SFWRENG 2MP3 – Programming for Mechatronics Fall 2018

**Assignment #7**

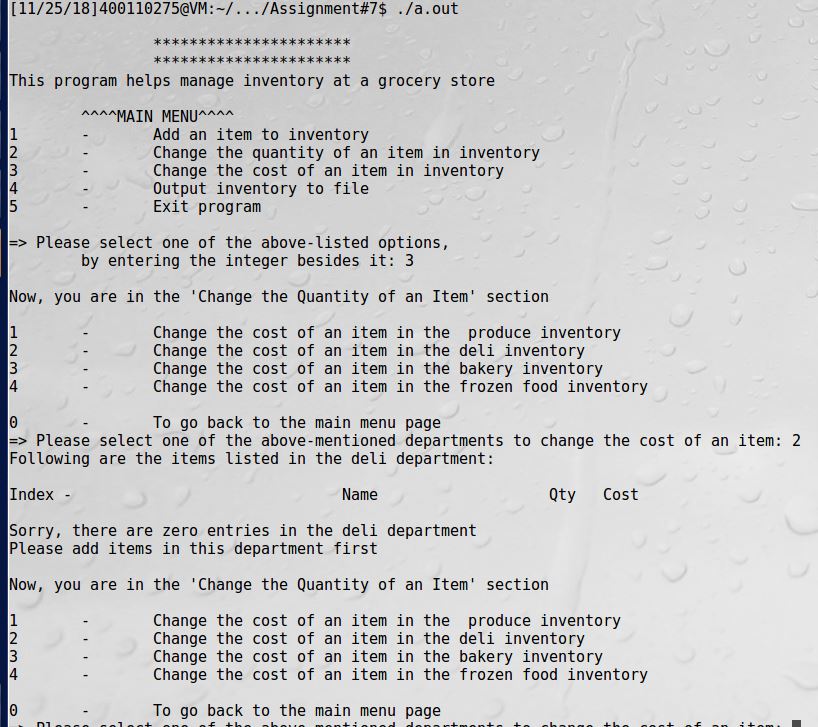
|  |  |
| --- | --- |
| Exercise 7 Solution | **Submitted By: HARNEET SINGH, #400110275, SINGHH76@MCMASTER.CA** |
| Question # | Answer |
| Entire Code | #include <stdio.h>  #include <string.h>  #define stock 100  struct groceryStore{  char itemName[150];  int itemQty;  float itemCost;  }produce[stock], deli[stock], bakery[stock], frozenFoods[stock];  void addAnItem();  void changeQty();  void changeCost();  void outputInventory();  int printInventory();  void printAllInventory();  int main()  {  int input = 0; //intializing a variable to retrieve input from the user  puts("\n\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");  puts("This program helps manage inventory at a grocery store\n");    mainMenu:  //creating an iterating loop to get user input  do {  puts("\t^^^^MAIN MENU^^^^");  puts("1 - Add an item to inventory");  puts("2 - Change the quantity of an item in inventory");  puts("3 - Change the cost of an item in inventory");  puts("4 - Output inventory to file");  puts("5 - Exit program\n");  printf("=> Please select one of the above-listed options, \n\tby entering the integer besides it: ");  scanf("%d", &input);  if (input >= 1 && input <=5) //condition if user enters an interger out of range  {  if (input == 1) //condition to enter subsequent department levels  {  int subInput = 0;  do{  puts("\nNow, you are in the 'Add An Item' section\n");  puts("1 - Add item to produce inventory");  puts("2 - Add item to deli inventory");  puts("3 - Add item to bakery inventory");  puts("4 - Add item to frozen food inventory\n");  puts("0 - To go back to the main menu page");  printf("=> Please select one of the above-mentioned departments to add an item: ");  scanf("%d", &subInput);  static int produce\_index = 0, deli\_index = 0, bakery\_index = 0, frozenFood\_index = 0;  switch (subInput)  {  case 1:  addAnItem(produce, "produce", produce\_index);  produce\_index++;  break;  case 2:  addAnItem(deli, "deli", deli\_index);  deli\_index++;  break;  case 3:  addAnItem(bakery, "bakery", bakery\_index);  bakery\_index++;  break;  case 4:  addAnItem(frozenFoods, "Frozen Food", frozenFood\_index);  frozenFood\_index++;  break;  case 0:  goto mainMenu;  break;  default:  puts("!!!Please enter a valid input");  break;  }  } while (subInput);  }  else if (input == 2)  {  int subInput = 0;  do{  puts("\nNow, you are in the 'Change the Quantity of an Item' section\n");  puts("1 - Change the quantity of an item in the produce inventory");  puts("2 - Change the quantity of an item in the deli inventory");  puts("3 - Change the quantity of an item in the bakery inventory");  puts("4 - Change the quantity of an item in the frozen food inventory\n");  puts("0 - To go back to the main menu page");  printf("=> Please select one of the above-mentioned departments to change the quantity of an item: ");  scanf("%d", &subInput);  int produce\_index = 0, deli\_index = 0, bakery\_index = 0, frozenFood\_index = 0;  switch (subInput)  {  case 1:  produce\_index = printInventory(produce, "produce", "Quantity");  if (produce\_index != -1)  {  changeQty(&produce[produce\_index].itemQty, "produce", produce);  break;  }  else  break;  case 2:  deli\_index = printInventory(deli, "deli", "Quantity");  if (deli\_index != -1)  {  changeQty(&deli[deli\_index].itemQty, "deli", deli);  break;  }  else  break;  case 3:  bakery\_index = printInventory(bakery, "bakery", "Quantity");  if (bakery\_index != -1)  {  changeQty(&bakery[bakery\_index].itemQty, "bakery", bakery);  break;  }  else  break;  case 4:  frozenFood\_index = printInventory(frozenFoods, "Frozen Food", "Quantity");  if (frozenFood\_index != -1)  {  changeQty(&frozenFoods[frozenFood\_index].itemQty, "Frozen Food", frozenFoods);  break;  }  else  break;  case 0:  goto mainMenu;  break;  default:  puts("!!!Please enter a valid input");  break;  }  } while (subInput);  }  else if (input == 3)  {  int subInput = 0;  do{  puts("\nNow, you are in the 'Change the Quantity of an Item' section\n");  puts("1 - Change the cost of an item in the produce inventory");  puts("2 - Change the cost of an item in the deli inventory");  puts("3 - Change the cost of an item in the bakery inventory");  puts("4 - Change the cost of an item in the frozen food inventory\n");  puts("0 - To go back to the main menu page");  printf("=> Please select one of the above-mentioned departments to change the cost of an item: ");  scanf("%d", &subInput);  int produce\_index = 0, deli\_index = 0, bakery\_index = 0, frozenFood\_index = 0;  switch (subInput)  {  case 1:  produce\_index = printInventory(produce, "produce", "Cost");  if (produce\_index != -1)  {  changeCost(&produce[produce\_index].itemCost, "produce", produce);  break;  }  else  break;  case 2:  deli\_index = printInventory(deli, "deli", "Cost");  if (deli\_index != -1)  {  changeCost(&deli[deli\_index].itemCost, "deli", deli);  break;  }  else  break;  case 3:  bakery\_index = printInventory(bakery, "bakery", "Cost");  if (bakery\_index != -1)  {  changeCost(&bakery[bakery\_index].itemCost, "bakery", bakery);  break;  }  else  break;  case 4:  frozenFood\_index = printInventory(frozenFoods, "Frozen Food", "Cost");  if (frozenFood\_index != -1)  {  changeCost(&frozenFoods[frozenFood\_index].itemCost, "Frozen Food", frozenFoods);  break;  }  else  break;  case 0:  goto mainMenu;  break;  default:  puts("!!!Please enter a valid input");  break;  }  } while (subInput);  }  else if (input == 4)  {  printAllInventory(produce, deli, bakery, frozenFoods);  int saveDecision = -1;  printf("Would you like to continue and transfer data to a .txt file? \n=> Enter 1 to continue and save (0 to modify):");  scanf("%d", &saveDecision);  if (saveDecision == 1)  outputInventory(produce, deli, bakery, frozenFoods);  else if (saveDecision == 0)  goto mainMenu;  else  break;  }  else  {  puts("Program is exiting now...\nProgram terminated successfully");  return 0;  }  }  else  puts ("\n!!!!Please select a valid number between 1 and 5\n");  }while (1);  return 0;  }  void addAnItem(struct groceryStore fDepartment[], char section[50], int index)  {  char Name[stock];  int Quantity;  float Cost;  printf("\n=>Enter name of the item for %s department: ", section);  scanf(" %[^\n]s", Name);  printf("Enter quantity of %s: ", Name);  scanf("%d", &Quantity);  printf("Enter cost of %s: ", Name);  scanf("%f", &Cost);  //storing values in the inventory  strcpy(fDepartment[index].itemName, Name);  fDepartment[index].itemQty = Quantity;  fDepartment[index].itemCost = Cost;  }  void changeQty(int \*ptrQty, char section[50], struct groceryStore fDepartment[])  {  int newQty = 0;  printf("\n=> Please enter the new quantity number: ");  scanf("%d", &newQty);  \*ptrQty = newQty;  printf("%5s -\t\t%25s\t\t%7s\t%7s\n", "Index", "Name", "Qty", "Cost\n");  for(int i = 0; (fDepartment[i].itemQty) != 0 && (i < 100); i++) //iterating through the entire inventory  {  printf("%5d -\t\t%25s\t\t%7d\t%7.2f", i+1, fDepartment[i].itemName, fDepartment[i].itemQty, fDepartment[i].itemCost);  puts("");  }  }  void changeCost(float \*ptrCost, char section[50], struct groceryStore fDepartment[])  {  float newCost = 0;    printf("\n=> Please enter the new cost amount: ");  scanf("%f", &newCost);  \*ptrCost = newCost;  printf("%5s -\t\t%25s\t\t%7s\t%7s\n", "Index", "Name", "Qty", "Cost\n");  for(int i = 0; (fDepartment[i].itemQty) != 0 && (i < 100); i++) //iterating through the entire inventory  {  printf("%5d -\t\t%25s\t\t%7d\t%7.2f", i+1, fDepartment[i].itemName, fDepartment[i].itemQty, fDepartment[i].itemCost);  puts("");  }  }  void outputInventory(struct groceryStore pInventory[], struct groceryStore dInventory[], struct groceryStore bInventory[], struct groceryStore fInventory[])  {  FILE \*file;  fopen("Inventory.txt", "w");  file = fopen("Inventory.txt", "a+");  fprintf(file, "%s\n", "File Created by HARNEET SINGH, 400110275, singhh76@mcmaster.ca");  fprintf(file, "\nGROCERY STORE - INVENTORY\n\n");    if (pInventory[0].itemQty != 0)  {  fprintf(file, "%s\n", "Produce Department Inventory:");  fprintf(file, "%5s -\t\t%25s\t\t%7s\t%7s\n", "Index", "Name", "Qty", "Cost\n");  for(int i = 0; (pInventory[i].itemQty) != 0 && (i < 100); i++) //iterating through the entire inventory  {  fprintf(file, "%5d -\t\t%25s\t\t%7d\t%7.2f\n", i+1, pInventory[i].itemName, pInventory[i].itemQty, pInventory[i].itemCost);  }  }  if (dInventory[0].itemQty != 0)  {  fprintf(file, "\n%s\n", "Deli Department Inventory:");  fprintf(file, "%5s -\t\t%25s\t\t%7s\t%7s\n", "Index", "Name", "Qty", "Cost\n");  for(int i = 0; (dInventory[i].itemQty) != 0 && (i < 100); i++) //iterating through the entire inventory  {  fprintf(file, "%5d -\t\t%25s\t\t%7d\t%7.2f\n", i+1, dInventory[i].itemName, dInventory[i].itemQty, dInventory[i].itemCost);  }  }  if (bInventory[0].itemQty != 0)  {  fprintf(file, "\n%s\n", "Bakery Department Inventory:");  fprintf(file, "%5s -\t\t%25s\t\t%7s\t%7s\n", "Index", "Name", "Qty", "Cost\n");  for(int i = 0; (bInventory[i].itemQty) != 0 && (i < 100); i++) //iterating through the entire inventory  {  fprintf(file,"%5d -\t\t%25s\t\t%7d\t%7.2f\n", i+1, bInventory[i].itemName, bInventory[i].itemQty, bInventory[i].itemCost);  }  }    if (fInventory[0].itemQty != 0)  {  fprintf(file, "\n%s\n", "Frozen-Foods Department Inventory:");  fprintf(file, "%5s -\t\t%25s\t\t%7s\t%7s\n", "Index", "Name", "Qty", "Cost\n");  for(int i = 0; (fInventory[i].itemQty) != 0 && (i < 100); i++) //iterating through the entire inventory  {  fprintf(file, "%5d -\t\t%25s\t\t%7d\t%7.2f\n", i+1, fInventory[i].itemName, fInventory[i].itemQty, fInventory[i].itemCost);  }  }  fclose(file);  printf("\n\*\*\*\*File successfully saved\*\*\*\*\n");  }  int printInventory(struct groceryStore fInventory[], char section[50], char attribute[20]) //print item details in the store  {  int count = 0;  printf("Following are the items listed in the %s department:\n\n", section);  printf("%5s -\t\t%25s\t\t%7s\t%7s\n", "Index", "Name", "Qty", "Cost\n");  for(int i = 0; (fInventory[i].itemQty) != 0 && (i < 100); i++) //iterating through the entire inventory  {  printf("%5d -\t\t%25s\t\t%7d\t%7.2f\n", i+1, fInventory[i].itemName, fInventory[i].itemQty, fInventory[i].itemCost);  count++;  }  int item\_index = -1;  if (count != 0)  {  puts("From the displayed list, please take note of the index number of the item which needs to be changed");  printf("=> Enter the exact index number here, to alter its %s: ", attribute);  scanf("%d", &item\_index);  }  else  {  printf("Sorry, there are zero entries in the %s department", section);  puts("\nPlease add items in this department first");  return -1;  }  //printf("\nAddress of the quantity: %p\n", &fInventory[item\_index-1].itemQty);  return (item\_index-1);  }  void printAllInventory(struct groceryStore pInventory[], struct groceryStore dInventory[], struct groceryStore bInventory[], struct groceryStore fInventory[])  {  puts("Following details will be printed in the file, please verify:");  printf("%5s -\t\t%25s\t\t%7s\t%7s\n", "Index", "Name", "Qty", "Cost\n");  if (pInventory[0].itemQty != 0)  puts("Produce Department:");  for(int i = 0; (pInventory[i].itemQty) != 0 && (i < 100); i++) //iterating through the entire inventory  {  printf("%5d -\t\t%25s\t\t%7d\t%7.2f\n", i+1, pInventory[i].itemName, pInventory[i].itemQty, pInventory[i].itemCost);  }  if (dInventory[0].itemQty != 0)  puts("Deli Department:");  for(int i = 0; (dInventory[i].itemQty) != 0 && (i < 100); i++) //iterating through the entire inventory  {  printf("%5d -\t\t%25s\t\t%7d\t%7.2f\n", i+1, dInventory[i].itemName, dInventory[i].itemQty, dInventory[i].itemCost);  }  if (bInventory[0].itemQty != 0)  puts("Bakery Department:");  for(int i = 0; (bInventory[i].itemQty) != 0 && (i < 100); i++) //iterating through the entire inventory  {  printf("%5d -\t\t%25s\t\t%7d\t%7.2f\n", i+1, bInventory[i].itemName, bInventory[i].itemQty, bInventory[i].itemCost);  }  if (fInventory[0].itemQty != 0)  puts("Frozen-Foods Department:");  for(int i = 0; (fInventory[i].itemQty) != 0 && (i < 100); i++) //iterating through the entire inventory  {  printf("%5d -\t\t%25s\t\t%7d\t%7.2f\n", i+1, fInventory[i].itemName, fInventory[i].itemQty, fInventory[i].itemCost);  }  } |
| #1-a (Struct and Variable both created here) | #include <stdio.h>  #include <string.h>  #define stock 100  struct groceryStore{  char itemName[150];  int itemQty;  float itemCost;  }produce[stock], deli[stock], bakery[stock], frozenFoods[stock]; |
| #1-b | void addAnItem();  void changeQty();  void changeCost();  void outputInventory();  int printInventory();  void printAllInventory(); |
| #1-c | int main()  {  int input = 0; //intializing a variable to retrieve input from the user  puts("\n\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");  puts("This program helps manage inventory at a grocery store\n");    mainMenu:  //creating an iterating loop to get user input  do {  puts("\t^^^^MAIN MENU^^^^");  puts("1 - Add an item to inventory");  puts("2 - Change the quantity of an item in inventory");  puts("3 - Change the cost of an item in inventory");  puts("4 - Output inventory to file");  puts("5 - Exit program\n");  printf("=> Please select one of the above-listed options, \n\tby entering the integer besides it: ");  scanf("%d", &input); |
|  |
| #1-d | if (input >= 1 && input <=5) //condition if user enters an interger out of range  {  if (input == 1) //condition to enter subsequent department levels  {  int subInput = 0;  do{  puts("\nNow, you are in the 'Add An Item' section\n");  puts("1 - Add item to produce inventory");  puts("2 - Add item to deli inventory");  puts("3 - Add item to bakery inventory");  puts("4 - Add item to frozen food inventory\n");  puts("0 - To go back to the main menu page");  printf("=> Please select one of the above-mentioned departments to add an item: ");  scanf("%d", &subInput);  static int produce\_index = 0, deli\_index = 0, bakery\_index = 0, frozenFood\_index = 0;  switch (subInput)  {  case 1:  addAnItem(produce, "produce", produce\_index);  produce\_index++;  break;  case 2:  addAnItem(deli, "deli", deli\_index);  deli\_index++;  break;  case 3:  addAnItem(bakery, "bakery", bakery\_index);  bakery\_index++;  break;  case 4:  addAnItem(frozenFoods, "Frozen Food", frozenFood\_index);  frozenFood\_index++;  break;  case 0:  goto mainMenu;  break;  default:  puts("!!!Please enter a valid input");  break;  }  } while (subInput);  } |
|  |
| #1-e | else if (input == 2)  {  int subInput = 0;  do{  puts("\nNow, you are in the 'Change the Quantity of an Item' section\n");  puts("1 - Change the quantity of an item in the produce inventory");  puts("2 - Change the quantity of an item in the deli inventory");  puts("3 - Change the quantity of an item in the bakery inventory");  puts("4 - Change the quantity of an item in the frozen food inventory\n");  puts("0 - To go back to the main menu page");  printf("=> Please select one of the above-mentioned departments to change the quantity of an item: ");  scanf("%d", &subInput);  int produce\_index = 0, deli\_index = 0, bakery\_index = 0, frozenFood\_index = 0;  switch (subInput)  {  case 1:  produce\_index = printInventory(produce, "produce", "Quantity");  if (produce\_index != -1)  {  changeQty(&produce[produce\_index].itemQty, "produce", produce);  break;  }  else  break;  case 2:  deli\_index = printInventory(deli, "deli", "Quantity");  if (deli\_index != -1)  {  changeQty(&deli[deli\_index].itemQty, "deli", deli);  break;  }  else  break;  case 3:  bakery\_index = printInventory(bakery, "bakery", "Quantity");  if (bakery\_index != -1)  {  changeQty(&bakery[bakery\_index].itemQty, "bakery", bakery);  break;  }  else  break;  case 4:  frozenFood\_index = printInventory(frozenFoods, "Frozen Food", "Quantity");  if (frozenFood\_index != -1)  {  changeQty(&frozenFoods[frozenFood\_index].itemQty, "Frozen Food", frozenFoods);  break;  }  else  break;  case 0:  goto mainMenu;  break;  default:  puts("!!!Please enter a valid input");  break;  }  } while (subInput);  } |
|  |
| #1-f | else if (input == 3)  {  int subInput = 0;  do{  puts("\nNow, you are in the 'Change the Quantity of an Item' section\n");  puts("1 - Change the cost of an item in the produce inventory");  puts("2 - Change the cost of an item in the deli inventory");  puts("3 - Change the cost of an item in the bakery inventory");  puts("4 - Change the cost of an item in the frozen food inventory\n");  puts("0 - To go back to the main menu page");  printf("=> Please select one of the above-mentioned departments to change the cost of an item: ");  scanf("%d", &subInput);  int produce\_index = 0, deli\_index = 0, bakery\_index = 0, frozenFood\_index = 0;  switch (subInput)  {  case 1:  produce\_index = printInventory(produce, "produce", "Cost");  if (produce\_index != -1)  {  changeCost(&produce[produce\_index].itemCost, "produce", produce);  break;  }  else  break;  case 2:  deli\_index = printInventory(deli, "deli", "Cost");  if (deli\_index != -1)  {  changeCost(&deli[deli\_index].itemCost, "deli", deli);  break;  }  else  break;  case 3:  bakery\_index = printInventory(bakery, "bakery", "Cost");  if (bakery\_index != -1)  {  changeCost(&bakery[bakery\_index].itemCost, "bakery", bakery);  break;  }  else  break;  case 4:  frozenFood\_index = printInventory(frozenFoods, "Frozen Food", "Cost");  if (frozenFood\_index != -1)  {  changeCost(&frozenFoods[frozenFood\_index].itemCost, "Frozen Food", frozenFoods);  break;  }  else  break;  case 0:  goto mainMenu;  break;  default:  puts("!!!Please enter a valid input");  break;  }  } while (subInput);  } |
|  |
| #1-g | else if (input == 4)  {  printAllInventory(produce, deli, bakery, frozenFoods);  int saveDecision = -1;  printf("Would you like to continue and transfer data to a .txt file? \n=> Enter 1 to continue and save (0 to modify):");  scanf("%d", &saveDecision);  if (saveDecision == 1)  outputInventory(produce, deli, bakery, frozenFoods);  else if (saveDecision == 0)  goto mainMenu;  else  break;  } |
|  |
| #1-h | else  {  puts("Program is exiting now...\nProgram terminated successfully");  return 0;  }  }  else  puts ("\n!!!!Please select a valid number between 1 and 5\n");  }while (1);  return 0;  }  void addAnItem(struct groceryStore fDepartment[], char section[50], int index)  {  char Name[stock];  int Quantity;  float Cost;  printf("\n=>Enter name of the item for %s department: ", section);  scanf(" %[^\n]s", Name);  printf("Enter quantity of %s: ", Name);  scanf("%d", &Quantity);  printf("Enter cost of %s: ", Name);  scanf("%f", &Cost);  //storing values in the inventory  strcpy(fDepartment[index].itemName, Name);  fDepartment[index].itemQty = Quantity;  fDepartment[index].itemCost = Cost;  }  void changeQty(int \*ptrQty, char section[50], struct groceryStore fDepartment[])  {  int newQty = 0;  printf("\n=> Please enter the new quantity number: ");  scanf("%d", &newQty);  \*ptrQty = newQty;  printf("%5s -\t\t%25s\t\t%7s\t%7s\n", "Index", "Name", "Qty", "Cost\n");  for(int i = 0; (fDepartment[i].itemQty) != 0 && (i < 100); i++) //iterating through the entire inventory  {  printf("%5d -\t\t%25s\t\t%7d\t%7.2f", i+1, fDepartment[i].itemName, fDepartment[i].itemQty, fDepartment[i].itemCost);  puts("");  }  }  void changeCost(float \*ptrCost, char section[50], struct groceryStore fDepartment[])  {  float newCost = 0;    printf("\n=> Please enter the new cost amount: ");  scanf("%f", &newCost);  \*ptrCost = newCost;  printf("%5s -\t\t%25s\t\t%7s\t%7s\n", "Index", "Name", "Qty", "Cost\n");  for(int i = 0; (fDepartment[i].itemQty) != 0 && (i < 100); i++) //iterating through the entire inventory  {  printf("%5d -\t\t%25s\t\t%7d\t%7.2f", i+1, fDepartment[i].itemName, fDepartment[i].itemQty, fDepartment[i].itemCost);  puts("");  }  }  void outputInventory(struct groceryStore pInventory[], struct groceryStore dInventory[], struct groceryStore bInventory[], struct groceryStore fInventory[])  {  FILE \*file;  fopen("Inventory.txt", "w");  file = fopen("Inventory.txt", "a+");  fprintf(file, "%s\n", "File Created by HARNEET SINGH, 400110275, singhh76@mcmaster.ca");  fprintf(file, "\nGROCERY STORE - INVENTORY\n\n");    if (pInventory[0].itemQty != 0)  {  fprintf(file, "%s\n", "Produce Department Inventory:");  fprintf(file, "%5s -\t\t%25s\t\t%7s\t%7s\n", "Index", "Name", "Qty", "Cost\n");  for(int i = 0; (pInventory[i].itemQty) != 0 && (i < 100); i++) //iterating through the entire inventory  {  fprintf(file, "%5d -\t\t%25s\t\t%7d\t%7.2f\n", i+1, pInventory[i].itemName, pInventory[i].itemQty, pInventory[i].itemCost);  }  }  if (dInventory[0].itemQty != 0)  {  fprintf(file, "\n%s\n", "Deli Department Inventory:");  fprintf(file, "%5s -\t\t%25s\t\t%7s\t%7s\n", "Index", "Name", "Qty", "Cost\n");  for(int i = 0; (dInventory[i].itemQty) != 0 && (i < 100); i++) //iterating through the entire inventory  {  fprintf(file, "%5d -\t\t%25s\t\t%7d\t%7.2f\n", i+1, dInventory[i].itemName, dInventory[i].itemQty, dInventory[i].itemCost);  }  }  if (bInventory[0].itemQty != 0)  {  fprintf(file, "\n%s\n", "Bakery Department Inventory:");  fprintf(file, "%5s -\t\t%25s\t\t%7s\t%7s\n", "Index", "Name", "Qty", "Cost\n");  for(int i = 0; (bInventory[i].itemQty) != 0 && (i < 100); i++) //iterating through the entire inventory  {  fprintf(file,"%5d -\t\t%25s\t\t%7d\t%7.2f\n", i+1, bInventory[i].itemName, bInventory[i].itemQty, bInventory[i].itemCost);  }  }    if (fInventory[0].itemQty != 0)  {  fprintf(file, "\n%s\n", "Frozen-Foods Department Inventory:");  fprintf(file, "%5s -\t\t%25s\t\t%7s\t%7s\n", "Index", "Name", "Qty", "Cost\n");  for(int i = 0; (fInventory[i].itemQty) != 0 && (i < 100); i++) //iterating through the entire inventory  {  fprintf(file, "%5d -\t\t%25s\t\t%7d\t%7.2f\n", i+1, fInventory[i].itemName, fInventory[i].itemQty, fInventory[i].itemCost);  }  }  fclose(file);  printf("\n\*\*\*\*File successfully saved\*\*\*\*\n");  }  int printInventory(struct groceryStore fInventory[], char section[50], char attribute[20]) //print item details in the store  {  int count = 0;  printf("Following are the items listed in the %s department:\n\n", section);  printf("%5s -\t\t%25s\t\t%7s\t%7s\n", "Index", "Name", "Qty", "Cost\n");  for(int i = 0; (fInventory[i].itemQty) != 0 && (i < 100); i++) //iterating through the entire inventory  {  printf("%5d -\t\t%25s\t\t%7d\t%7.2f\n", i+1, fInventory[i].itemName, fInventory[i].itemQty, fInventory[i].itemCost);  count++;  }  int item\_index = -1;  if (count != 0)  {  puts("From the displayed list, please take note of the index number of the item which needs to be changed");  printf("=> Enter the exact index number here, to alter its %s: ", attribute);  scanf("%d", &item\_index);  }  else  {  printf("Sorry, there are zero entries in the %s department", section);  puts("\nPlease add items in this department first");  return -1;  }  //printf("\nAddress of the quantity: %p\n", &fInventory[item\_index-1].itemQty);  return (item\_index-1);  }  void printAllInventory(struct groceryStore pInventory[], struct groceryStore dInventory[], struct groceryStore bInventory[], struct groceryStore fInventory[])  {  puts("Following details will be printed in the file, please verify:");  printf("%5s -\t\t%25s\t\t%7s\t%7s\n", "Index", "Name", "Qty", "Cost\n");  if (pInventory[0].itemQty != 0)  puts("Produce Department:");  for(int i = 0; (pInventory[i].itemQty) != 0 && (i < 100); i++) //iterating through the entire inventory  {  printf("%5d -\t\t%25s\t\t%7d\t%7.2f\n", i+1, pInventory[i].itemName, pInventory[i].itemQty, pInventory[i].itemCost);  }  if (dInventory[0].itemQty != 0)  puts("Deli Department:");  for(int i = 0; (dInventory[i].itemQty) != 0 && (i < 100); i++) //iterating through the entire inventory  {  printf("%5d -\t\t%25s\t\t%7d\t%7.2f\n", i+1, dInventory[i].itemName, dInventory[i].itemQty, dInventory[i].itemCost);  }  if (bInventory[0].itemQty != 0)  puts("Bakery Department:");  for(int i = 0; (bInventory[i].itemQty) != 0 && (i < 100); i++) //iterating through the entire inventory  {  printf("%5d -\t\t%25s\t\t%7d\t%7.2f\n", i+1, bInventory[i].itemName, bInventory[i].itemQty, bInventory[i].itemCost);  }  if (fInventory[0].itemQty != 0)  puts("Frozen-Foods Department:");  for(int i = 0; (fInventory[i].itemQty) != 0 && (i < 100); i++) //iterating through the entire inventory  {  printf("%5d -\t\t%25s\t\t%7d\t%7.2f\n", i+1, fInventory[i].itemName, fInventory[i].itemQty, fInventory[i].itemCost);  }  } |
|  |

Until this point, only one executable file (a.out) was running, i.e. all the above-image shown images are of the same inventory file.

**Additional Examples:**

1. Following exercise shows that when no entries are registered in the struct then, the program notifies the use by printing “Sorry, there are zero entries …” statement.

User tries to modify cost value of an empty list:



1. User tries to change the quantity of an empty list, again same error message pops up on the screen to inform user about it:

