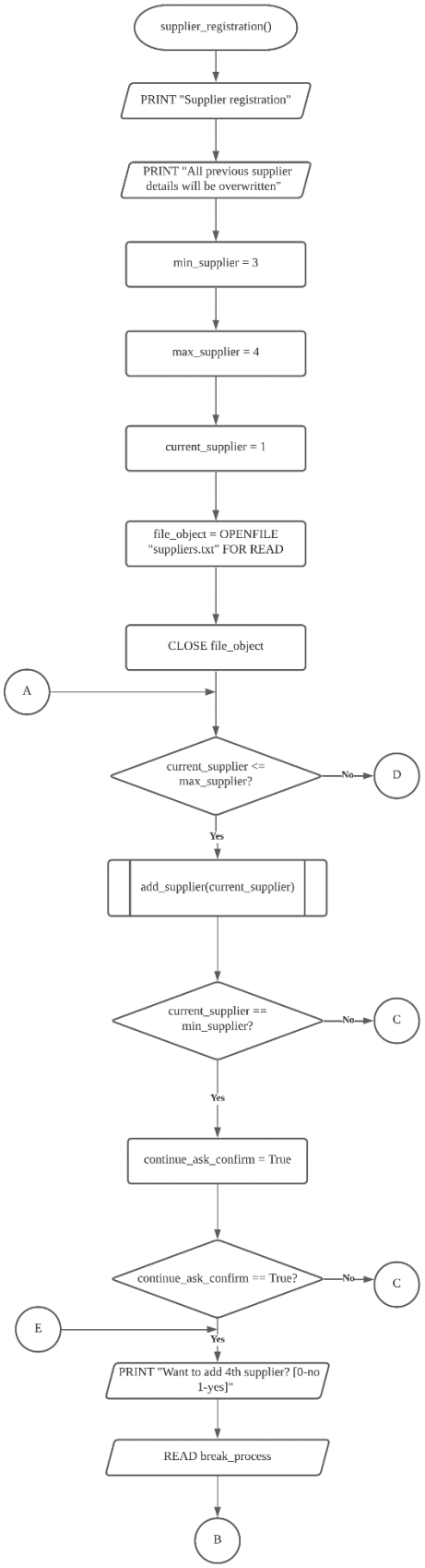

add_supplier(current_supplier)
PRINT "Insert supplier", current_supplier, "details"
supplier_code = get_input_supplier_code(True)
is_supplier_exist == True?
Yes
PRINT "Supplier code already exist"
supplier_code = get_input_supplier_code(True)
supplier_file_object = OPENFILE "suppliers.txt" FOR APPEND
supplier_name = get_string_input("Supplier Name")
supplier_address = get_string_input("Supplier Address")
WRITEFILE supplier_file_object, supplier_code, supplier_name, supplier_address
WRITEFILE supplier_file_object, "\n"
CLOSE supplier_file_object
PRINT "Supplier", current_supplier, "added successfully"
END

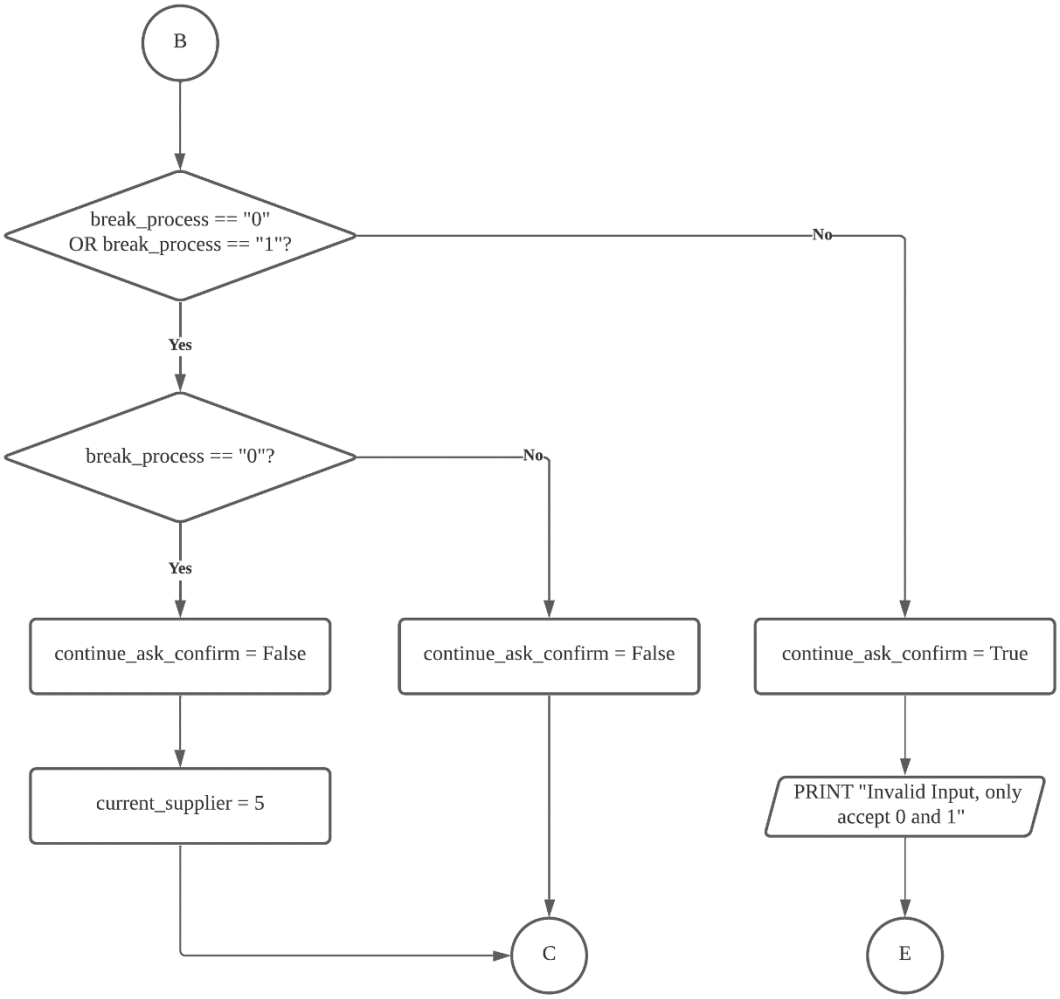
supplier_registration

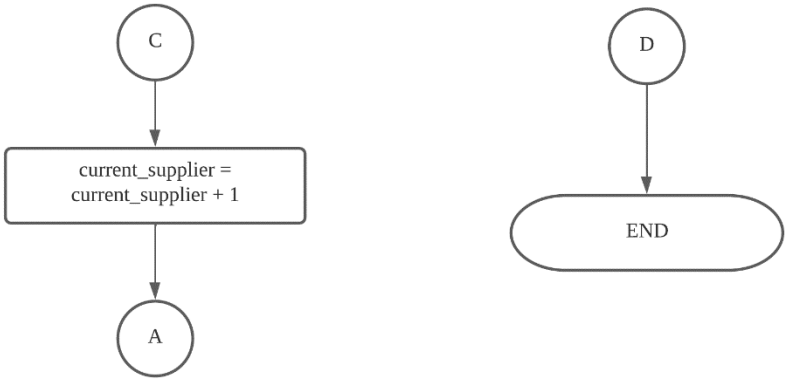
Pseudocode

```
FUNCTION supplier_registration()
  PRINT "Supplier Registration"
  PRINT "All previous supplier details will be overwritten"
  min_supplier = 3
  max_supplier = 4
  current_supplier = 1
  file_object = OPENFILE "suppliers.txt" FOR WRITE
  file_object.close()
  DOWHILE current_supplier <= max_supplier
    CALL add_supplier(current_supplier)
    IF current_supplier == min_supplier THEN
      continue_ask_confirm = True
      DOWHILE continue_ask_confirm
        PRINT "Want to add 4th supplier? [0-no 1-yes]"
        IF break_process == "0" OR break_process == "1" THEN
          IF break_process == "0" THEN
            continue_ask_confirm = False
            current_supplier = 5
          ELSE
            continue_ask_confirm = False
          ENDIF
        ELSE
          continue_ask_confirm = True
          PRINT "Invalid Input, only accept 0 and 1"
        ENDIF
      ENDDO
    ENDIF
    current_supplier = current_supplier + 1
  ENDDO
ENDFUNCTION
```

Flow Chart


supplier_registration()
PRINT "Supplier registration"
PRINT "All previous supplier details will be overwritten"
min_supplier = 3
max_supplier = 4
current_supplier = 1
file_object = OPENFILE "suppliers.txt" FOR READ
CLOSE file_object
A
current_supplier <= max_supplier?
No
D
Yes
add_supplier(current_supplier)
current_supplier == min_supplier?
No
C
Yes
continue_ask_confirm = True
continue_ask_confirm == True?
No
C
E
Yes
PRINT "Want to add 4th supplier? [0-no 1-yes]"
READ break_process
B


B
break_process == "0"
OR break_process == "1"?
No
Yes
break_process == "0"?
No
Yes
continue_ask_confirm = False
continue_ask_confirm = False
continue_ask_confirm = True
current_supplier = 5
PRINT "Invalid Input, only accept 0 and 1"
C
E

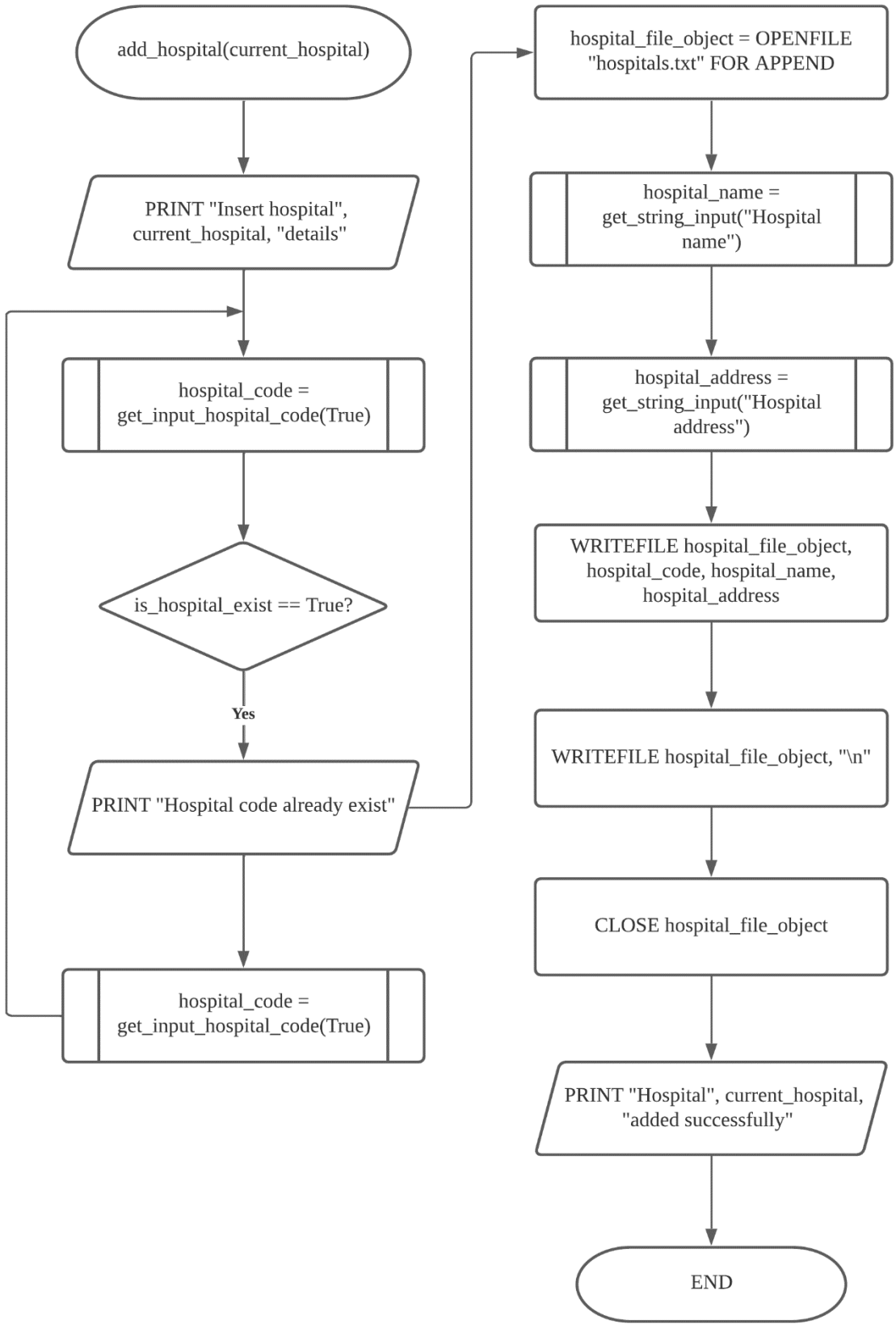

C
current_supplier = current_supplier + 1
A
D
END

add_hospital

Pseudocode

```
FUNCTION add_hospital(current_hospital)
  PRINT "Insert hospital", current_hospital,"details"
  hospital_code = CALL get_input_hospital_code()
  DOWHILE True
    is_hospital_exist = CALL check_is_code_exist(hospital_code, "HOSPITAL")
    IF is_hospital_exist == True THEN
      PRINT "Hospital code already exist"
      hospital_code = CALL get_input_hospital_code()
    ELSE
      BREAK
    ENDIF
  ENDDO
  hospital_file_object = OPENFILE "hospitals.txt" FOR APPEND
  hospital_name = CALL get_string_input("Hospital name")
  hospital_address = CALL get_string_input("Hospital address")
  WRITEFILE hospital_file_object, hospital_code, hospital_name, hospital_address
  WRITEFILE hospital_file_object "\n"
  CLOSE hospital_file_object
  PRINT "Hospital", current_hospital,"added successfully\n"
ENDFUNCTION
```

Flow Chart

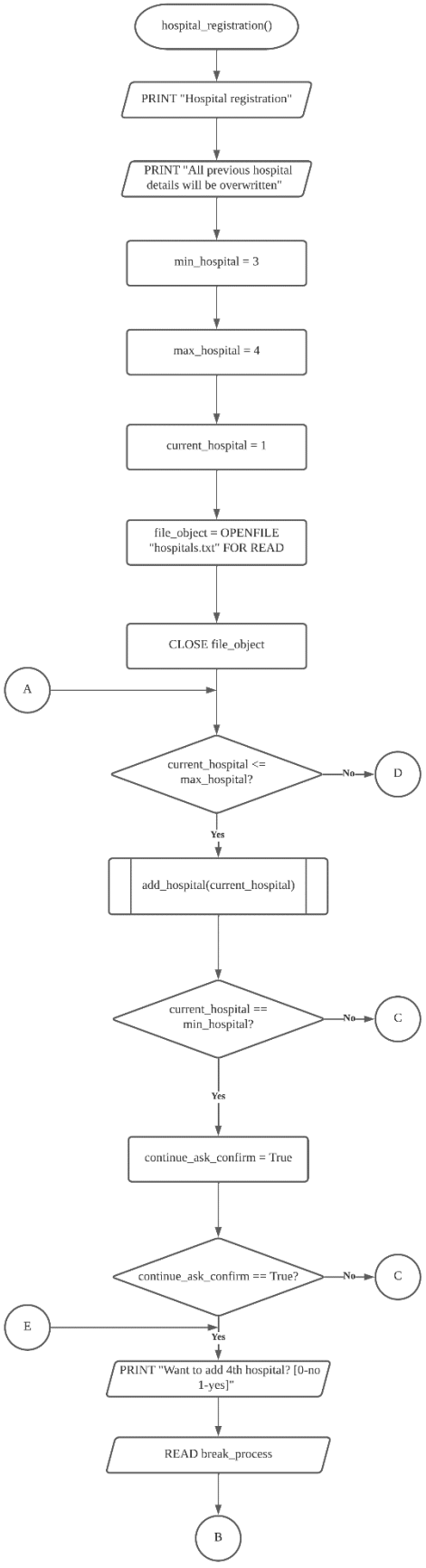

add_hospital(current_hospital)
PRINT "Insert hospital", current_hospital, "details"
hospital_code = get_input_hospital_code(True)
is_hospital_exist == True?
Yes
PRINT "Hospital code already exist"
hospital_code = get_input_hospital_code(True)
hospital_file_object = OPENFILE "hospitals.txt" FOR APPEND
hospital_name = get_string_input("Hospital name")
hospital_address = get_string_input("Hospital address")
WRITEFILE hospital_file_object, hospital_code, hospital_name, hospital_address
WRITEFILE hospital_file_object, "\n"
CLOSE hospital_file_object
PRINT "Hospital", current_hospital, "added successfully"
END

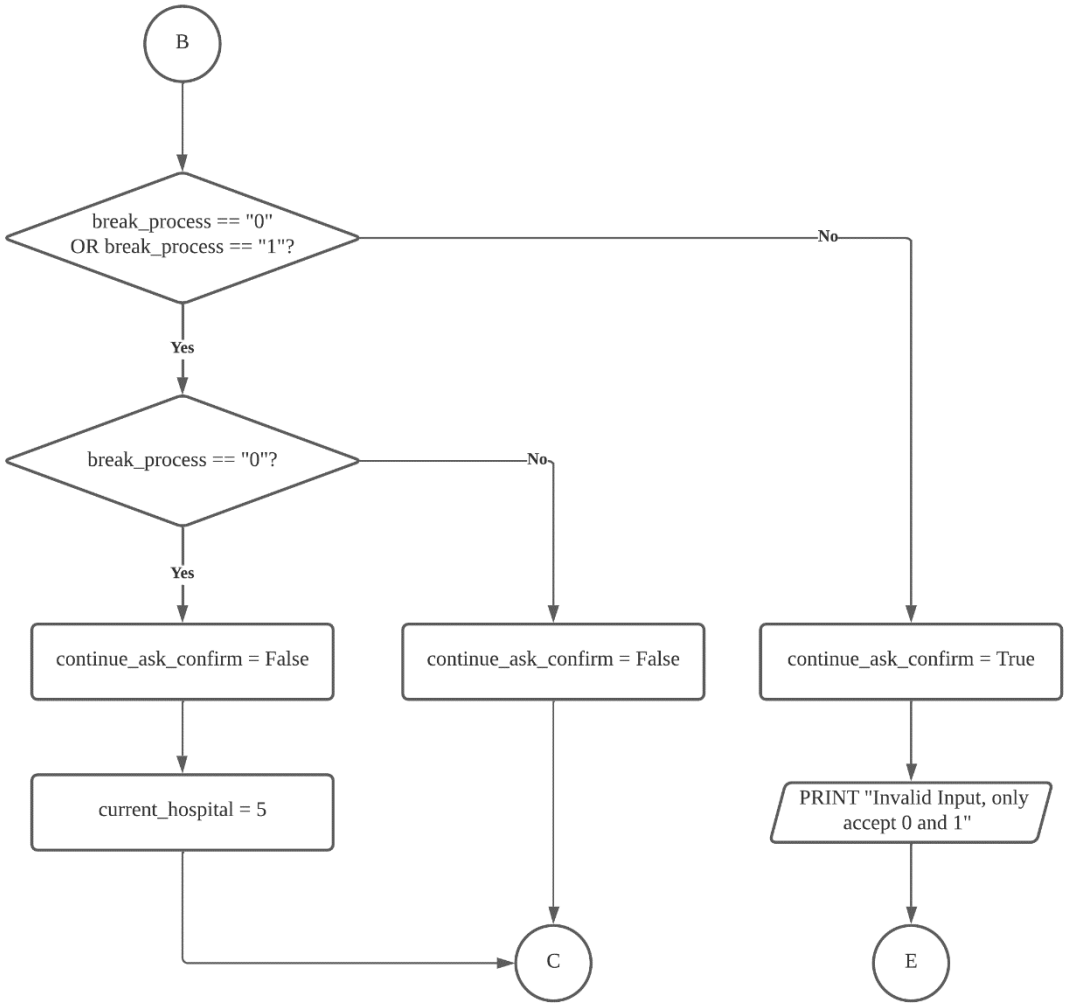
hospital_registration

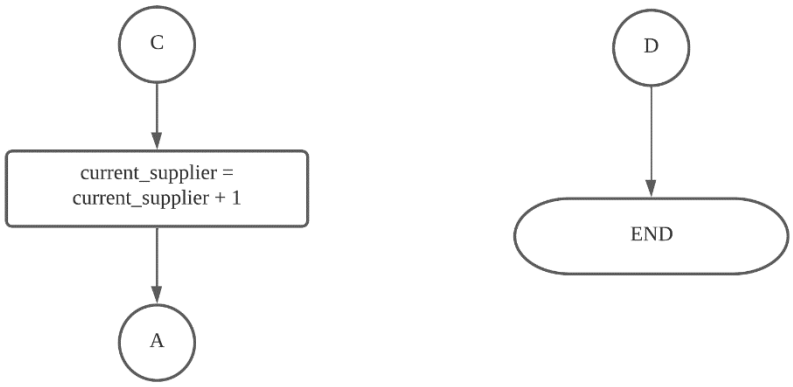
Pseudocode

```
FUNCTION hospital_registration()
  PRINT "Hospital registration"
  PRINT "All previous hospital details will be overwritten"
  min_hospital = 3
  max_hospital = 4
  current_hospital = 1
  file_object = OPENFILE "hospitals.txt" FOR WRITE
  file_object.close()
  DOWHILE current_hospital <= max_hospital
    CALL add_hospital(current_hospital)
    IF current_hospital == min_hospital THEN
      continue_ask_confirm = True
      DOWHILE continue_ask_confirm
        PRINT "Want to add 4th hospital? [0-no 1-yes]"
        IF break_process == "0" OR break_process == "1" THEN
          IF break_process == "0" THEN
            continue_ask_confirm = False
            current_supplier = 5
          ELSE
            continue_ask_confirm = False
          ENDIF
        ELSE
          continue_ask_confirm = True
          PRINT "Invalid Input, only accept 0 and 1"
        ENDIF
      ENDDO
    ENDIF
    current_hospital = current_hospital + 1
  ENDDO
ENDFUNCTION
```

Flow Chart


hospital_registration()
PRINT "Hospital registration"
PRINT "All previous hospital details will be overwritten"
min_hospital = 3
max_hospital = 4
current_hospital = 1
file_object = OPENFILE "hospitals.txt" FOR READ
CLOSE file_object
A
current_hospital <= max_hospital?
No
D
Yes
add_hospital(current_hospital)
current_hospital == min_hospital?
No
C
Yes
continue_ask_confirm = True
continue_ask_confirm == True?
No
C
E
Yes
PRINT "Want to add 4th hospital? [0-no 1-yes]"
READ break_process
B


B
break_process == "0"
OR break_process == "1"?
No
Yes
break_process == "0"?
No
Yes
continue_ask_confirm = False
continue_ask_confirm = False
continue_ask_confirm = True
current_hospital = 5
PRINT "Invalid Input, only accept 0 and 1"
C
E

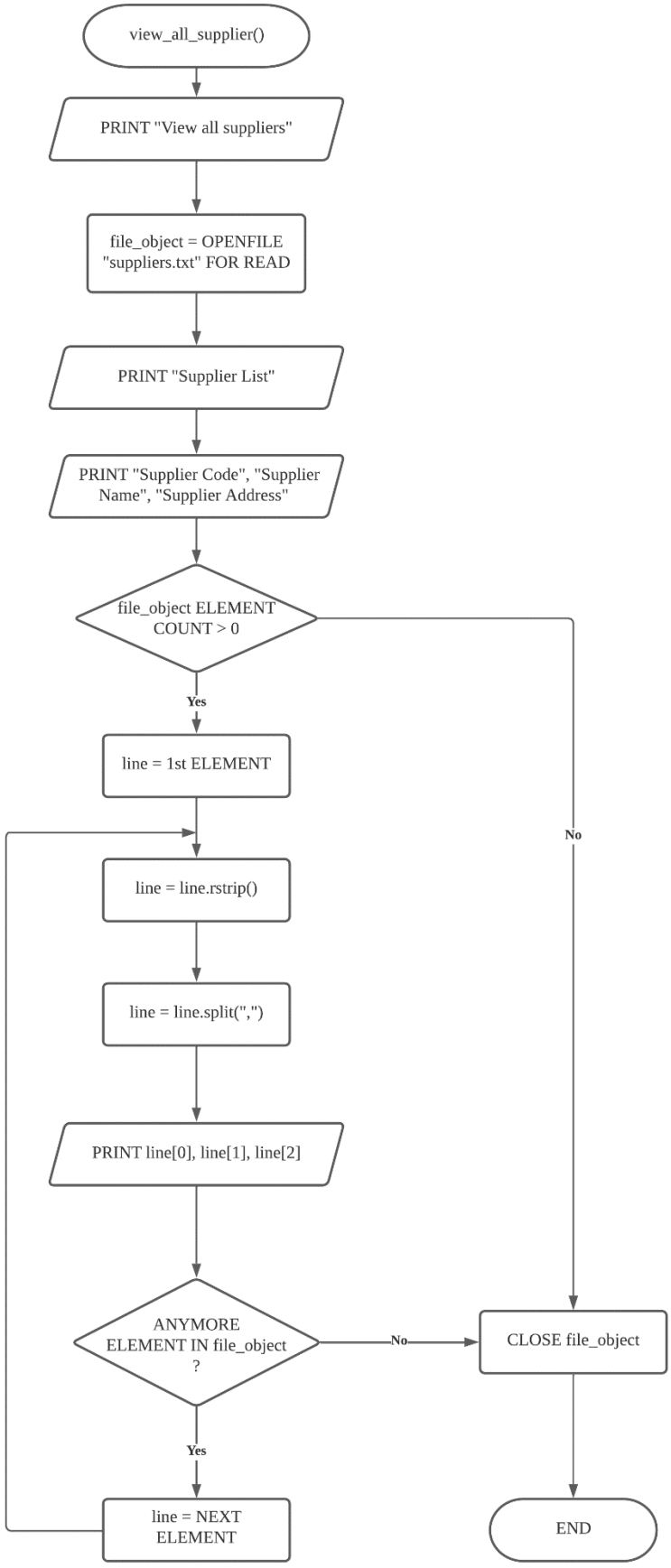

C
current_supplier =
current_supplier + 1
A
D
END

view_all_supplier

Pseudocode

```
FUNCTION view_all_supplier()
  PRINT "View all suppliers"
  file_object = OPENFILE "hospitals.txt" FOR WRITE
  PRINT "Suppliers list"
  PRINT "Supplier Code", "Supplier Name", "Supplier Address"
  FOREACH line IN file_object
    line = line.rstrip()
    line = line.split(",")
    PRINT line[0], line[1], line[2]
  ENDFOR
  CLOSE file_object
ENDFUNCTION
```

Flow Chart

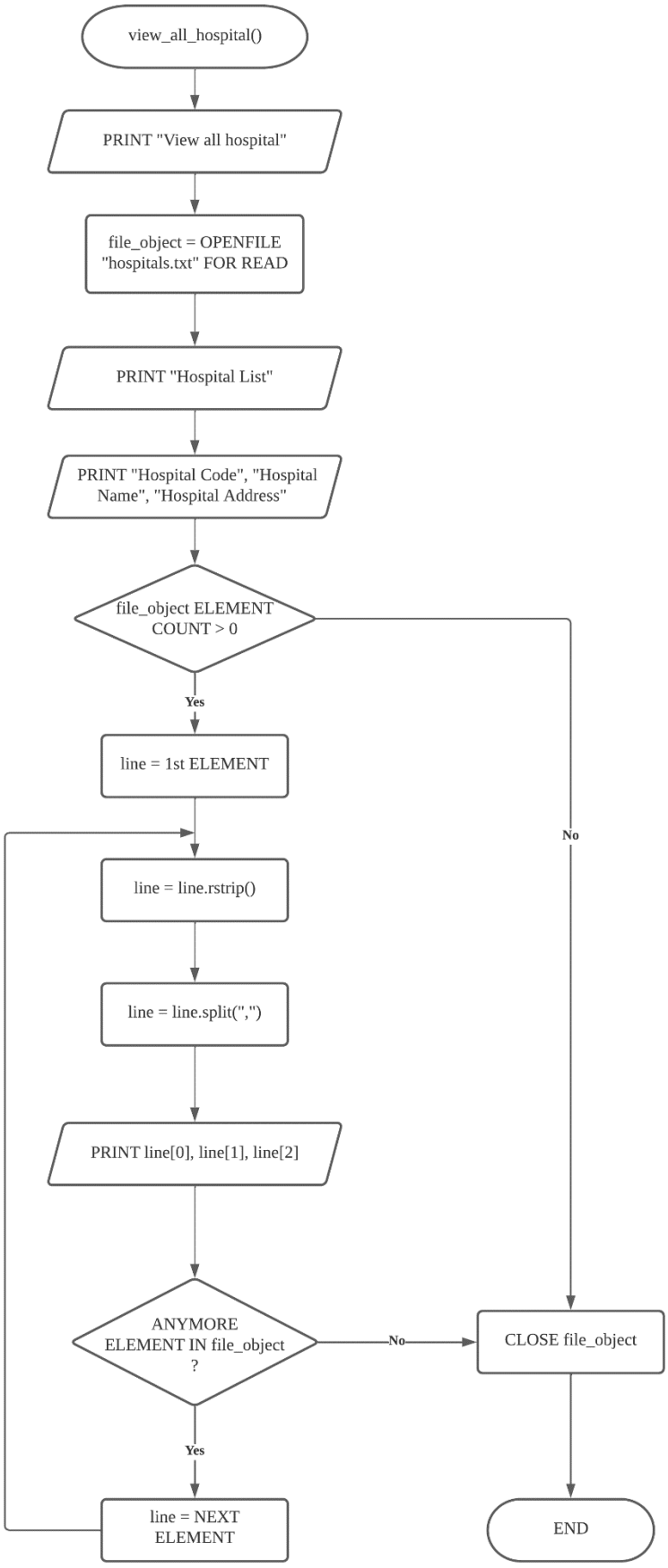

view_all_supplier()
PRINT "View all suppliers"
file_object = OPENFILE "suppliers.txt" FOR READ
PRINT "Supplier List"
PRINT "Supplier Code", "Supplier Name", "Supplier Address"
file_object ELEMENT COUNT > 0
Yes
No
line = 1st ELEMENT
line = line.rstrip()
line = line.split(",")
PRINT line[0], line[1], line[2]
ANYMORE ELEMENT IN file_object ?
No
Yes
CLOSE file_object
line = NEXT ELEMENT
END

view_all_hospital

Pseudocode

```
FUNCTION view_all_hospital()
  PRINT "View all hospital"
  file_object = OPENFILE "hospitals.txt" FOR WRITE
  PRINT "Hospital list"
  PRINT "Hospital Code", "Hospital Name", "Hospital Address"
  FOREACH line IN file_object
    line = line.rstrip()
    line = line.split(",")
    PRINT line[0], line[1], line[2]
  ENDFOR
  CLOSE file_object
ENDFUNCTION
```

Flow Chart

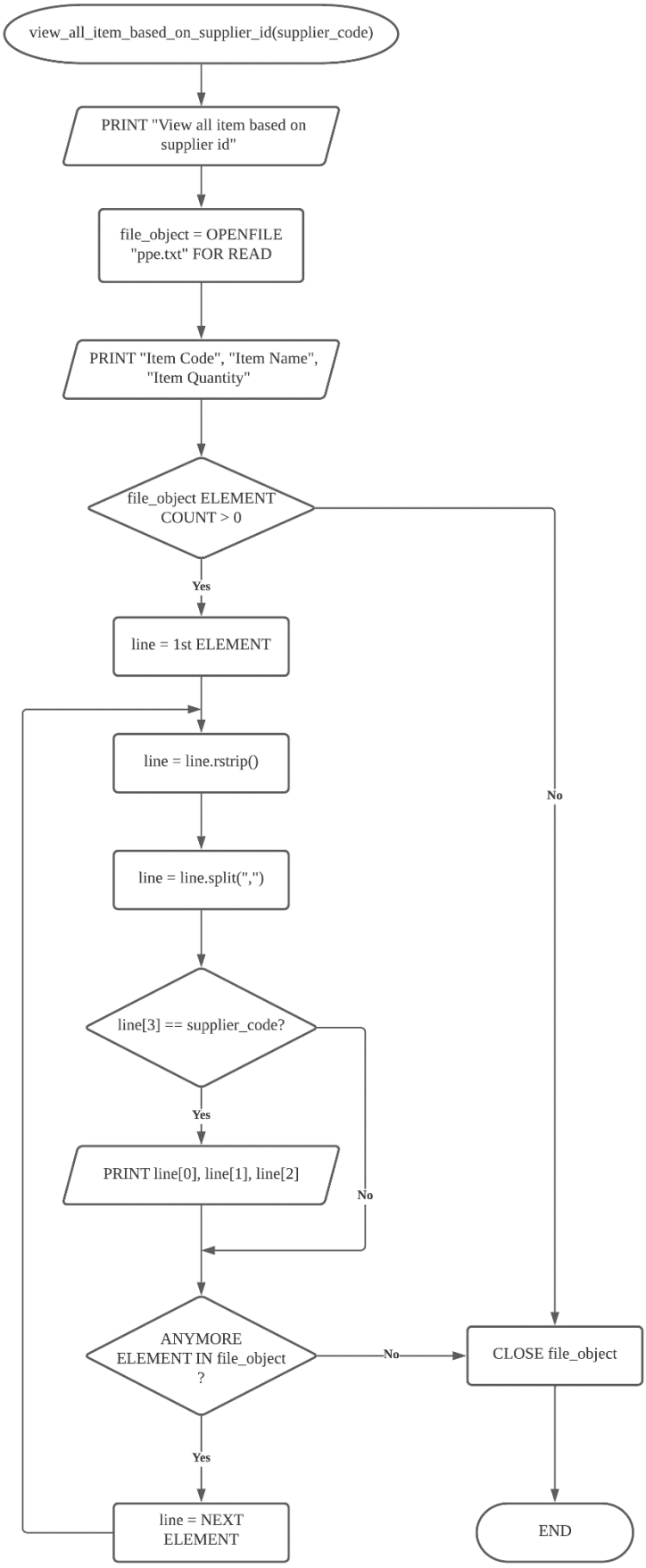

view_all_hospital()
PRINT "View all hospital"
file_object = OPENFILE "hospitals.txt" FOR READ
PRINT "Hospital List"
PRINT "Hospital Code", "Hospital Name", "Hospital Address"
file_object ELEMENT COUNT > 0
Yes
No
line = 1st ELEMENT
line = line.rstrip()
line = line.split(",")
PRINT line[0], line[1], line[2]
ANYMORE ELEMENT IN file_object ?
No
Yes
line = NEXT ELEMENT
CLOSE file_object
END

view_all_item_based_on_supplier_code

Pseudocode

```
FUNCTION view_all_item_based_on_supplier_id(supplier_code)
  PRINT "View all item based on supplier id"
  file_object = OPENFILE "ppe.txt" FOR WRITE
  PRINT "Item Code", "Item Name", "Item Address"
  FOREACH line IN file_object
    line = line.rstrip()
    line = line.split(",")
    IF line[3] == supplier_code THEN
      PRINT line[0], line[1], line[2]
    ENDIF
  ENDFOR
  CLOSE file_object
ENDFUNCTION
```

Flow Chart


view_all_item_based_on_supplier_id(supplier_code)
PRINT "View all item based on supplier id"
file_object = OPENFILE "ppe.txt" FOR READ
PRINT "Item Code", "Item Name", "Item Quantity"
file_object ELEMENT COUNT > 0
Yes
No
line = 1st ELEMENT
line = line.rstrip()
line = line.split(",")
line[3] == supplier_code?
Yes
No
PRINT line[0], line[1], line[2]
ANYMORE ELEMENT IN file_object ?
No
Yes
CLOSE file_object
line = NEXT ELEMENT
END

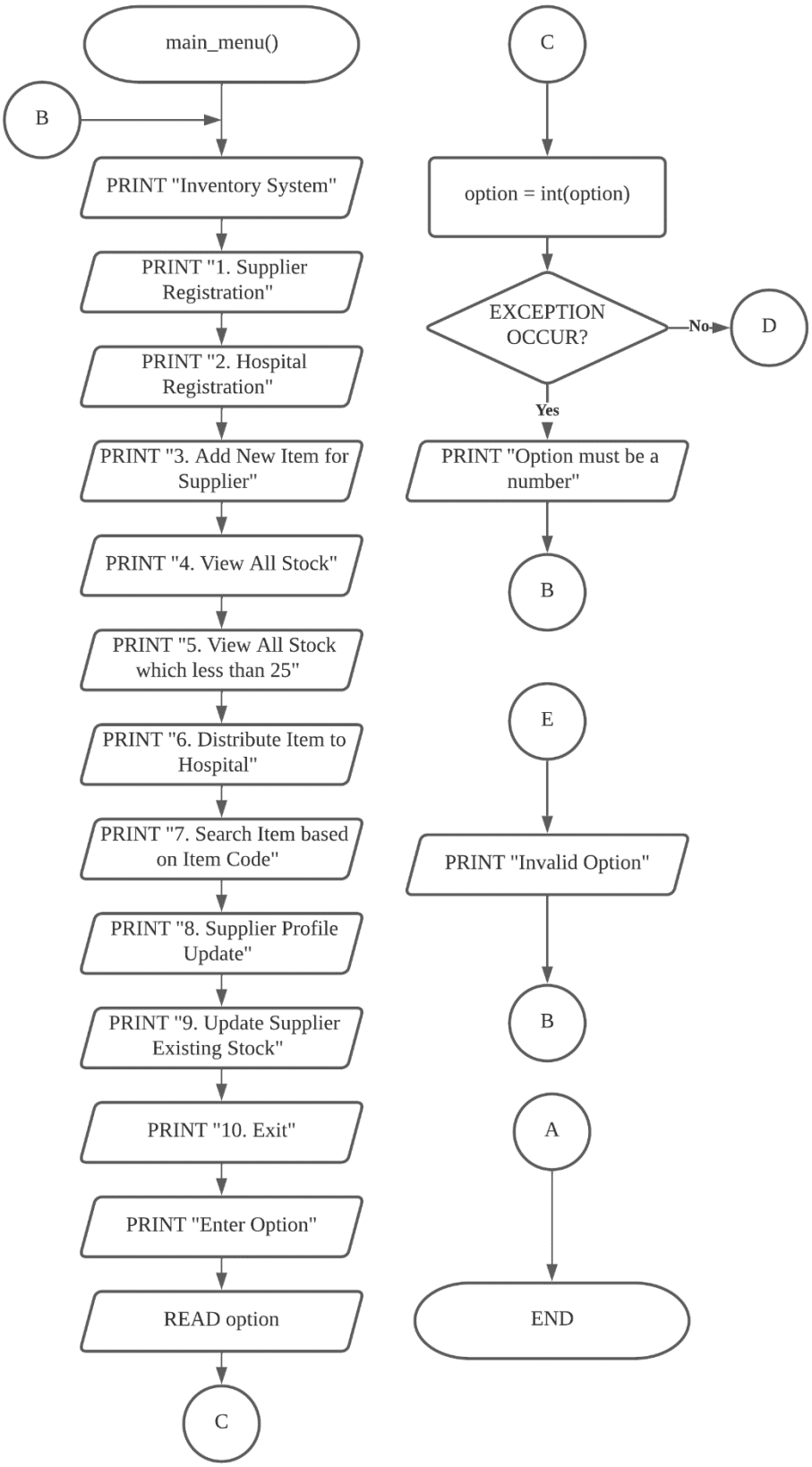
main_menu

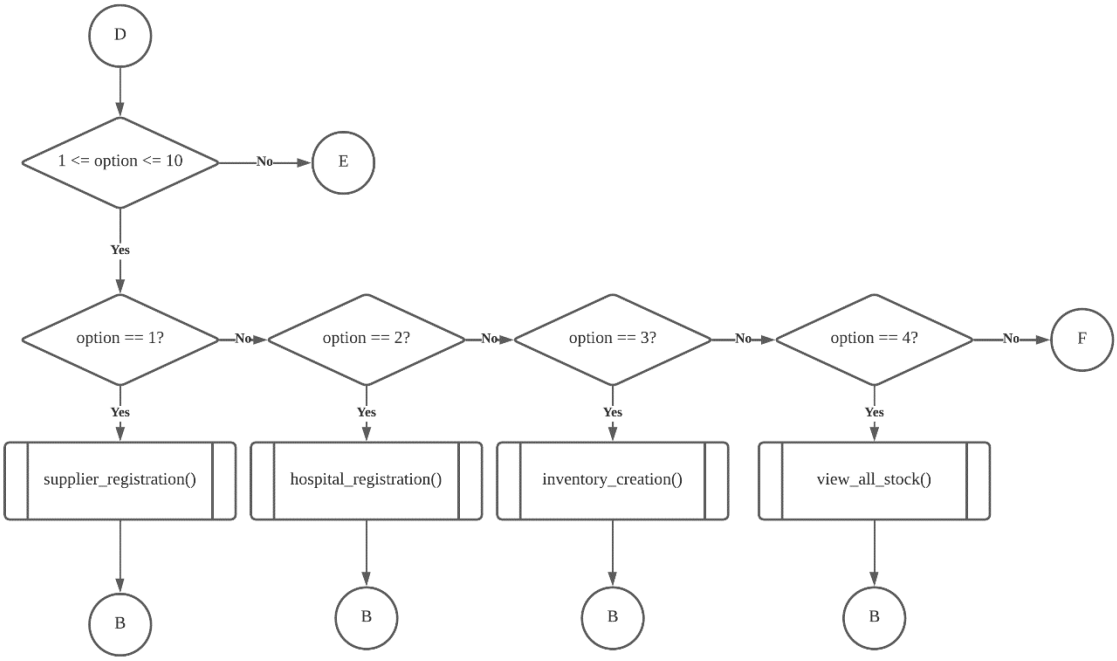
Pseudocode

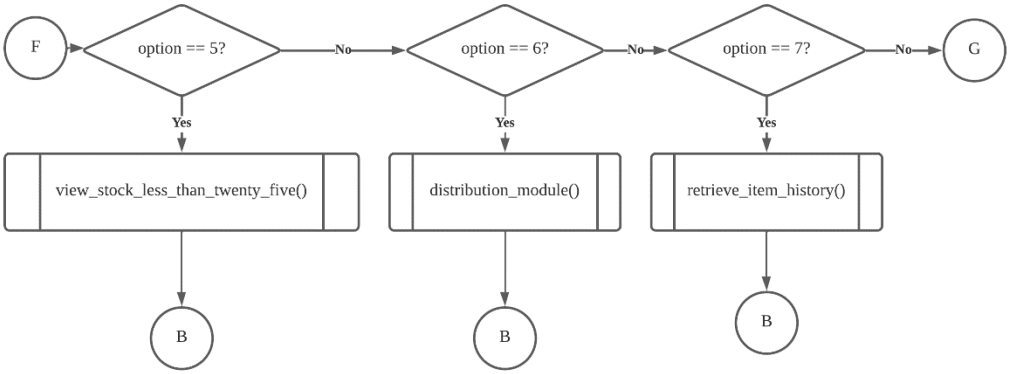
```
FUNCTION main_menu()
  DOWHILE True
    PRINT "Inventory System"
    PRINT "1. Supplier Registration"
    PRINT "2. Hospital Registration"
    PRINT "3. Add New Item for Supplier"
    PRINT "4. View All Stock"
    PRINT "5. View Stock which less than 25"
    PRINT "6. Distribute Item to Hospital"
    PRINT "7. Search Item based on Item Code"
    PRINT "8. Supplier Profile Update"
    PRINT "9. Update Supplier Existing Stock"
    PRINT "10. Exit"
    DOWHILE True
      TRY
        PRINT "Enter Option"
          READ option
        IF 1 <= option <= 10 THEN
        IF option == 1 THEN
          supplier_registration()
        ELSE
          IF option == 2 THEN
            hospital_registration()
          ELSE
            IF option == 3 THEN
              inventory_creation()
            ELSE
              IF option == 4 THEN
                view_all_stock()
              ELSE
                IF option == 5 THEN
                  view_stock_less_than_twenty_five()
                ELSE
                  IF option == 6 THEN
                    view_stock_less_than_twenty_five()
                  ELSE
                    IF option == 7 THEN
                      retrieve_item_history()
                    ELSE
                      IF option == 8 THEN
                        update_supplier_details()
                      ELSE
                        IF option == 9 THEN
                          inventory_creation()
                        ELSE
                          IF option == 10 THEN
                            PRINT "Bye"
                            BREAK
                          ELSE
                            PRINT "Invalid Option"
                          ENDIF
                        ENDIF
                      ENDIF
                    ENDIF
                  ENDIF
                ENDIF
```

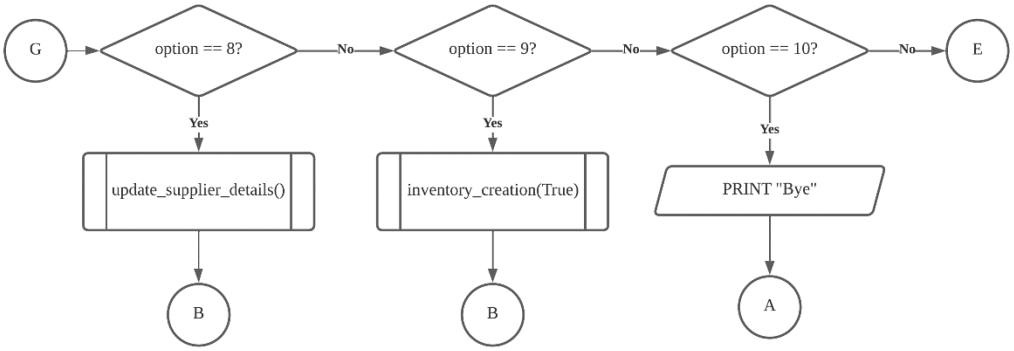
```
            ENDIF
          ENDIF
        ENDIF
      EXCEPT
        PRINT "Option must be a number"
    ENDDO
  ENDDO
ENDFUNCTION
```

Flow Chart


main_menu()
B
PRINT "Inventory System"
PRINT "1. Supplier Registration"
PRINT "2. Hospital Registration"
PRINT "3. Add New Item for Supplier"
PRINT "4. View All Stock"
PRINT "5. View All Stock which less than 25"
PRINT "6. Distribute Item to Hospital"
PRINT "7. Search Item based on Item Code"
PRINT "8. Supplier Profile Update"
PRINT "9. Update Supplier Existing Stock"
PRINT "10. Exit"
PRINT "Enter Option"
READ option
C
C
option = int(option)
EXCEPTION OCCUR?
No
D
Yes
PRINT "Option must be a number"
B
E
PRINT "Invalid Option"
B
A
END


D
1 <= option <= 10
No
E
Yes
option == 1?
No
option == 2?
No
option == 3?
No
option == 4?
No
F
Yes
Yes
Yes
Yes
supplier_registration()
hospital_registration()
inventory_creation()
view_all_stock()
B
B
B
B


F
option == 5?
No
option == 6?
No
option == 7?
No
G
Yes
Yes
Yes
view_stock_less_than_twenty_five()
distribution_module()
retrieve_item_history()
B
B
B

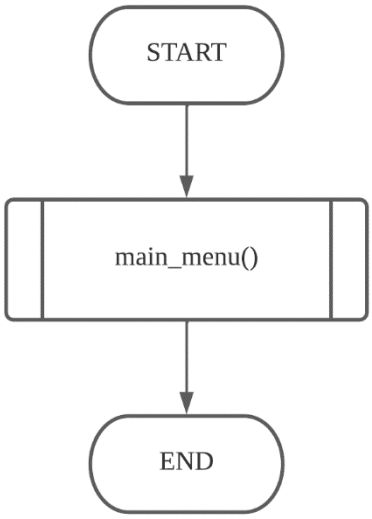

G
option == 8?
No
option == 9?
No
option == 10?
No
E
Yes
Yes
Yes
update_supplier_details()
inventory_creation(True)
PRINT "Bye"
B
B
A

main

Pseudocode

```
PROGRAM inventory_system
BEGIN
  CALL main_menu()
END
```

Flow Chart



Program Source Code and Explanation

All the business logic in every function will be explained in this section with source code.

check_is_code_exist

```
def check_is_code_exist(target_code, target):
    """
    Function to check if supplier/hospital/item code is valid
    :param target_code: supplier code/hospital code/item code
    :param target: HOSPITAL/SUPPLIER/ITEM
    :return: return True if exist, False if not
    """
    if target == "SUPPLIER":
        file_object = open("suppliers.txt", "r")
    elif target == "HOSPITAL":
        file_object = open("hospitals.txt", "r")
    elif target == "ITEM":
        file_object = open("ppe.txt", "r")
    else:
        print("Invalid target")
        return False
    for details in file_object:
        details = details.rstrip()
        details = details.split(",")
        if details[0] == target_code:
            file_object.close()
            return True
    file_object.close()
    return False
```

In function *check_is_code_exist* function, it take target_code and target as parameter. *target* parameter is used to let the system know which file needs to open, while *target_code* is the value that we need to check whether it exists system will split the string using a comma, loop the file line by line and search if the target_code is found. If found, this function will return True and False if it doesn't find anything.

get_input_supplier_code

```
def get_input_supplier_code(check_is_exist = False):
    """
    Function to ask valid supplier code input
    :param check_is_exist: boolean, check if supplier code exist
    :return: return supplier code
    """
    supplier_code = input("Enter Supplier Code: ")
    while True:
        if supplier_code == "":
            print("Supplier code cannot be empty")
            supplier_code = input("Enter supplier code: ")
        else:
            if check_is_exist:
                is_valid_supplier = check_is_code_exist(supplier_code, "SUPPLIER")
                if is_valid_supplier:
                    return supplier_code
                else:
                    print("Supplier not found")
                    supplier_code = input("Enter Supplier Code: ")
            else:
                return supplier_code
```

In function *get_input_supplier_code*, it can make sure the user is submitting a valid string and it can check whether the supplier_code inserted exists in record.

get_string_input

```
def get_string_input(title):
    """
    Function to ask valid string input
    :param title: title
    :return: valid input from user
    """
    string_input = input("Enter {}: ".format(title))
    while True:
        if string_input == "":
            print("{} cannot be empty".format(title))
            string_input = input("Enter {}: ".format(title))
        else:
            return string_input
```

In function *get_string_input*, it can make sure users submit a valid string and print the question based on the *title* parameter value.

update_supplier_details

```
def update_supplier_details():
    """
    This function will update the supplier details
    """
    view_all_supplier()
    supplier_code = get_input_supplier_code()
    supplier_new_name = get_string_input("New Supplier Name")
    supplier_new_address = get_string_input("New Supplier Address")

    initial_supplier_file_object = open("suppliers.txt", "r")
    initial_supplier_file_object_list = []

    for initial_supplier_details in initial_supplier_file_object:
        initial_supplier_details = initial_supplier_details.rstrip()
        initial_supplier_details = initial_supplier_details.split(",")
        supplier_code_to_save = supplier_code
        if initial_supplier_details[0] == supplier_code:
            supplier_name_to_save = supplier_new_name
            supplier_address_to_save = supplier_new_address
        else:
            supplier_name_to_save = initial_supplier_details[1]
            supplier_address_to_save = initial_supplier_details[2]
        initial_supplier_file_object_list.append("{},{},{}".format(supplier_code_to_save, supplier_name_to_save,
                                                                   supplier_address_to_save))

    temp_supplier_file_object = open("suppliers.txt", "w")
    for initial_supplier_file_item in initial_supplier_file_object_list:
        temp_supplier_file_object.write(initial_supplier_file_item + "\n")
    temp_supplier_file_object.write(
        "{},{},{}".format(supplier_code_to_save, supplier_name_to_save, supplier_address_to_save))
    temp_supplier_file_object.write("\n")

    initial_supplier_file_object.close()
    temp_supplier_file_object.close()
    print("Profile Updated Successfully")
```

This function will ask for new supplier details and updates based on *supplier_code*.

sort_and_filter_item_search

```
def sort_and_filter_item_search(supplier_code_list, target_hospital_list, quantity_list, item_code_list):
    """
    This function will sort and filter the item list
    :param supplier_code_list: List of Supplier Code
    :param target_hospital_list:  List of Hospital Code
    :param quantity_list: List of Quantity Code
    :param item_code_list: List if Item Code
    """
    supplier_code_list_to_show = []
    target_hospital_list_to_show = []
    quantity_list_to_show = []
    item_code_list_to_show = []
    while len(supplier_code_list) > 0:
        ref1 = supplier_code_list.pop(0)
        ref2 = target_hospital_list.pop(0)
        ref3 = quantity_list.pop(0)
        ref4 = item_code_list.pop(0)
        j = 0
        while j < len(supplier_code_list):
            data1 = supplier_code_list[j]
            data2 = target_hospital_list[j]
            data3 = quantity_list[j]
            data4 = item_code_list[j]
            if data1 == ref1 and data2 == ref2 and data4 == ref4:
                ref3 += data3
                supplier_code_list.pop(j)
                target_hospital_list.pop(j)
                quantity_list.pop(j)
                item_code_list.pop(j)
                continue
            j += 1
        supplier_code_list_to_show.append(ref1)
        target_hospital_list_to_show.append(ref2)
        quantity_list_to_show.append(ref3)
        item_code_list_to_show.append(ref4)

    print("{:<15} {:<15} {:<15} {:<15}".format("Supplier Code", "Target Hospital", "Quantity", "Item Code"))
    index = 0
    for supplier_code in supplier_code_list_to_show:
        print("{:<15} {:<15} {:<15} {:<15}".format(supplier_code, target_hospital_list_to_show[index],
                                                   quantity_list_to_show[index], item_code_list_to_show[index]))
        index = index + 1
```

This function will combine the number of items from the same supplier and display it.

get_input_item_code

```
def get_input_item_code(check_is_exist = False, add_item = False):
    """
    Function to ask valid item code input
    :param check_is_exist: boolean, check if item code exist
    :return: return item code
    """
    item_code = input("Enter item code: ")
    while True:
        if item_code == "":
            print("Item code cannot be empty")
            item_code = input("Enter item code: ")
        else:
            if check_is_exist:
                is_valid_item = check_is_code_exist(item_code, "ITEM")
                if is_valid_item:
                    if add_item:
                        print("This item already assigned to another supplier")
                        item_code = input("Enter Item Code: ")
                    else:
                        return item_code
                else:
                    if add_item:
                        return item_code
                    print("Item not found")
                    item_code = input("Enter Item Code: ")

            else:
                return item_code
```

In function *get_input_item_code*, it can make sure the user is submitting a valid string and it can check whether the item_code inserted exists in the record. Besides that, it can check whether the item is assigned to any supplier.

retrieve_item_history

```
def retrieve_item_history():
    """
    This function will retrieve the distributed item and display the history
    """
    view_all_supplier()
    supplier_code = get_input_supplier_code()
    view_all_item_based_on_supplier_id(supplier_code)
    item_code = get_input_item_code(True)
    distribution_file_object = open("distribution.txt", "r")
    supplier_code_list = []
    target_hospital_list = []
    item_code_list = []
    quantity_list = []
    for distribution_details in distribution_file_object:
        distribution_details = distribution_details.rstrip()
        distribution_details = distribution_details.split(",")
        if distribution_details[0] == supplier_code and distribution_details[2] == item_code:
            supplier_code_list.append(distribution_details[0])
            target_hospital_list.append(distribution_details[1])
            item_code_list.append(distribution_details[2])
            quantity_list.append(int(distribution_details[3]))
    sort_and_filter_item_search(supplier_code_list, target_hospital_list, quantity_list, item_code_list)
```

This function will retrieve the distribution details based on supplier_code and item_code. All the details will append into multiple lists, and all the lists will pass to surt_and filter_item_search to perform filtering and sorting.

update_stock

```
def update_stock(supplier_code, item_code, new_quantity, action):
    """
    This function will update the stock of the item based on confition
    :param supplier_code: supplier code
    :param item_code: item code
    :param new_quantity: quantity to be updated
    :param action: MINUS/ADD
    :return:
    """
    initial_ppe_file_object = open("ppe.txt", "r")
    initial_ppe_file_object_list = []
    if action == "MINUS":
        for initial_item_details in initial_ppe_file_object:
            initial_item_details = initial_item_details.rstrip()
            initial_item_details = initial_item_details.split(",")
            if initial_item_details[3] == supplier_code and initial_item_details[0] == item_code:
                final_quantity = str(int(initial_item_details[2]) - new_quantity)
            else:
                final_quantity = initial_item_details[2]
            initial_ppe_file_object_list.append("{},{},{},{}".format(initial_item_details[0], initial_item_details[1],
                                                                     final_quantity, supplier_code))

        temp_ppe_file_object = open("ppe.txt", "w")
        for initial_ppe_file_item in initial_ppe_file_object_list:
            temp_ppe_file_object.write(initial_ppe_file_item + "\n")

    elif action == "ADD":
        for initial_item_details in initial_ppe_file_object:
            initial_item_details = initial_item_details.rstrip()
            initial_item_details = initial_item_details.split(",")
            if initial_item_details[3] == supplier_code and initial_item_details[0] == item_code:
                initial_ppe_file_object_list.append("{},{},{},{}".format(initial_item_details[0], initial_item_details[1],
                                    str(int(initial_item_details[2]) + new_quantity), supplier_code))
            else:
                initial_ppe_file_object_list.append(
                    "{},{},{},{}".format(initial_item_details[0], initial_item_details[1], initial_item_details[2],
                                         supplier_code))

        temp_ppe_file_object = open("ppe.txt", "w")
        for initial_ppe_file_item in initial_ppe_file_object_list:
            temp_ppe_file_object.write(initial_ppe_file_item + "\n")
    initial_ppe_file_object.close()
    temp_ppe_file_object.close()
    print("Stock updated successfully")
```

This function will update the item quantity. Four parameters are taken, *supplier_code*, *item_code*, *new_quantity*, and *action* to perform business logic. It able to deduct or add a new item to the supplier.

check_item_stock_is_enough

```
def check_item_stock_is_enough(supplier_code, item_code, quantity_needed):
    """
    This function will check the stock of the item is enough or not
    :param supplier_code: supplier code
    :param item_code: ttem code
    :param quantity_needed: quantity needed, use to compare with stock in database
    :return:
    """
    ppe_file_object = open("ppe.txt", "r")
    for ppe_details in ppe_file_object:
        ppe_details = ppe_details.rstrip()
        ppe_details = ppe_details.split(",")
        if ppe_details[3] == supplier_code and ppe_details[0] == item_code:
            if int(ppe_details[2]) < int(quantity_needed):
                return False, ppe_details[2]
            else:
                return True, ppe_details[2]
    return False, 0
```

This function can check whether the number of items is enough to distribute. It will return True (enough) or False (not enough) with the remaining number of items as value.

get_input_hospital

```
def get_input_hospital_code(check_is_exist = False):
    """
    Function to ask valid hospital code input
    :param check_is_exist: boolean, check if hospital code exist
    :return: return hospital code
    """
    hospital_code = input("Enter hospital code: ")
    while True:
        if hospital_code == "":
            print("Hospital code cannot be empty")
            hospital_code = input("Enter hospital Code: ")
        else:
            if check_is_exist:
                is_valid_hospital = check_is_code_exist(hospital_code, "HOSPITAL")
                if is_valid_hospital:
                    return hospital_code
                else:
                    print("Hospital not found")
                    hospital_code = input("Enter Hospital Code: ")
            else:
                return hospital_code
```

This function will ensure the user submits a valid hospital code or valid string.

get_number_input

```
def get_number_input(title):
    """
    Function to ask valid number input
    :param title: title
    :return: valid input from user
    """
    number_input = input("Enter {}: ".format(title))
    while True:
        if number_input == "":
            print("{} cannot be empty".format(title))
            number_input = input("Enter {}: ".format(title))
        else:
            try:
                number_input = int(number_input)
                return number_input
            except:
                print("{} must be a number".format(title))
                number_input = input("Enter {}: ".format(title))
```

This function will ensure user inputs a valid number.

distribute_process

```
def distribution_process():
    """
    This function will process the distribution record and update the stock
    """
    distribution_file_object = open("distribution.txt", "a")
    view_all_supplier()
    supplier_code = get_input_supplier_code(True)
    view_all_item_based_on_supplier_id(supplier_code)
    view_all_hospital()
    target_hospital_code = get_input_hospital_code(True)
    item_code = get_input_item_code(True)
    quantity_to_distribute = get_number_input("Quantity to distribute")
    is_stock_enough, balance = check_item_stock_is_enough(supplier_code, item_code, quantity_to_distribute)
    if is_stock_enough:
        update_stock(supplier_code, item_code, quantity_to_distribute, "MINUS")
        distribution_file_object.write(
            "{},{},{},{}".format(supplier_code, target_hospital_code, item_code, quantity_to_distribute))
        distribution_file_object.write("\n")
    else:
        print("Invalid stock, remaining {} only".format(balance))
    distribution_file_object.close()
```

This function will perform the item distribution process. It will get inputs from the user, check the stock balance using check_item_stock_is_enough function, update the stock using update_stock function, and record it in *distribution.txt*.

distribution_module

```
def distribution_module():
    """
    This function will process the distribution module
    """
    distribute_item = True
    while distribute_item:
        distribution_process()
        continue_ask_confirm = True
        while continue_ask_confirm:
            break_process = input("Want to distribute next item? [0-no 1-yes]: ")
            if break_process == "0" or break_process == "1":
                if break_process == "0":
                    continue_ask_confirm = False
                    distribute_item = False
                else:
                    continue_ask_confirm = False
                    distribute_item = True
            else:
                continue_ask_confirm = True
                print("Invalid Input, only accept 0 and 1")
```

This function will ask the user whether he/she wants to distribute items to the hospital by calling *distribute_process* function.

sort_item_list

```
def sort_item_list(item_list):
    """
    This function will sort the item list by item code
    :param item_list:
    :return: sorted item_list
    """
    init_list = len(item_list)
    for i in range(0, init_list):
        for j in range(0, init_list - i - 1):
            if item_list[j][0] > item_list[j + 1][0]:
                tempo = item_list[j]
                item_list[j] = item_list[j + 1]
                item_list[j + 1] = tempo
    return item_list
```

This function will sort the nested list by 1st element in the list by ascending order.

retrieve_all_based_on_supplier

```
def retrieve_all_based_on_supplier(supplier_code, less_than_threshold = False):
    """
    This function will retrieve all the item based on supplier code
    :param supplier_code: supplier code
    :param less_than_threshold: if True only retrieve the item less than 25, else will retrieve all item
    :return:
    """
    ppe_file_object = open("ppe.txt", "r")
    item_to_show = []
    for ppe_details in ppe_file_object:
        ppe_details = ppe_details.rstrip()
        ppe_details = ppe_details.split(",")
        if ppe_details[3] == supplier_code:
            if less_than_threshold:
                if int(ppe_details[2]) < 25:
                    item_to_show.append(ppe_details)
            else:
                item_to_show.append(ppe_details)
    item_to_show = sort_item_list(item_to_show)
    return item_to_show
```

This function will retrieve the details of the items based on supplier_code. If *the less_than_thresold parameter is True*, it will only retrieve the item details with less than 25 in the record.

display_item

```
def display_item(items_list):
    """
    This function will display the item list
    :param items_list: nested item list
    """
    print()
    print("-" * 50)
    print("View Stocks")
    print("-" * 50)
    print("{:<15} {:<15} {:<15}".format("Supplier Code", "Item Code", "Quantity"))
    for item in items_list:
        print("{:<15} {:<15} {:<15}".format(item[0], item[1], item[2]))
    print()
```

This function will loop through the *items_list* parameter and display it in a table view.

view_stock_less_than_twenty_five

```
def view_stock_less_than_twenty_five():
    """
    This function will view the stock less than 25 based on supplier code
    """
    view_all_supplier()
    supplier_code = get_input_supplier_code()
    items_list = retrieve_all_based_on_supplier(supplier_code, True)
    if len(items_list) > 0:
        display_item(items_list)
    else:
        print("No Data")
```

This function will only display the details of items less than 25 items in the record by calling *retrieve_all_based_on_supplier* function and *display_item* function based on *supplier_code*.

view_all_stock

```
def view_all_stock():
    """
    This function will view all the stock
    """
    view_all_supplier()
    supplier_code = get_input_supplier_code()
    items_list = retrieve_all_based_on_supplier(supplier_code)
    if len(items_list) > 0:
        display_item(items_list)
    else:
        print("No Data")
```

This function will display all the details of items in the record by calling *retrieve_all_based_on_supplier* function and *display_item* function based on *supplier_code*.

add_item

```
def add_item(flexible_quantity):
    """
    This function will add the stock to supplier
    :param flexible_quantity: if True, will update the quantity based on the input, else will update the quantity with 100
    """
    supplier_code = get_input_supplier_code(True)
    if flexible_quantity:
        item_code = get_input_item_code(True, False)
        item_quantity = get_number_input("Item Quantity")
        print('Added {} with extra quantity {} to supplier {}'.format(item_code, item_quantity, supplier_code))
        update_stock(supplier_code, item_code, item_quantity, "ADD")
    else:
        item_name = get_string_input("Item Name")
        item_code = get_input_item_code(True, True)
        item_quantity = 100
        print('Added {} with quantity {} to supplier {}'.format(item_code, 100, supplier_code))
        ppe_file_object = open("ppe.txt", "a")
        ppe_file_object.write("{},{},{},{}".format(item_code, item_name, item_quantity, supplier_code))
        ppe_file_object.write("\n")
        ppe_file_object.close()
```

This function will update the number or item based on supplier or initialize a item with 100 quantities in the record.

inventory_creation

```
def inventory_creation(flexible_quantity = False):
    """
    This function will create the assign item to supplier
    :param flexible_quantity: if True, will update the quantity based on the input, else will update the quantity with 100
    """
    if not flexible_quantity:
        file_object = open("ppe.txt", "w")
        file_object.close()
    continue_add_item = True
    while continue_add_item:
        view_all_supplier()
        add_item(flexible_quantity)
        continue_ask_confirm = True
        while continue_ask_confirm:
            break_process = input("Want to add next item? [0-no 1-yes]: ")
            if break_process == "0" or break_process == "1":
                if break_process == "0":
                    continue_ask_confirm = False
                    continue_add_item = False
                else:
                    continue_ask_confirm = False
                    continue_add_item = True
            else:
                continue_ask_confirm = True
                print("Invalid Input, only accept 0 and 1")
```

This function will let the user initialize the item or update the supplier. If the user choose to initialize the items again, all the previous data will be overwritten.

add_supplier

```
def add_supplier(current_supplier):
    """
    Function to add supplier
    :param current_supplier: current supplier
    """
    print("Insert supplier {} details".format(current_supplier))
    supplier_code = get_input_supplier_code()
    while True:
        is_supplier_exist = check_is_code_exist(supplier_code, "SUPPLIER")
        if is_supplier_exist:
            print("Supplier code already exist")
            supplier_code = get_input_supplier_code()
        else:
            break
    supplier_file_object = open("suppliers.txt", "a")
    supplier_name = get_string_input("Supplier name")
    supplier_address = get_string_input("Supplier address")
    supplier_file_object.write("{},{},{}".format(supplier_code, supplier_name, supplier_address))
    supplier_file_object.write("\n")
    supplier_file_object.close()
    print("Supplier {} added successfully\n".format(current_supplier))
```

This function will get the supplier details from the user and record them into *suppliers.txt.*

supplier_registration

```
def supplier_registration():
    """
    Function to register supplier
    """
    print("Supplier registration")
    print("All previous supplier details will be overwritten")
    min_supplier = 3
    max_supplier = 4
    current_supplier = 1

    # Clear Supplier File
    file_object = open("suppliers.txt", "w")
    file_object.close()

    while current_supplier <= max_supplier:
        add_supplier(current_supplier)
        if current_supplier == min_supplier:
            continue_ask_confirm = True
            while continue_ask_confirm:
                break_process = input("Want to add 4th supplier? [0-no 1-yes]: ")
                if break_process == "0" or break_process == "1":
                    if break_process == "0":
                        continue_ask_confirm = False
                        current_supplier = 5
                    else:
                        continue_ask_confirm = False
                else:
                    continue_ask_confirm = True
                    print("Invalid Input, only accept 0 and 1")
        current_supplier += 1
```

This function will make sure the user only initialize 3 or 4 suppliers in the system. Every time this function being called, the supplier data will be overwritten.

add_hospital

```
def add_hospital(current_hospital):
    """
    Function to add hospital
    :param current_hospital: current supplier
    """
    print("Insert hospital {} details".format(current_hospital))
    hospital_code = get_input_hospital_code()
    while True:
        is_hospital_exist = check_is_code_exist(hospital_code, "HOSPITAL")
        if is_hospital_exist:
            print("Hospital code already exist")
            hospital_code = get_input_hospital_code()
        else:
            break
    hospital_file_object = open("hospitals.txt", "a")
    hospital_name = get_string_input("Hospital name")
    hospital_address = get_string_input("Hospital address")
    hospital_file_object.write("{},{},{}".format(hospital_code, hospital_name, hospital_address))
    hospital_file_object.write("\n")
    hospital_file_object.close()
    print("Hospital {} added successfully\n".format(current_hospital))
```

This function will get the hospital details from the user and record them into *hospitals.txt.*

hospital_registration

```
def hospital_registration():
    """
    Function to register hospital
    """
    print("Hospital registration")
    print("All previous hospital details will be overwritten")
    min_hospital = 3
    max_hospital = 4
    current_hospital = 1

    # Clear Hospital File
    file_object = open("hospitals.txt", "w")
    file_object.close()

    while current_hospital <= max_hospital:
        add_hospital(current_hospital)
        if current_hospital == min_hospital:
            continue_ask_confirm = True
            while continue_ask_confirm:
                break_process = input("Want to add 4th hospital? [0-no 1-yes]: ")
                if break_process == "0" or break_process == "1":
                    if break_process == "0":
                        continue_ask_confirm = False
                        current_hospital = 5
                    else:
                        continue_ask_confirm = False
                else:
                    continue_ask_confirm = True
                    print("Invalid Input, only accept 0 and 1")
        current_hospital += 1
```

This function will make sure the user only initialize 3 or 4 hospitals in the system. Every time this function being called, the hospital data will be overwritten.

main_menu

```
def main_menu():
    """
    Function to display main menu
    """
    while True:
        print("-" * 50)
        print("Inventory System")
        print("-" * 50)
        print("1. Supplier Registration")
        print("2. Hospital Registration")
        print("3. Add New Item for Supplier")
        print("4. View All Stock")
        print("5. View Stock which less than 25")
        print("6. Distribute Item to Hospital")
        print("7. Search Item based on Item Code")
        print("8. Supplier Profile Update")
        print("9. Update Supplier Existing Stock")
        print("10. Exit")
        option = input("Enter Option: ")
        try:
            option = int(option)
            if 1 <= option <= 10:
                if option == 1:
                    supplier_registration()
                elif option == 2:
                    hospital_registration()
                elif option == 3:
                    inventory_creation()
                elif option == 4:
                    view_all_stock()
                elif option == 5:
                    view_stock_less_than_twenty_five()
                elif option == 6:
                    distribution_module()
                elif option == 7:
                    retrieve_item_history()
                elif option == 8:
                    update_supplier_details()
                elif option == 9:
                    inventory_creation(True)
                elif option == 10:
                    print("Bye")
                    break
            else:
                print("Invalid Option")
        except:
            print("Option must be a number")


if __name__ == '__main__':
    main_menu()
```

The main_menu function will show all the options available in the system. It will also always make sure the user is inserting a valid input. This function will get called in the main function.

Screenshots of Sample Input/Output and Explanation

Main Menu

```
(venv) C:\Users\hansheng\OneDrive\Desktop\apu-python-labs\assignment>python main.py
------------------------------------------------
Inventory System
------------------------------------------------
1. Supplier Registration
2. Hospital Registration
3. Add New Item for Supplier
4. View All Stock
5. View Stock which less than 25
6. Distribute Item to Hospital
7. Search Item based on Item Code
8. Supplier Profile Update
9. Update Supplier Existing Stock
10. Exit
Enter Option:
```

When the system start, it will show the menu. There are 10 option available:

1. Supplier Registration
2. Hospital Registration
3. Add New Item for Supplier
4. View All Stock based on Supplier
5. View Stock which less than 25 based on Supplier
6. Distribute Item to Hospital
7. Search Item based on Item Code
8. Supplier Profile Update
9. Update Suppliuer Existing Stock
10. Exit

```
--------------------------------------------------
Inventory System
--------------------------------------------------
1. Supplier Registration
2. Hospital Registration
3. Add New Item for Supplier
4. View All Stock
5. View Stock which less than 25
6. Distribute Item to Hospital
7. Search Item based on Item Code
8. Supplier Profile Update
9. Update Supplier Existing Stock
10. Exit
Enter Option: as
Option must be a number
--------------------------------------------------
Inventory System
--------------------------------------------------
1. Supplier Registration
2. Hospital Registration
3. Add New Item for Supplier
4. View All Stock
5. View Stock which less than 25
6. Distribute Item to Hospital
7. Search Item based on Item Code
8. Supplier Profile Update
9. Update Supplier Existing Stock
10. Exit
Enter Option:
Option must be a number
```

System will always validate the option inserted by the user is valid. Empty string and string character is not allowed.

```
--------------------------------------------------
Inventory System
--------------------------------------------------
1. Supplier Registration
2. Hospital Registration
3. Add New Item for Supplier
4. View All Stock
5. View Stock which less than 25
6. Distribute Item to Hospital
7. Search Item based on Item Code
8. Supplier Profile Update
9. Update Supplier Existing Stock
10. Exit
Enter Option: 11
Invalid Option
--------------------------------------------------
Inventory System
--------------------------------------------------
1. Supplier Registration
2. Hospital Registration
3. Add New Item for Supplier
4. View All Stock
5. View Stock which less than 25
6. Distribute Item to Hospital
7. Search Item based on Item Code
8. Supplier Profile Update
9. Update Supplier Existing Stock
10. Exit
Enter Option: 0
Invalid Option
```

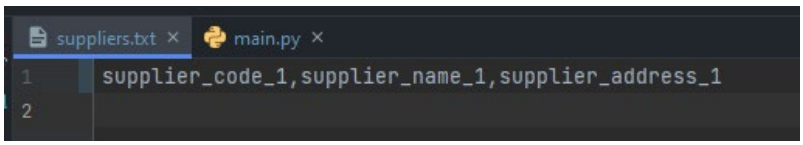
Since the system only provide 10 option in menu, if the user insert any number larger than 10 or smaller than 1, it will raise *Invalid Option* exception.

Supplier Registration

```
--------------------------------------------------
Inventory System
--------------------------------------------------
1. Supplier Registration
2. Hospital Registration
3. Add New Item for Supplier
4. View All Stock
5. View Stock which less than 25
6. Distribute Item to Hospital
7. Search Item based on Item Code
8. Supplier Profile Update
9. Update Supplier Existing Stock
10. Exit
Enter Option: 1
Supplier registration
All previous supplier details will be overwritten
Insert supplier 1 details
Enter Supplier Code:
```

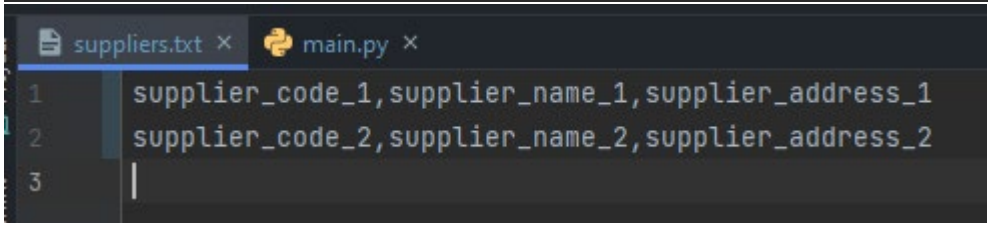
If user choose *Supplier Registration,* a message "*Supplie Registration, All previous supplier details will be overwritten*" will display as this function will overwrite all the previous data.

```
Insert supplier 1 details
Enter Supplier Code:
Supplier code cannot be empty
Enter supplier code: supplier_code_1
Enter Supplier name:
Supplier name cannot be empty
Enter Supplier name: supplier_name_1
Enter Supplier address:
Supplier address cannot be empty
Enter Supplier address: supplier_address_1
Supplier 1 added successfully
```



System will always make sure it receive a valid input from user when creating supplier. If all the values are validated, system will instantly save the supplier details into the record, so that system able to prevent duplicated record.

```
Insert supplier 2 details
Enter Supplier Code: supplier_code_2
Enter Supplier name: supplier_name_2
Enter Supplier address: supplier_address_2
Supplier 2 added successfully
```



Once supplier 2 details received, the details will save into the record.

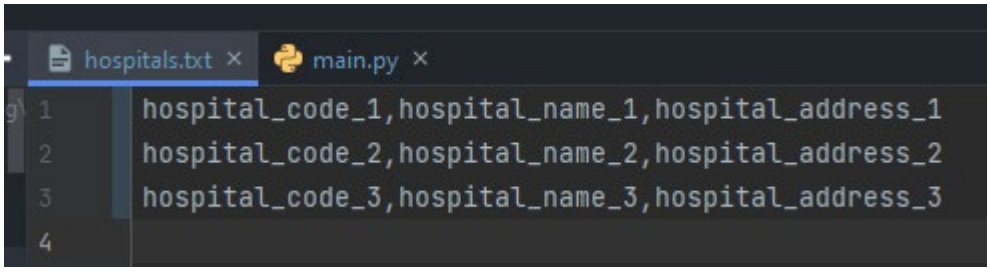
```
Insert supplier 3 details
Enter Supplier Code: supplier_code_1
Supplier code already exist
Enter Supplier Code: supplier_code_2
Supplier code already exist
Enter Supplier Code: supplier_code_3
Enter Supplier name: supplier_name_3
Enter Supplier address: supplier_address_3
Supplier 3 added successfully

Want to add 4th supplier? [0-no 1-yes]:
```

If user receive a duplicated supplier code, it will raise message “Supplier code already exist” and user need to insert again the code. Once user done filled the details for supplier 3, system will confirm whether user want to insert the 4th supplier details.

```
Want to add 4th supplier? [0-no 1-yes]: 3
Invalid Input, only accept 0 and 1
Want to add 4th supplier? [0-no 1-yes]: a
Invalid Input, only accept 0 and 1
Want to add 4th supplier? [0-no 1-yes]: asas
Invalid Input, only accept 0 and 1
Want to add 4th supplier? [0-no 1-yes]:
Invalid Input, only accept 0 and 1
Want to add 4th supplier? [0-no 1-yes]: 1
Insert supplier 4 details
Enter Supplier Code: supplier_code_4
Enter Supplier name: supplier_name_4
Enter Supplier address: supplier_address_4
Supplier 4 added successfully
```

Validation module implemented to validate the input. If the user choose to continue, system will prompt the input for 4th supplier, else it will show main menu.

```
Hospital registration
All previous hospital details will be overwritten
Insert hospital 1 details
Enter hospital code:
Hospital code cannot be empty
Enter hospital Code: hospital_code_1
Enter Hospital name: hospital_name_1
Enter Hospital address: hospital_address_1
Hospital 1 added successfully

Insert hospital 2 details
Enter hospital code: hospital_code_2
Enter Hospital name: hospital_name_2
Enter Hospital address: hospital_address_2
Hospital 2 added successfully

Insert hospital 3 details
Enter hospital code: hospital_code_1
Hospital code already exist
Enter hospital code: hospital_code_3
Enter Hospital name: hospital_name_3
Enter Hospital address: hospital_address_3
Hospital 3 added successfully

Want to add 4th hospital? [0-no 1-yes]: 0
-------------------------------------------------
Inventory System
-------------------------------------------------
1. Supplier Registration
2. Hospital Registration
3. Add New Item for Supplier
4. View All Stock
5. View Stock which less than 25
6. Distribute Item to Hospital
7. Search Item based on Item Code
8. Supplier Profile Update
9. Update Supplier Existing Stock
10. Exit
Enter Option:
```

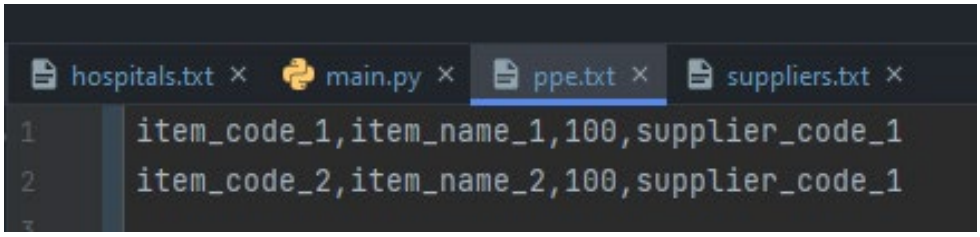


Same logics in Supplier Registration are implemented in Hospital Registration module.

Add New Item for Supplier

```
------------------------------------------------
Inventory System
------------------------------------------------
1. Supplier Registration
2. Hospital Registration
3. Add New Item for Supplier
4. View All Stock
5. View Stock which less than 25
6. Distribute Item to Hospital
7. Search Item based on Item Code
8. Supplier Profile Update
9. Update Supplier Existing Stock
10. Exit
Enter Option: 3
View all suppliers
------------------------------------------------
Suppliers List
------------------------------------------------
Supplier Code   Supplier Name   Supplier Address
supplier_code_1 supplier_name_1 supplier_address_1
supplier_code_2 supplier_name_2 supplier_address_2
supplier_code_3 supplier_name_3 supplier_address_3
supplier_code_4 supplier_name_4 supplier_address_4
------------------------------------------------
Enter Supplier Code: supplier_code_1
Enter item code: item_code_1
Enter Item Name: item_name_1
Added item_code_1 with quantity 100 to supplier supplier_code_1
Want to add next item? [0-no 1-yes]: 1
View all suppliers
------------------------------------------------
Suppliers List
------------------------------------------------
Supplier Code   Supplier Name   Supplier Address
supplier_code_1 supplier_name_1 supplier_address_1
supplier_code_2 supplier_name_2 supplier_address_2
supplier_code_3 supplier_name_3 supplier_address_3
supplier_code_4 supplier_name_4 supplier_address_4
------------------------------------------------
Enter Supplier Code: supplier_code_2
Enter item code: item_code_2
Enter Item Name: item_name_2
Added item_code_2 with quantity 100 to supplier supplier_code_2
Want to add next item? [0-no 1-yes]: as
Invalid Input, only accept 0 and 1
Want to add next item? [0-no 1-yes]:
Invalid Input, only accept 0 and 1
Want to add next item? [0-no 1-yes]: 0
------------------------------------------------
Inventory System
------------------------------------------------
1. Supplier Registration
2. Hospital Registration
3. Add New Item for Supplier
4. View All Stock
```

For the “Add New Item for Supplier”, it will assign item to supplier. All the data will be overwritten to prevent any conflicnt. The red boxes are the part user insert item details and assign to supplier, while blue box is the error handing.



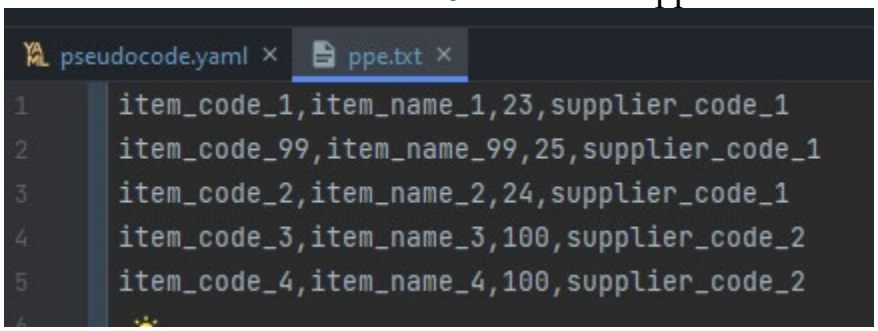
All the details are recorded in *ppe.txt*.

View All Stock based on Supplier

```
------------------------------------------------
Inventory System
------------------------------------------------
1. Supplier Registration
2. Hospital Registration
3. Add New Item for Supplier
4. View All Stock
5. View Stock which less than 25
6. Distribute Item to Hospital
7. Search Item based on Item Code
8. Supplier Profile Update
9. Update Supplier Existing Stock
10. Exit
Enter Option: 4
View all suppliers
------------------------------------------------
Suppliers List
------------------------------------------------
Supplier Code   Supplier Name   Supplier Address
supplier_code_1 supplier_name_1 supplier_address_1
supplier_code_2 supplier_name_2 supplier_address_2
supplier_code_3 supplier_name_3 supplier_address_3
supplier_code_4 supplier_name_4 supplier_address_4
------------------------------------------------
Enter Supplier Code: supplier_code_1

------------------------------------------------
View Stocks
------------------------------------------------
Item Code       Item Name       Quantity
item_code_1     item_name_1     23
item_code_2     item_name_2     24
item_code_99    item_name_99    25
```

This menu will retrieve the stocks based on *supplier_code*. System will show all the suppliers list available so that user able to insert accordingly.

View Stock which less than 25 based on Supplier



The items details are recorded as above.

```
--------------------------------------------------
Inventory System
--------------------------------------------------
1. Supplier Registration
2. Hospital Registration
3. Add New Item for Supplier
4. View All Stock
5. View Stock which less than 25
6. Distribute Item to Hospital
7. Search Item based on Item Code
8. Supplier Profile Update
9. Update Supplier Existing Stock
10. Exit
Enter Option: 5
View all suppliers
--------------------------------------------------
Suppliers List
--------------------------------------------------
Supplier Code   Supplier Name   Supplier Address
supplier_code_1 supplier_name_1 supplier_address_1
supplier_code_2 supplier_name_2 supplier_address_2
supplier_code_3 supplier_name_3 supplier_address_3
supplier_code_4 supplier_name_4 supplier_address_4
--------------------------------------------------
Enter Supplier Code: supplier_code_1

--------------------------------------------------
View Stocks
--------------------------------------------------
Item Code       Item Name       Quantity
item_code_1     item_name_1     23
item_code_2     item_name_2     24
```

In this menu, it will only display the item which less than 25 in stock based on supplier.

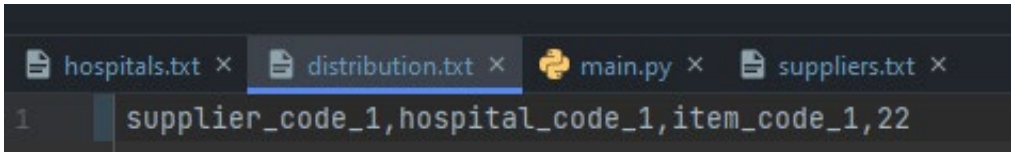
Distribute Item to Hospital

```
------------------------------------------------
Inventory System
------------------------------------------------
1. Supplier Registration
2. Hospital Registration
3. Add New Item for Supplier
4. View All Stock
5. View Stock which less than 25
6. Distribute Item to Hospital
7. Search Item based on Item Code
8. Supplier Profile Update
9. Update Supplier Existing Stock
10. Exit
Enter Option: 6
View all suppliers
------------------------------------------------
Suppliers List
------------------------------------------------
Supplier Code   Supplier Name   Supplier Address
supplier_code_1 supplier_name_1 supplier_address_1
supplier_code_2 supplier_name_2 supplier_address_2
supplier_code_3 supplier_name_3 supplier_address_3
supplier_code_4 supplier_name_4 supplier_address_4
------------------------------------------------
Enter Supplier Code: supplier_code_1
View all hospital

------------------------------------------------
Hospital List
------------------------------------------------
Hospital Code   Hospital Name   Hospital Address
hospital_code_1 hospital_name_1 hospital_address_1
hospital_code_2 hospital_name_2 hospital_address_2
hospital_code_3 hospital_name_3 hospital_address_3
------------------------------------------------
Enter hospital code: hospital_code_1

------------------------------------------------
View all item based on supplier id
------------------------------------------------
Item Code       Item Name       Item Quantity
item_code_1     item_name_1     23
item_code_99    item_name_99    25
item_code_2     item_name_2     24
Enter item code: item_code_1
Enter Quantity to distribute: 99
Invalid stock, remaining 23 only
Want to distribute next item? [0-no 1-yes]:
```

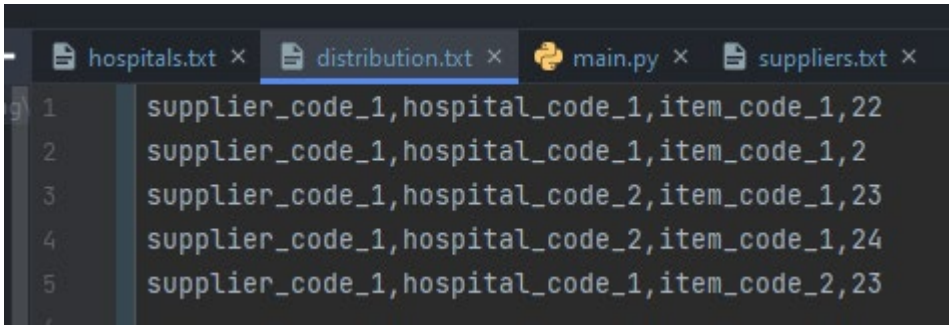
In this distribute menu, user will need to insert supplier code, target hospital code, item code and quantity want to distribute. The system will check the remaining stock first before it distribute.

```
View all suppliers
-------------------------------------------------
Suppliers List
-------------------------------------------------
Supplier Code   Supplier Name   Supplier Address
supplier_code_1 supplier_name_1 supplier_address_1
supplier_code_2 supplier_name_2 supplier_address_2
supplier_code_3 supplier_name_3 supplier_address_3
supplier_code_4 supplier_name_4 supplier_address_4
-------------------------------------------------
Enter Supplier Code: supplier_code_1
View all hospital

-------------------------------------------------
Hospital List
-------------------------------------------------
Hospital Code   Hospital Name   Hospital Address
hospital_code_1 hospital_name_1 hospital_address_1
hospital_code_2 hospital_name_2 hospital_address_2
hospital_code_3 hospital_name_3 hospital_address_3
-------------------------------------------------
Enter hospital code: hospital_code_1

-------------------------------------------------
View all item based on supplier id
-------------------------------------------------
Item Code       Item Name       Item Quantity
item_code_1     item_name_1     23
item_code_99    item_name_99    25
item_code_2     item_name_2     24
Enter item code: item_code_1
Enter Quantity to distribute: 22
Stock updated successfully
Want to distribute next item? [0-no 1-yes]:
```

If the distribution process is success, it will record in the *distribution.txt*.



Search Item based on Item Code

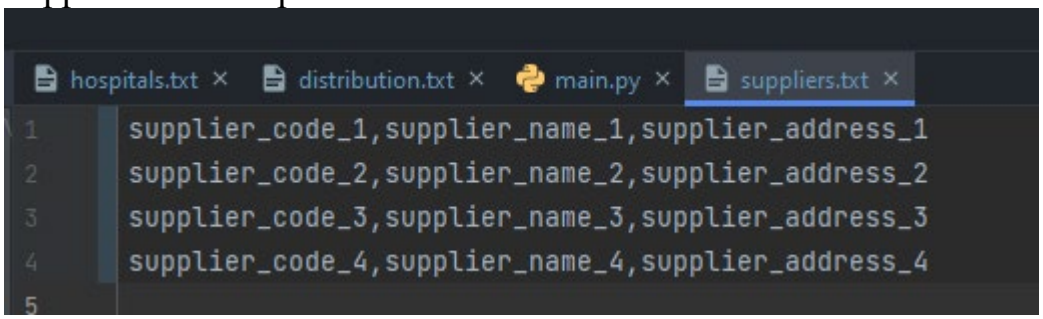


Here's the distribution recorded in txt file.

```
--------------------------------------------------
Inventory System
--------------------------------------------------
1. Supplier Registration
2. Hospital Registration
3. Add New Item for Supplier
4. View All Stock
5. View Stock which less than 25
6. Distribute Item to Hospital
7. Search Item based on Item Code
8. Supplier Profile Update
9. Update Supplier Existing Stock
10. Exit
Enter Option: 7
Enter item code: item_code_1
Supplier Code   Target Hospital Quantity        Item Code
supplier_code_1 hospital_code_1 24              item_code_1
supplier_code_1 hospital_code_2 47              item_code_1
```

If user insert *item_code_1*, it will combine the quantity distributed grouped by hospital.

Supplier Profile Update

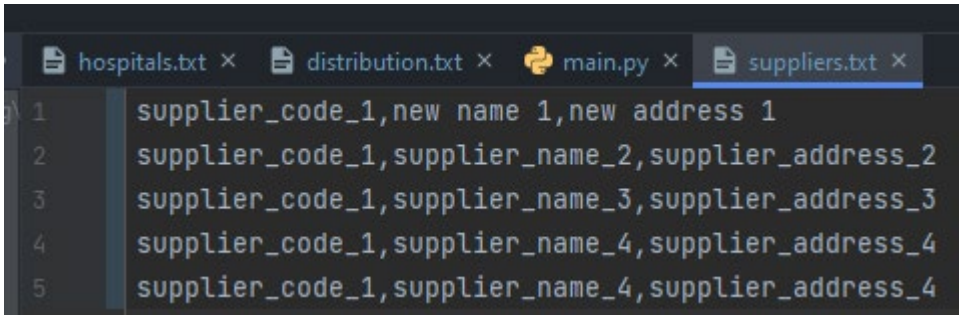


The initial supplier details are recorded as above.

```
------------------------------------------------
Inventory System
------------------------------------------------
1. Supplier Registration
2. Hospital Registration
3. Add New Item for Supplier
4. View All Stock
5. View Stock which less than 25
------------------------------------------------
1. Supplier Registration
2. Hospital Registration
3. Add New Item for Supplier
4. View All Stock
5. View Stock which less than 25
6. Distribute Item to Hospital
7. Search Item based on Item Code
8. Supplier Profile Update
9. Update Supplier Existing Stock
10. Exit
Enter Option: 8
View all suppliers
------------------------------------------------
Suppliers List
------------------------------------------------
Supplier Code   Supplier Name   Supplier Address
supplier_code_1 supplier_name_1 supplier_address_1
supplier_code_2 supplier_name_2 supplier_address_2
supplier_code_3 supplier_name_3 supplier_address_3
supplier_code_4 supplier_name_4 supplier_address_4
------------------------------------------------
Enter Supplier Code: supplier_code_1
Enter New Supplier Name: new name 1
Enter New Supplier Address: new address 1
Profile Updated Successfully
```

After the user select this menu, system will ask for new supplier name and address, and update based on supplier code.

New supplier details are recorded in *suppliers.txt*.



Update Supplier Existing Stock

```
Enter Supplier Code: supplier_code_1

------------------------------------------------
View all item based on supplier id
------------------------------------------------
Item Code       Item Name       Item Quantity
item_code_1     item_name_1     1
item_code_99    item_name_99    25
item_code_2     item_name_2     24
item_code_3     item_name_3     100
item_code_4     item_name_4     100
Enter item code: item_code_1
Enter Item Quantity: 100
Added item_code_1 with extra quantity 100 to supplier supplier_code_1
Stock updated successfully
Want to add next item? [0-no 1-yes]: 0
------------------------------------------------
Inventory System
------------------------------------------------
1. Supplier Registration
2. Hospital Registration
3. Add New Item for Supplier
4. View All Stock
5. View Stock which less than 25
6. Distribute Item to Hospital
7. Search Item based on Item Code
8. Supplier Profile Update
9. Update Supplier Existing Stock
10. Exit
Enter Option: 4
View all suppliers
------------------------------------------------
Suppliers List
------------------------------------------------
Supplier Code   Supplier Name   Supplier Address
supplier_code_1 new name 1      new address 1
supplier_code_1 supplier_name_2 supplier_address_2
supplier_code_1 supplier_name_3 supplier_address_3
supplier_code_1 supplier_name_4 supplier_address_4
supplier_code_1 supplier_name_4 supplier_address_4
------------------------------------------------
Enter Supplier Code: supplier_code_1

------------------------------------------------
View Stocks
------------------------------------------------
Item Code       Item Name       Quantity
item_code_1     item_name_1     101
item_code_2     item_name_2     24
item_code_3     item_name_3     100
item_code_4     item_name_4     100
item_code_99    item_name_99    25
```

This menu will ask for supplier code, item code and quantity wish to add. The system will update the data accordingly.

Conclusion

In a nutshell, this program able to function perfectly and help user to solve the problems. From system design (pseudocode and flowchart) until implementation, everything work as expected and good fit to the requirement.