**1: MAIN FUNCTION**

BEGIN main function

int choice

WHILE 1, always stay inside the loop until forcefully terminated THAN

PRINT \*\*\*\*\*\*\*\*\*\*\*\*\*\* AUTOMOBILE PARTS INVENTORY MANAGEMENT SYSTEM \*\*\*\*\*\*\*\*\*\*\*\*\*

PRINT 1. Parts Inventory Creation in Warehouses

PRINT 2. Parts Inventory Update

PRINT 3. Parts Inventory Tracking

PRINT 4. Searching Functionalities

PRINT 5. Exit

PRINT Enter your choice:

choice = INPUT from user

SWITCH choice, select on the basis of the Input value

CASE 1, input is 1

PRINT \*\*\*\*\*\*\*\*\*\*\*\*\*\* CREATION OF PARTS INVENTORY IN WAREHOUSES \*\*\*\*\*\*\*\*\*\*\*\*\*\*

call PartsInventoryCreation Function

BREAK

CASE 2, input is 2

PRINT \*\*\*\*\*\*\*\*\*\*\*\*\*\* UPDATE OF PARTS INVENTORY IN WAREHOUSES \*\*\*\*\*\*\*\*\*\*\*\*\*\*

call PartsInventoryUpdate Function

BREAK

CASE 3, input is 3

PRINT \*\*\*\*\*\*\*\*\*\*\*\*\*\* TRACKING OF PARTS INVENTORY IN WAREHOUSES \*\*\*\*\*\*\*\*\*\*\*\*\*\*

call PartsInventoryTracking Function

BREAK

CASE 4, input is 4

PRINT \*\*\*\*\*\*\*\*\*\*\*\*\*\* SEARCHING FUNCTIONALITIES \*\*\*\*\*\*\*\*\*\*\*\*\*\*

call PartsInventorySearch Function

BREAK

CASE5, input is 5

PRINT Have A Nice Day!

EXIT, get out of WHILE loop

DEFAULT

PRINT \*\*\*\*\*\*\*Enter Correct Input\*\*\*\*\*\*\*

END SWITCH

PRINT Have A Nice Day!

END main function

**2: PartsInventoryCreation Function**

BEGIN PartsInventoryCreation Function

FOR i=0 to 0

IF i = 0 THEN

Copy String “EngineBlock” to warehouse1’s (instance of warehouse struct) parts\_name array’s zero index

Copy String “ExhaustValve” to parts\_name array’s first index

Copy String “RoofPanel” to parts\_name array’s second index

Copy String “Bumper” to parts\_name array’s third index

Copy String “ACCondenser” to parts\_name array’s fourth index

FOR i = 0 to 4

Assign 50 to parts\_quantity array, which is a part of warehouse1 (an instance of struct warehouse)

END FOR

FOR i = 0 to 4

Assign 101 + i to parts\_id array, which is a part of warehouse1 (an instance of struct warehouse)

END FOR

END IF

END FOR

FOR i = 0 to 4

PRINT “Parts Name :”, these are all the names of the parts which are stored in warehouse1’s parts\_name array

PRINT “Parts ID : ”, these are the ID numbers of all the parts which are stored in warehouse1’s parts\_id array

PRINT “Parts Quantity:”, these are the quantity of all the parts which are stored in warehouse’1 parts\_quantity array

END FOR

OPEN file which would be used to store information of WBZ warehouse

IF fp (wbz.txt file) = NULL (file doesn’t exist)

PRINT “Error in opening the file”

EXIT(1) terminate the program

END IF

FOR i = 0 to 4

Put contents of warehouse1’s parts\_name[i] array, parts\_id[i] array and parts\_quantity[i] array into wbz.txt file

END FOR

CLOSE the file

PRINT Parts Inventory Created Warehouse WBZ Successfully !

FOR i=0 to 0

IF i = 0 THEN

Copy String “WaterPump” to warehouse2’s (instance of warehouse struct) parts\_name array’s zero index

Copy String “OilFilter” to parts\_name array’s first index

Copy String “Door” to parts\_name array’s second index

Copy String “Hood” to parts\_name array’s third index

Copy String “Compressor” to parts\_name array’s fourth index

FOR i = 0 to 4

Assign 50 to parts\_quantity array, which is a part of warehouse2 (an instance of struct warehouse)

END FOR

FOR i = 0 to 4

Assign 106 + i to parts\_id array, which is a part of warehouse2 (an instance of struct warehouse)

END FOR

END IF

END FOR

FOR i = 0 to 4

PRINT “Parts Name :”, these are all the names of the parts which are stored in warehouse2’s parts\_name array

PRINT “Parts ID : ”, these are the ID numbers of all the parts which are stored in warehouse2’s parts\_id array

PRINT “Parts Quantity:”, these are the quantity of all the parts which are stored in warehouse2’s parts\_quantity array

END FOR

OPEN file which would be used to store information of WSL warehouse

IF fp1 (wsl.txt file) = NULL (file doesn’t exist)

PRINT “Error in opening the file”

EXIT(1) terminate the program

END IF

FOR i = 0 to 4

Put contents of warehouse2’s parts\_name[i] array, parts\_id[i] array and parts\_quantity[i] array into wsl.txt file

END FOR

CLOSE the file

PRINT Parts Inventory Created Warehouse WSL Successfully !

FOR i=0 to 0

IF i = 0 THEN

Copy String “SteeringWheel” to warehouse3’s (instance of warehouse struct) parts\_name array’s zero index

Copy String “Tire” to parts\_name array’s first index

Copy String “Brake” to parts\_name array’s second index

Copy String “BrakePad” to parts\_name array’s third index

Copy String “BrakeDisc” to parts\_name array’s fourth index

FOR i = 0 to 4

Assign 50 to parts\_quantity array, which is a part of warehouse3 (an instance of struct warehouse)

END FOR

FOR i = 0 to 4

Assign 111 + i to parts\_id array, which is a part of warehouse3 (an instance of struct warehouse)

END FOR

END IF

END FOR

FOR i = 0 to 4

PRINT “Parts Name :”, these are all the names of the parts which are stored in warehouse3’s parts\_name[i] array

PRINT “Parts ID : ”, these are the ID numbers of all the parts which are stored in warehouse3’s parts\_id[i] array

PRINT “Parts Quantity:”, these are the quantity of all the parts which are stored in warehouse3’s parts\_quantity[i] array

END FOR

OPEN file which would be used to store information of WAR warehouse

IF fp2 (war.txt file) = NULL (file doesn’t exist)

PRINT “Error in opening the file”

EXIT(1) terminate the program

END IF

FOR i = 0 to 4

Put contents of warehouse3’s parts\_name[i] array, parts\_id[i] array and parts\_quantity[i] array into war.txt file

END FOR

CLOSE the file

PRINT Parts Inventory Created Warehouse WAR Successfully !

END PartsInventoryCreation Function

**3:PartsInventoryUpdate Function**

BEGIN PartsInventoryUpdate Function

INT choice

PRINT \*\*\*\*\*\*\*Select whether your a supplier or distributor\*\*\*\*\*\*\*

PRINT 1. Supplier

PRINT 2. Distributor

PRINT Enter your choice :

choice = INPUT from user

SWITCH choice

CASE 1, choice variable contains 1

PRINT You are a Supplier

PRINT You can update the parts inventory

PRINT 1. WBZ Warehouse

PRINT 2. WSL Warehouse

PRINT 3. WAR Warehouse

PRINT Enter your choice :

int choice1

choice1 = INPUT from user

IF choice1 = 1 THEN

OPEN file containing information of WBZ warehouse

int count = 0

char c

IF fp2 (wbz.txt file) = NULL (doesn’t exist)

PRINT Could not open file wbz.txt

END IF

FOR c = characters from the file (wbz.txt) TO EOF (end of file)

IF c = \n (newline)

count = count +1

END IF

END FOR

CLOSE fp2 file

FOR i = 0 to count

Put contents of f(wbz.txt) file into readfromfile’s (another instance of warehouse struct) parts\_name[i] array, parts\_id[i] array, and parts\_quantity[i] array

END FOR

CLOSE f file

PRINT You are supplying part to the WBZ Warehouse

PRINT You are WBZ Supplier

PRINT You can only supply parts of Warehouse 1

PRINT Current Parts and their Quantity in WBZ Warehouse:

FOR i = 0 to 4

PRINT “Parts Name :”, these are all the names of the parts which are stored in readfromfile’s parts\_name array

PRINT “Parts ID : ”, these are the ID numbers of all the parts which are stored in readfromfile’s parts\_id array

PRINT “Parts Quantity:”, these are the quantity of all the parts which are stored in readfromfile’s parts\_quantity array

END FOR

PRINT Enter part id you want to supply more :

int choice2

choice2 = INPUT from user

IF choice2 = 101 THEN

PRINT Enter the part quantity (+) :

int part\_quantity

part\_quantity = INPUT from user

readfromfile’s parts\_quantity[0] (zero index) = readfromfile’s parts\_quantity[0] + part\_quantity

PRINT “Updated Part quantity of ”, name inside readfromfile’s parts\_name[0] “is ”, value inside readfromfile’s parts\_quantity[0]

OPEN suppliers.txt file

Put contents of readfromfile’s parts\_id[0] and part\_quantity into suppliers.txt file next to “WBZ\_Supplier” text

CLOSE fp3 (suppliers.txt) file

PRINT Parts Inventory Updated Successfully !

END IF

ELSE IF choice2 = 102 THEN

PRINT Enter the part quantity (+) :

int part\_quantity

part\_quantity = INPUT from user

readfromfile’s parts\_quantity[1] (index one) = readfromfile’s parts\_quantity[1] + part\_quantity

PRINT “Updated Part quantity of ”, name inside readfromfile’s parts\_name[1] “is ”, value inside readfromfile’s parts\_quantity[1]

OPEN suppliers.txt file

Put contents of readfromfile’s parts\_id[1] and part\_quantity into suppliers.txt file next to “WBZ\_Supplier” text

CLOSE fp3 (suppliers.txt) file

PRINT Parts Inventory Updated Successfully !

END IF

ELSE IF choice2 = 103 THEN

PRINT Enter the part quantity (+) :

int part\_quantity

part\_quantity = INPUT from user

readfromfile’s parts\_quantity[2] (index two) = readfromfile’s parts\_quantity[2] + part\_quantity

PRINT “Updated Part quantity of ”, name inside readfromfile’s parts\_name[2] “is ”, value inside readfromfile’s parts\_quantity[2]

OPEN suppliers.txt file

Put contents of readfromfile’s parts\_id[2] and part\_quantity into suppliers.txt file next to “WBZ\_Supplier” text

CLOSE fp3 (suppliers.txt) file

PRINT Parts Inventory Updated Successfully !

END IF

ELSE IF choice2 = 104 THEN

PRINT Enter the part quantity (+) :

int part\_quantity

part\_quantity = INPUT from user

readfromfile’s parts\_quantity[3] (index three) = readfromfile’s parts\_quantity[3] + part\_quantity

PRINT “Updated Part quantity of ”, name inside readfromfile’s parts\_name[3] “is ”, value inside readfromfile’s parts\_quantity[3]

OPEN suppliers.txt file

Put contents of readfromfile’s parts\_id[3] and part\_quantity into suppliers.txt file next to “WBZ\_Supplier” text

CLOSE fp3 (suppliers.txt) file

PRINT Parts Inventory Updated Successfully !

END IF

ELSE IF choice2 = 105 THEN

PRINT Enter the part quantity (+) :

int part\_quantity

part\_quantity = INPUT from user

readfromfile’s parts\_quantity[4] (index fourth) = readfromfile’s parts\_quantity[4] + part\_quantity

PRINT “Updated Part quantity of ”, name inside readfromfile’s parts\_name[4] “is ”, value inside readfromfile’s parts\_quantity[4]

OPEN suppliers.txt file

Put contents of readfromfile’s parts\_id[4] and part\_quantity into suppliers.txt file next to “WBZ\_Supplier” text

CLOSE fp3 (suppliers.txt) file

PRINT Parts Inventory Updated Successfully !

END IF

ELSE

PRINT Invalid Choice !

END ELSE

OPEN wbz.txt file

FOR i = 0 to 4

Put contents of readfromfile’s parts\_name[i] array, parts\_id[i] array and parts\_quantity[i] array into wbz.txt file

END FOR

CLOSE fp1(wbz.txt) file

END IF

ELSE IF choice1 = 2 THEN

OPEN file containing information of WSL warehouse

int count = 0

char c

IF fp2 (wsl.txt file) = NULL (doesn’t exist)

PRINT Could not open file wsl.txt

END IF

FOR c = characters from the file (wbz.txt) TO EOF (end of file)

IF c = \n (newline)

count = count +1

END IF

END FOR

CLOSE fp2 file

FOR i = 0 to count

Put contents of f(wbz.txt) file into readfromfile’s (another instance of warehouse struct) parts\_name[i] array, parts\_id[i] array, and parts\_quantity[i] array

END FOR

CLOSE f(wsl.txt) file

PRINT You are supplying part to the WSL Warehouse

PRINT You are WSL Supplier

PRINT You can only supply parts of Warehouse 2

PRINT Current Parts and their Quantity in WSL Warehouse:

FOR i = 0 to 4

PRINT “Parts Name :”, these are all the names of the parts which are stored in readfromfile’s parts\_name array

PRINT “Parts ID : ”, these are the ID numbers of all the parts which are stored in readfromfile’s parts\_id array

PRINT “Parts Quantity:”, these are the quantity of all the parts which are stored in readfromfile’s parts\_quantity array

END FOR

PRINT Enter part id you want to supply more :

int choice2

choice2 = INPUT from user

IF choice2 = 106 THEN

PRINT Enter the part quantity (+) :

int part\_quantity

part\_quantity = INPUT from user

readfromfile’s parts\_quantity[0] (zero index) = readfromfile’s parts\_quantity[0] + part\_quantity

PRINT “Updated Part quantity of ”, name inside readfromfile’s parts\_name[0] “is ”, value inside readfromfile’s parts\_quantity[0]

OPEN suppliers.txt file

Put contents of readfromfile’s parts\_id[0] and part\_quantity into suppliers.txt file next to “WSL\_Supplier” text

CLOSE fp3 (suppliers.txt) file

PRINT Parts Inventory Updated Successfully !

END IF

ELSE IF choice2 = 107 THEN

PRINT Enter the part quantity (+) :

int part\_quantity

part\_quantity = INPUT from user

readfromfile’s parts\_quantity[1] (index one) = readfromfile’s parts\_quantity[1] + part\_quantity

PRINT “Updated Part quantity of ”, name inside readfromfile’s parts\_name[1], “is ”, value inside readfromfile’s parts\_quantity[1]

OPEN suppliers.txt file

Put contents of readfromfile’s parts\_id[1] and part\_quantity into suppliers.txt file next to “WSL\_Supplier” text

CLOSE fp3 (suppliers.txt) file

PRINT Parts Inventory Updated Successfully !

END IF

ELSE IF choice2 = 108 THEN

PRINT Enter the part quantity (+) :

int part\_quantity

part\_quantity = INPUT from user

readfromfile’s parts\_quantity[2] (index two) = readfromfile’s parts\_quantity[2] + part\_quantity

PRINT “Updated Part quantity of ”, name inside readfromfile’s parts\_name[2] “is ”, value inside readfromfile’s parts\_quantity[2]

OPEN suppliers.txt file

Put contents of readfromfile’s parts\_id[2] and part\_quantity into suppliers.txt file next to “WSL\_Supplier” text

CLOSE fp3 (suppliers.txt) file

PRINT Parts Inventory Updated Successfully !

END IF

ELSE IF choice2 = 109 THEN

PRINT Enter the part quantity (+) :

int part\_quantity

part\_quantity = INPUT from user

readfromfile’s parts\_quantity[3] (index three) = readfromfile’s parts\_quantity[3] + part\_quantity

PRINT “Updated Part quantity of ”, name inside readfromfile’s parts\_name[3] “is ”, value inside readfromfile’s parts\_quantity[3]

OPEN suppliers.txt file

Put contents of readfromfile’s parts\_id[3] and part\_quantity into suppliers.txt file next to “WSL\_Supplier” text

CLOSE fp3 (suppliers.txt) file

PRINT Parts Inventory Updated Successfully !

END IF

ELSE IF choice2 = 110 THEN

PRINT Enter the part quantity (+) :

int part\_quantity

part\_quantity = INPUT from user

readfromfile’s parts\_quantity[4] (index fourth) = readfromfile’s parts\_quantity[4] + part\_quantity

PRINT “Updated Part quantity of ”, name inside readfromfile’s parts\_name[4] “is ”, value inside readfromfile’s parts\_quantity[4]

OPEN suppliers.txt file

Put contents of readfromfile’s parts\_id[4] and part\_quantity into suppliers.txt file next to “WSL\_Supplier” text

CLOSE fp3 (suppliers.txt) file

PRINT Parts Inventory Updated Successfully !

END IF

ELSE

PRINT Invalid Part ID !

END ELSE

OPEN wsl.txt file

FOR i = 0 to 4

Put contents of readfromfile’s parts\_name[i] array, parts\_id[i] array and parts\_quantity[i] array into wsl.txt file

END FOR

CLOSE fp1(wsl.txt) file

END IF

ELSE IF choice1 = 3 THEN

OPEN file containing information of WAR warehouse

int count = 0

char c

IF fp2 (war.txt file) = NULL (doesn’t exist)

PRINT Could not open file war.txt

END IF

FOR c = characters from the file (war.txt) TO EOF (end of file)

IF c = \n (newline)

count = count +1

END IF

END FOR

CLOSE fp2 file

FOR i = 0 to count

Put contents of f(war.txt) file into readfromfile’s (another instance of warehouse struct) parts\_name[i] array, parts\_id[i] array, and parts\_quantity[i] array

END FOR

CLOSE f(war.txt) file

PRINT You are supplying part to the WAR Warehouse

PRINT You are WAR Supplier

PRINT You can only supply parts of Warehouse 3

PRINT Current Parts and their Quantity in WAR Warehouse:

FOR i = 0 to 4

PRINT “Parts Name :”, these are all the names of the parts which are stored in readfromfile’s parts\_name array

PRINT “Parts ID : ”, these are the ID numbers of all the parts which are stored in readfromfile’s parts\_id array

PRINT “Parts Quantity:”, these are the quantity of all the parts which are stored in readfromfile’s parts\_quantity array

END FOR

PRINT Enter part id you want to supply more :

int choice2

choice2 = INPUT from user

IF choice2 = 111 THEN

PRINT Enter the part quantity (+) :

int part\_quantity

part\_quantity = INPUT from user

readfromfile’s parts\_quantity[0] (zero index) = readfromfile’s parts\_quantity[0] + part\_quantity

PRINT “Updated Part quantity of ”, name inside readfromfile’s parts\_name[0] “is ”, value inside readfromfile’s parts\_quantity[0]

OPEN suppliers.txt file

Put contents of readfromfile’s parts\_id[0] and part\_quantity into suppliers.txt file next to “WAR\_Supplier” text

CLOSE fp3 (suppliers.txt) file

PRINT Parts Inventory Updated Successfully !

END IF

ELSE IF choice2 = 112 THEN

PRINT Enter the part quantity (+) :

int part\_quantity

part\_quantity = INPUT from user

readfromfile’s parts\_quantity[1] (index one) = readfromfile’s parts\_quantity[1] + part\_quantity

PRINT “Updated Part quantity of ”, name inside readfromfile’s parts\_name[1], “is ”, value inside readfromfile’s parts\_quantity[1]

OPEN suppliers.txt file

Put contents of readfromfile’s parts\_id[1] and part\_quantity into suppliers.txt file next to “WAR\_Supplier” text

CLOSE fp3 (suppliers.txt) file

PRINT Parts Inventory Updated Successfully !

END IF

ELSE IF choice2 = 113 THEN

PRINT Enter the part quantity (+) :

int part\_quantity

part\_quantity = INPUT from user

readfromfile’s parts\_quantity[2] (index two) = readfromfile’s parts\_quantity[2] + part\_quantity

PRINT “Updated Part quantity of ”, name inside readfromfile’s parts\_name[2] “is ”, value inside readfromfile’s parts\_quantity[2]

OPEN suppliers.txt file

Put contents of readfromfile’s parts\_id[2] and part\_quantity into suppliers.txt file next to “WAR\_Supplier” text

CLOSE fp3 (suppliers.txt) file

PRINT Parts Inventory Updated Successfully !

END IF

ELSE IF choice2 = 114 THEN

PRINT Enter the part quantity (+) :

int part\_quantity

part\_quantity = INPUT from user

readfromfile’s parts\_quantity[3] (index three) = readfromfile’s parts\_quantity[3] + part\_quantity

PRINT “Updated Part quantity of ”, name inside readfromfile’s parts\_name[3] “is ”, value inside readfromfile’s parts\_quantity[3]

OPEN suppliers.txt file

Put contents of readfromfile’s parts\_id[3] and part\_quantity into suppliers.txt file next to “WAR\_Supplier” text

CLOSE fp3 (suppliers.txt) file

PRINT Parts Inventory Updated Successfully !

END IF

ELSE IF choice2 = 115 THEN

PRINT Enter the part quantity (+) :

int part\_quantity

part\_quantity = INPUT from user

readfromfile’s parts\_quantity[4] (index fourth) = readfromfile’s parts\_quantity[4] + part\_quantity

PRINT “Updated Part quantity of ”, name inside readfromfile’s parts\_name[4] “is ”, value inside readfromfile’s parts\_quantity[4]

OPEN suppliers.txt file

Put contents of readfromfile’s parts\_id[4] and part\_quantity into suppliers.txt file next to “WAR\_Supplier” text

CLOSE fp3 (suppliers.txt) file

PRINT Parts Inventory Updated Successfully !

END IF

ELSE

PRINT Invalid Part ID !

END ELSE

OPEN war.txt file

FOR i = 0 to 4

Put contents of readfromfile’s parts\_name[i] array, parts\_id[i] array and parts\_quantity[i] array into war.txt file

END FOR

CLOSE fp1(war.txt) file

END IF

BREAK

END CASE

CASE 2, choice variable contains 2

PRINT You are a Distributor

PRINT You can provide parts to the assembly sections

PRINT 1. WBZ Warehouse

PRINT 2. WSL Warehouse

PRINT 3. WAR Warehouse

PRINT Enter your choice :

int choices1

choices1 = INPUT from user

IF choices1 = 1 THEN

OPEN file containing information of WBZ warehouse

int count = 0

char c

IF fp2 (wbz.txt file) = NULL (doesn’t exist)

PRINT Could not open file wbz.txt

END IF

FOR c = characters from the file (wbz.txt) TO EOF (end of file)

IF c = \n (newline)

count = count +1

END IF

END FOR

CLOSE fp2 file

FOR i = 0 to count

Put contents of f(wbz.txt) file into readfromfile’s (another instance of warehouse struct) parts\_name[i] array, parts\_id[i] array, and parts\_quantity[i] array

END FOR

CLOSE f(wbz.txt) file

PRINT You are distributing part from the WBZ Warehouse

PRINT You are WBZ Distributor

PRINT You can only Distribute parts from Warehouse 1

PRINT Current Parts and their Quantity in WBZ Warehouse:

FOR i = 0 to 4

PRINT “Parts Name :”, these are all the names of the parts which are stored in readfromfile’s parts\_name[i] array

PRINT “Parts ID : ”, these are the ID numbers of all the parts which are stored in readfromfile’s parts\_id[i] array

PRINT “Parts Quantity:”, these are the quantity of all the parts which are stored in readfromfile’s parts\_quantity[i] array

END FOR

PRINT Enter part id you want to distribute to assembly sections :

INT choice2

choice2 = INPUT from user

IF choice2 = 101 THEN

PRINT Enter the part quantity (-) :

INT part\_quantity

part\_quantity = INPUT from user

IF readfromfile’s parts\_quantity[0] (zero index) - part\_quantity <= 0 THEN

PRINT “You can't distribute more or equal than”, value inside readfromfile’s parts\_quantity[0], “parts of ”, name inside readfromfile’s parts\_name[0]

END IF

ELSE

readfromfile’s parts\_quantity[0] (zero index) = readfromfile’s parts\_quantity[0] - part\_quantity

PRINT “Updated Part quantity of ”, name inside readfromfile’s parts\_name[0] “is ”, value inside readfromfile’s parts\_quantity[0]

OPEN distributers.txt file

Put contents of readfromfile’s parts\_id[0] and part\_quantity into distributers.txt file next to “WBZ\_Distributer” text

CLOSE fp3 (distributers.txt) file

PRINT Parts Inventory Updated Successfully !

END ELSE

END IF

ELSE IF choice2 = 102 THEN

PRINT Enter the part quantity (-) :

INT part\_quantity

part\_quantity = INPUT from user

IF readfromfile’s parts\_quantity[1] (zero index) - part\_quantity <= 0 THEN

PRINT “You can't distribute more or equal than”, value inside readfromfile’s parts\_quantity[1], “parts of ”, name inside readfromfile’s parts\_name[1]

END IF

ELSE

readfromfile’s parts\_quantity[1] (zero index) = readfromfile’s parts\_quantity[1] - part\_quantity

PRINT “Updated Part quantity of ”, name inside readfromfile’s parts\_name[1] “is ”, value inside readfromfile’s parts\_quantity[1]

OPEN distributers.txt file

Put contents of readfromfile’s parts\_id[1] and part\_quantity into distributers.txt file next to “WBZ\_Distributer” text

CLOSE fp3 (distributers.txt) file

PRINT Parts Inventory Updated Successfully !

END ELSE

END IF

ELSE IF choice2 = 103 THEN

PRINT Enter the part quantity (-) :

INT part\_quantity

part\_quantity = INPUT from user

IF readfromfile’s parts\_quantity[2] (zero index) - part\_quantity <= 0 THEN

PRINT “You can't distribute more or equal than”, value inside readfromfile’s parts\_quantity[2], “parts of ”, name inside readfromfile’s parts\_name[2]

END IF

ELSE

readfromfile’s parts\_quantity[2] (zero index) = readfromfile’s parts\_quantity[2] - part\_quantity

PRINT “Updated Part quantity of ”, name inside readfromfile’s parts\_name[2] “is ”, value inside readfromfile’s parts\_quantity[2]

OPEN distributers.txt file

Put contents of readfromfile’s parts\_id[2] and part\_quantity into distributers.txt file next to “WBZ\_Distributer” text

CLOSE fp3 (suppliers.txt) file

PRINT Parts Inventory Updated Successfully !

END ELSE

END IF

ELSE IF choice2 = 104 THEN

PRINT Enter the part quantity (-) :

INT part\_quantity

part\_quantity = INPUT from user

IF readfromfile’s parts\_quantity[3] (zero index) - part\_quantity <= 0 THEN

PRINT “You can't distribute more or equal than”, value inside readfromfile’s parts\_quantity[3], “parts of ”, name inside readfromfile’s parts\_name[3]

END IF

ELSE

readfromfile’s parts\_quantity[3] (zero index) = readfromfile’s parts\_quantity[3] - part\_quantity

PRINT “Updated Part quantity of ”, name inside readfromfile’s parts\_name[3] “is ”, value inside readfromfile’s parts\_quantity[3]

OPEN distributers.txt file

Put contents of readfromfile’s parts\_id[3] and part\_quantity into distributers.txt file next to “WBZ\_Distributer” text

CLOSE fp3 (distributers.txt) file

PRINT Parts Inventory Updated Successfully !

END ELSE

END IF

ELSE IF choice2 = 105 THEN

PRINT Enter the part quantity (-) :

INT part\_quantity

part\_quantity = INPUT from user

IF readfromfile’s parts\_quantity[4] (zero index) - part\_quantity <= 0 THEN

PRINT “You can't distribute more or equal than”, value inside readfromfile’s parts\_quantity[4], “parts of ”, name inside readfromfile’s parts\_name[4]

END IF

ELSE

readfromfile’s parts\_quantity[4] (zero index) = readfromfile’s parts\_quantity[4] - part\_quantity

PRINT “Updated Part quantity of ”, name inside readfromfile’s parts\_name[4] “is ”, value inside readfromfile’s parts\_quantity[4]

OPEN distributers.txt file

Put contents of readfromfile’s parts\_id[4] and part\_quantity into distributers.txt file next to “WBZ\_Distributer” text

CLOSE fp3 (suppliers.txt) file

PRINT Parts Inventory Updated Successfully !

END ELSE

END IF

ELSE

PRINT Invalid Choice !

END ELSE

OPEN wbz.txt file

FOR i = 0 to 4

Put contents of readfromfile’s parts\_name[i] array, parts\_id[i] array and parts\_quantity[i] array into wbz.txt file

END FOR

CLOSE fp1(wbz.txt) file

END IF

IF choices1 = 2 THEN

OPEN file containing information of WSL warehouse

int count = 0

char c

IF fp2 (wsl.txt file) = NULL (doesn’t exist)

PRINT Could not open file wbz.txt

END IF

FOR c = characters from the file (wsl.txt) TO EOF (end of file)

IF c = \n (newline)

count = count +1

END IF

END FOR

CLOSE fp2 file

FOR i = 0 to count

Put contents of f(wsl.txt) file into readfromfile’s (another instance of warehouse struct) parts\_name[i] array, parts\_id[i] array, and parts\_quantity[i] array

END FOR

CLOSE f(wsl.txt) file

PRINT You are distributing part from the WSL Warehouse

PRINT You are WSL Distributor

PRINT You can only Distribute parts from Warehouse 2

PRINT Current Parts and their Quantity in WSL Warehouse:

FOR i = 0 to 4

PRINT “Parts Name :”, these are all the names of the parts which are stored in readfromfile’s parts\_name[i] array

PRINT “Parts ID : ”, these are the ID numbers of all the parts which are stored in readfromfile’s parts\_id[i] array

PRINT “Parts Quantity:”, these are the quantity of all the parts which are stored in readfromfile’s parts\_quantity[i] array

END FOR

PRINT Enter part id you want to distribute to assembly sections :

INT choice2

choice2 = INPUT from user

IF choice2 = 106 THEN

PRINT Enter the part quantity (-) :

INT part\_quantity

part\_quantity = INPUT from user

IF readfromfile’s parts\_quantity[0] (zero index) - part\_quantity <= 0 THEN

PRINT “You can't distribute more or equal than”, value inside readfromfile’s parts\_quantity[0], “parts of ”, name inside readfromfile’s parts\_name[0]

END IF

ELSE

readfromfile’s parts\_quantity[0] (zero index) = readfromfile’s parts\_quantity[0] - part\_quantity

PRINT “Updated Part quantity of ”, name inside readfromfile’s parts\_name[0] “is ”, value inside readfromfile’s parts\_quantity[0]

OPEN distributers.txt file

Put contents of readfromfile’s parts\_id[0] and part\_quantity into distributers.txt file next to “WSL\_Distributer” text

CLOSE fp3 (distributers.txt) file

PRINT Parts Inventory Updated Successfully !

END ELSE

END IF

ELSE IF choice2 = 107 THEN

PRINT Enter the part quantity (-) :

INT part\_quantity

part\_quantity = INPUT from user

IF readfromfile’s parts\_quantity[1] (zero index) - part\_quantity <= 0 THEN

PRINT “You can't distribute more or equal than”, value inside readfromfile’s parts\_quantity[1], “parts of ”, name inside readfromfile’s parts\_name[1]

END IF

ELSE

readfromfile’s parts\_quantity[1] (zero index) = readfromfile’s parts\_quantity[1] - part\_quantity

PRINT “Updated Part quantity of ”, name inside readfromfile’s parts\_name[1] “is ”, value inside readfromfile’s parts\_quantity[1]

OPEN distributers.txt file

Put contents of readfromfile’s parts\_id[1] and part\_quantity into distributers.txt file next to “WSL\_Distributer” text

CLOSE fp3 (distributers.txt) file

PRINT Parts Inventory Updated Successfully !

END ELSE

END IF

ELSE IF choice2 = 108 THEN

PRINT Enter the part quantity (-) :

INT part\_quantity

part\_quantity = INPUT from user

IF readfromfile’s parts\_quantity[2] (zero index) - part\_quantity <= 0 THEN

PRINT “You can't distribute more or equal than”, value inside readfromfile’s parts\_quantity[2], “parts of ”, name inside readfromfile’s parts\_name[2]

END IF

ELSE

readfromfile’s parts\_quantity[2] (zero index) = readfromfile’s parts\_quantity[2] - part\_quantity

PRINT “Updated Part quantity of ”, name inside readfromfile’s parts\_name[2] “is ”, value inside readfromfile’s parts\_quantity[2]

OPEN distributers.txt file

Put contents of readfromfile’s parts\_id[2] and part\_quantity into distributers.txt file next to “WSL\_Distributer” text

CLOSE fp3 (distributers.txt) file

PRINT Parts Inventory Updated Successfully !

END ELSE

END IF

ELSE IF choice2 = 109 THEN

PRINT Enter the part quantity (-) :

INT part\_quantity

part\_quantity = INPUT from user

IF readfromfile’s parts\_quantity[3] (zero index) - part\_quantity <= 0 THEN

PRINT “You can't distribute more or equal than”, value inside readfromfile’s parts\_quantity[3], “parts of ”, name inside readfromfile’s parts\_name[3]

END IF

ELSE

readfromfile’s parts\_quantity[3] (zero index) = readfromfile’s parts\_quantity[3] - part\_quantity

PRINT “Updated Part quantity of ”, name inside readfromfile’s parts\_name[3] “is ”, value inside readfromfile’s parts\_quantity[3]

OPEN distributers.txt file

Put contents of readfromfile’s parts\_id[3] and part\_quantity into distributers.txt file next to “WSL\_Distributer” text

CLOSE fp3 (distributers.txt) file

PRINT Parts Inventory Updated Successfully !

END ELSE

END IF

ELSE IF choice2 = 110 THEN

PRINT Enter the part quantity (-) :

INT part\_quantity

part\_quantity = INPUT from user

IF readfromfile’s parts\_quantity[4] (zero index) - part\_quantity <= 0 THEN

PRINT “You can't distribute more or equal than”, value inside readfromfile’s parts\_quantity[4], “parts of ”, name inside readfromfile’s parts\_name[4]

END IF

ELSE

readfromfile’s parts\_quantity[4] (zero index) = readfromfile’s parts\_quantity[4] - part\_quantity

PRINT “Updated Part quantity of ”, name inside readfromfile’s parts\_name[4] “is ”, value inside readfromfile’s parts\_quantity[4]

OPEN distributers.txt file

Put contents of readfromfile’s parts\_id[4] and part\_quantity into distributers.txt file next to “WSL\_Distributer” text

CLOSE fp3 (distributers.txt) file

PRINT Parts Inventory Updated Successfully !

END ELSE

END IF

ELSE

PRINT Invalid Choice !

END ELSE

OPEN wsl.txt file

FOR i = 0 to 4

Put contents of readfromfile’s parts\_name[i] array, parts\_id[i] array and parts\_quantity[i] array into wsl.txt file

END FOR

CLOSE fp1(wsl.txt) file

END IF

IF choices1 = 3 THEN

OPEN file containing information of WAR warehouse

int count = 0

char c

IF fp2 (war.txt file) = NULL (doesn’t exist)

PRINT Could not open file war.txt

END IF

FOR c = characters from the file (war.txt) TO EOF (end of file)

IF c = \n (newline)

count = count +1

END IF

END FOR

CLOSE fp2 file

FOR i = 0 to count

Put contents of f(war.txt) file into readfromfile’s (another instance of warehouse struct) parts\_name[i] array, parts\_id[i] array, and parts\_quantity[i] array

END FOR

CLOSE f(war.txt) file

PRINT You are distributing part from the WAR Warehouse

PRINT You are WAR Distributor

PRINT You can only Distribute parts from Warehouse 3

PRINT Current Parts and their Quantity in WAR Warehouse:

FOR i = 0 to 4

PRINT “Parts Name :”, these are all the names of the parts which are stored in readfromfile’s parts\_name[i] array

PRINT “Parts ID : ”, these are the ID numbers of all the parts which are stored in readfromfile’s parts\_id[i] array

PRINT “Parts Quantity:”, these are the quantity of all the parts which are stored in readfromfile’s parts\_quantity[i] array

END FOR

PRINT Enter part id you want to distribute to assembly sections :

INT choice2

choice2 = INPUT from user

IF choice2 = 111 THEN

PRINT Enter the part quantity (-) :

INT part\_quantity

part\_quantity = INPUT from user

IF readfromfile’s parts\_quantity[0] (zero index) - part\_quantity <= 0 THEN

PRINT “You can't distribute more or equal than”, value inside readfromfile’s parts\_quantity[0], “parts of ”, name inside readfromfile’s parts\_name[0]

END IF

ELSE

readfromfile’s parts\_quantity[0] (zero index) = readfromfile’s parts\_quantity[0] - part\_quantity

PRINT “Updated Part quantity of ”, name inside readfromfile’s parts\_name[0] “is ”, value inside readfromfile’s parts\_quantity[0]

OPEN distributers.txt file

Put contents of readfromfile’s parts\_id[0] and part\_quantity into distributers.txt file next to “WAR\_Distributer” text

CLOSE fp3 (distributers.txt) file

PRINT Parts Inventory Updated Successfully !

END ELSE

END IF

ELSE IF choice2 = 112 THEN

PRINT Enter the part quantity (-) :

INT part\_quantity

part\_quantity = INPUT from user

IF readfromfile’s parts\_quantity[1] (zero index) - part\_quantity <= 0 THEN

PRINT “You can't distribute more or equal than”, value inside readfromfile’s parts\_quantity[1], “parts of ”, name inside readfromfile’s parts\_name[1]

END IF

ELSE

readfromfile’s parts\_quantity[1] (zero index) = readfromfile’s parts\_quantity[1] - part\_quantity

PRINT “Updated Part quantity of ”, name inside readfromfile’s parts\_name[1] “is ”, value inside readfromfile’s parts\_quantity[1]

OPEN distributers.txt file

Put contents of readfromfile’s parts\_id[1] and part\_quantity into distributers.txt file next to “WAR\_Distributer” text

CLOSE fp3 (distributers.txt) file

PRINT Parts Inventory Updated Successfully !

END ELSE

END IF

ELSE IF choice2 = 113 THEN

PRINT Enter the part quantity (-) :

INT part\_quantity

part\_quantity = INPUT from user

IF readfromfile’s parts\_quantity[2] (zero index) - part\_quantity <= 0 THEN

PRINT “You can't distribute more or equal than”, value inside readfromfile’s parts\_quantity[2], “parts of ”, name inside readfromfile’s parts\_name[2]

END IF

ELSE

readfromfile’s parts\_quantity[2] (zero index) = readfromfile’s parts\_quantity[2] - part\_quantity

PRINT “Updated Part quantity of ”, name inside readfromfile’s parts\_name[2] “is ”, value inside readfromfile’s parts\_quantity[2]

OPEN distributers.txt file

Put contents of readfromfile’s parts\_id[2] and part\_quantity into distributers.txt file next to “WAR\_Distributer” text

CLOSE fp3 (distributers.txt) file

PRINT Parts Inventory Updated Successfully !

END ELSE

END IF

ELSE IF choice2 = 114 THEN

PRINT Enter the part quantity (-) :

INT part\_quantity

part\_quantity = INPUT from user

IF readfromfile’s parts\_quantity[3] (zero index) - part\_quantity <= 0 THEN

PRINT “You can't distribute more or equal than”, value inside readfromfile’s parts\_quantity[3], “parts of ”, name inside readfromfile’s parts\_name[3]

END IF

ELSE

readfromfile’s parts\_quantity[3] (zero index) = readfromfile’s parts\_quantity[3] - part\_quantity

PRINT “Updated Part quantity of ”, name inside readfromfile’s parts\_name[3] “is ”, value inside readfromfile’s parts\_quantity[3]

OPEN distributers.txt file

Put contents of readfromfile’s parts\_id[3] and part\_quantity into distributers.txt file next to “WAR\_Distributer” text

CLOSE fp3 (distributers.txt) file

PRINT Parts Inventory Updated Successfully !

END ELSE

END IF

ELSE IF choice2 = 115 THEN

PRINT Enter the part quantity (-) :

INT part\_quantity

part\_quantity = INPUT from user

IF readfromfile’s parts\_quantity[4] (zero index) - part\_quantity <= 0 THEN

PRINT “You can't distribute more or equal than”, value inside readfromfile’s parts\_quantity[4], “parts of ”, name inside readfromfile’s parts\_name[4]

END IF

ELSE

readfromfile’s parts\_quantity[4] (zero index) = readfromfile’s parts\_quantity[4] - part\_quantity

PRINT “Updated Part quantity of ”, name inside readfromfile’s parts\_name[4] “is ”, value inside readfromfile’s parts\_quantity[4]

OPEN distributers.txt file

Put contents of readfromfile’s parts\_id[4] and part\_quantity into distributers.txt file next to “WAR\_Distributer” text

CLOSE fp3 (distributers.txt) file

PRINT Parts Inventory Updated Successfully !

END ELSE

END IF

ELSE

PRINT Invalid Choice !

END ELSE

OPEN war.txt file

FOR i = 0 to 4

Put contents of readfromfile’s parts\_name[i] array, parts\_id[i] array and parts\_quantity[i] array into war.txt file

END FOR

CLOSE fp1(war.txt) file

END IF

BREAK

END CASE

DEFAULT

PRINT \*\*\*\*\*\*\*Enter Correct Input\*\*\*\*\*\*\*

BREAK

END SWITCH

END PartsInventoryUpdate Function

**4:PartsInventroyTracking Function**

BEGIN PartsInventroyTracking Function

INT choice

PRINT 1. Total available quantity of all parts sorted in ascending order

PRINT Enter your choice :

choice = INPUT from user

SWITCH choice

CASE 1

INT i, j, n = 5, m

INT idxArray[5]

FLOAT a, b

INT total\_quantity = 0

INT total\_quantity\_array[5]

PRINT Select which warehouse to sort

PRINT 1. WBZ Warehouse

PRINT 2. WSL Warehouse

PRINT 3. WAR Warehouse

PRINT Enter your choice :

INT choice1

choice1 = INPUT from user

IF choice1 = 1 THEN

OPEN file containing information of WBZ warehouse

INT count = 0

CHAR c

IF fp2 (wbz.txt file) = NULL (doesn’t exist)

PRINT Could not open file wbz.txt

END IF

FOR c = characters from the file (wbz.txt) TO EOF (end of file)

IF c = \n (newline)

count = count +1

END IF

END FOR

CLOSE fp2 file

FOR i = 0 to count

Put contents of f(wbz.txt) file into readfromfile’s (another instance of warehouse struct) parts\_name[i] array, parts\_id[i] array, and parts\_quantity[i] array

idxArray[i] = i

END FOR

CLOSE f(wbz.txt) file

FOR i=0 to 4

Put content from readfromfile and parts\_quantity[i] into total\_quantity\_array[i] Add the contents of total\_quantity and total\_quantity\_array[i] and put into total\_quantity

END FOR

PRINT: Total available quantity of all parts is %d", total\_quantity

PRINT: Sorted in Original order

FOR i=0 to 4

PRINT: readfromfile->parts\_id[i], readfromfile->parts\_name[i], total\_quantity\_array[i]

END FOR

FOR i=0 to n

FOR i+1 to n

IF (readfromfile,parts\_quantity[idxArray[i]] greater then readfromfile, parts\_quantity[idxArray[j]])

THEN

Put idxArray[i] into a

Put idxArray[j] into idxArray[i]

Put a into idxArray[j]

END IF

END FOR

END FOR

PRINT: Sorted in Ascending order

FOR i=0 to 4

PRINT: parts\_id[idxArray[i]] into readfromfile and parts\_name[idxArray[i]] into readfromfile

END FOR

INT int less\_than\_10 = 0

FOR i=0 to 4

IF parts\_quantity of readfromfile is less then 10

THEN  
increment less\_than\_10

END IF

IF less\_than\_10 equal to 0

THEN  
PRINT: No parts has stock quantity less than 10 units

END IF

ELSE IF less\_than\_10 is less the or equal to 10

THEN

PRINT: Records of all parts that has stock quantity less than 10 units

IF parts\_quantity of readfromfile is less then 10

THEN

PRINT:parts\_id,parts\_name,total\_quantity\_array into readfromfile

END IF  
END ELSE IF

BREAK

ELSE  
PRINT: Something went wrong

END ELSE  
ELSE IF

choice1 = 2

OPEN f (wsl.txt)

INT count=0

CHAR filename[]=(wsl.txt)

CHAR C

OPEN fp2( wsl.tx)

IF fp2 =NULL

THEN  
PRINT: Could not open file

END IF  
FOR c = characters from the file (wbz.txt) TO EOF (end of file)

IF c = \n (newline)

count = count +1

END IF

END FOR

CLOSE fp2 file

FOR i = 0 to count

Put contents of f(wbz.txt) file into readfromfile’s (another instance of warehouse struct) parts\_name[i] array, parts\_id[i] array, and parts\_quantity[i] array

idxArray[i] = i

END FOR

CLOSE f(wbz.txt) file

FOR i=0 to 4

Put parts\_quantity of readfromfile into total\_quantity\_array

Put the addition of total\_quantity and total\_quantity\_array into total\_quantity

END FOR

PRINT Total available quantity of all parts is %d

PRINT Sorted in Original order

FOR i=0 to 4

PRINT:parts\_id,parts\_name,total\_quantity\_array into readfromfile

END FOR

FOR i=0 to n

FOR i+1 to n

IF (readfromfile,parts\_quantity[idxArray[i]] greater then readfromfile, parts\_quantity[idxArray[j]])

THEN

Put idxArray[i] into a

Put idxArray[j] into idxArray[i]

Put a into idxArray[j]

END IF

END FOR

END FOR

PRINT Sorted in Ascending order

FOR i=0 to 4

PRINT parts\_id[idxArray[i]],parts\_name[idxArray[i]],>parts\_quantity[idxArray[i] into readfromfile

INT less\_than\_10 = 0

FOR i=0 to 4

IF

readfromfile->parts\_quantity[i] lesss then 10

THEN

Increment the less\_than\_10

END IF

END FOR

IF less\_than\_10 equal to 0

PRINT No parts has stock quantity less than 10 units

END IF

ELSE IF

less\_than\_10 less then or equal to 10

PRINT tRecords of all parts that has stock quantity less than 10 units

FOR i=0 to 4

IF readfromfile->parts\_quantity[i] less then 10

PRINT readfromfile->parts\_id[i], readfromfile->parts\_name[i], total\_quantity\_array[i]

END IF  
END FOR

BREAK  
ELSE  
PRINT Something went wrong

ELSE END

BREAK

DEFAULT

PRINT Enter Correct Input

BREAK

ENF OF PartsInventoryTracking Function

**5: PartsInventorySearch Function**

BEGIN of PartsInventorySearch Function

INT search\_id

PRINT Enter the ID of the part you want to search:

SCAN Get ID from user

IF search\_id less then 100 and less then or equal to 105

THEN

int count = 0

char c

IF fp2 (wbz.txt file) = NULL (doesn’t exist)

PRINT Could not open file wbz.txt

END IF

FOR c = characters from the file (wbz.txt) TO EOF (end of file)

IF c = \n (newline)

count = count +1

END IF

END FOR

CLOSE fp2 file

FOR i = 0 to count

Put contents of f(wbz.txt) file into readfromfile’s (another instance of warehouse struct) parts\_name[i] array, parts\_id[i] array, and parts\_quantity[i] array

END FOR

CLOSE fp2 file

PRINT readfromfile->parts\_id[search\_id - 106], readfromfile

>parts\_name[search\_id - 106], readfromfile->parts\_quantity[search\_id - 106]

ELSE IF

search\_id greater then 110 and less then or equal to 115

IF fp2 (wbz.txt file) = NULL (doesn’t exist)

PRINT Could not open file wbz.txt

END IF

FOR c = characters from the file (wbz.txt) TO EOF (end of file)

IF c = \n (newline)

count = count +1

END IF

END FOR

CLOSE fp2 file

FOR i = 0 to count

Put contents of f(wbz.txt) file into readfromfile’s (another instance of warehouse struct) parts\_name[i] array, parts\_id[i] array, and parts\_quantity[i] array

END FOR

CLOSE f file

PRINT readfromfile->parts\_id[search\_id - 111], readfromfile- >parts\_name[search\_id - 111], readfromfile->parts\_quantity[search\_id - 111]

END ELSE IF

ELSE  
PRINT ID not found

END ELSE