Design and implement a program to manage a Grocery Store Inventory using the following concepts in C programming:
Structures to represent items in the inventory.
Unions to store either the supplier's contact number or email.
Dynamic Memory Allocation to allow scalable storage for items in the inventory.
Typedef to simplify code readability.
Array of Structures to store and manipulate multiple inventory items.

Requirements:
Define Data Types:
Create a structure Item to hold the following fields:
itemID (integer): Unique ID for the item.
itemName (string): Name of the item.
price (float): Price per unit of the item.
quantity (integer): Quantity of the item in stock.
totalValue (float): Total value of the item in stock (calculated as price * quantity).
Create a union SupplierContact to store either:
phoneNumber (string): Contact number of the supplier.
email (string): Email address of the supplier.
Features:
Dynamic Memory Allocation:
Allocate memory dynamically for an array of Item structures based on user input (N items).
Input and Output:
Input the details of each item, including the supplier's contact details.
Calculate and store the totalValue for each item.
Display:
Display the details of all items in the inventory, including the total value for each item.
Search:
Search for an item by its ID and display its details.
Update:
Update the price or quantity of an item, and recalculate its totalValue.
Sorting:
Sort items based on their total value in descending order.
Typedef:
Use typedef to define aliases for the Item and SupplierContact structures.

Structure and Union Definitions:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

// Typedef for structures and unions
typedef struct {
  int itemID;
  char itemName[50];
  float price;
  int quantity;
  float totalValue; // price * quantity
} Item;

typedef union {
  char phoneNumber[15];
  char email[50];
} SupplierContact;
```

Program Requirements:
Menu Options:
Input item details.

Display all item details.
Search for an item by ID.
Update an item's price or quantity.
Sort items by total value in descending order.
Exit the program.
Sample Menu:
Menu:
1. Input Item Details
2. Display All Items
3. Search Item by ID
4. Update Item Price or Quantity
5. Sort Items by Total Value
6. Exit
***/**

CODE:

```
#include <stdio.h>
#include <stdlib.h>

typedef struct {
  int itemID;
  char itemName[50];
  float price;
  int quantity;
  float totalValue; // price * quantity
} Item;

typedef union {
  char phoneNo[15];
  char email[50];
} SupplierContact;

void addDetails(Item *items,SupplierContact *info,int *count);
void displayDetails(Item *items,SupplierContact *info,int count);
void serchID(Item *items,SupplierContact *info,int count,int ID);
void updateDetails(Item *items,SupplierContact *info,int count,int ID);
void Sorting(Item *items,SupplierContact *info,int count);
float findAvg(float *testResult);

int main(){
   int n;
   printf("Enter number of items:");
   scanf(" %d",&n);

   Item *items = (Item *)malloc(n * sizeof(Item));
   SupplierContact *info = (SupplierContact *)malloc(n * sizeof(SupplierContact));
   int op;
   int count =0;
   do{
      printf("\nChoose an option:\n1.Input details\n2.Display\n3.Search\n4.Update
details\n5.Sorting\n6.Exit\n");
      scanf("%d",&op);

      switch(op){
         case 1:
```

```
            addDetails(items,info,&count);
            break;

            case 2:
            displayDetails(items,info,count);
            break;

            case 3:
            int ID;
            printf("Enter the iD to be searched:");
            scanf("%d",&ID);
            serchID(items,info,count,ID);
            break;

            case 4:
            printf("Enter the iD to be searched:");
            scanf("%d",&ID);
            updateDetails(items,info,count,ID);
            break;

            case 5:
            Sorting(items,info,count);
            break;

            case 6:
            printf("Exiting programme....\n");
            break;

        }
    }while(op != 6);

}


void addDetails(Item *items,SupplierContact *info,int *count){
    printf("Enter the details of items  :\n");
    int chance=3,isUnique = 1,ID;
    while(chance >= 0){
        isUnique = 1;
        printf("\nEnter the unique item ID:");
        scanf("%d",&ID);
        for(int i=0;i<(* count);i++){
            if(items[i].itemID == ID){
            printf("Entered ID is already existing.\nTry another one!\n");
            printf("%d chances left!",chance);
            isUnique = 0;
            chance--;
            break;

            }
        }
        if(isUnique)break;
    }
    if(!isUnique){
        return;
    }
    items[*count].itemID = ID;
```

```
    printf("Enter the name:");
    scanf(" %[^\n]",items[*count].itemName);

    printf("Enter the price:");
    scanf(" %f",&items[*count].price);

    printf("Enter the quantity:");
    scanf("%d",&items[*count].quantity);

    items[*count].totalValue = (items[*count].price) * (items[*count].quantity);

    printf("Enter contact info:\n 1->Phone number\n2->Email Id\n");

    int cnt_op;
    scanf("%d",&cnt_op);

    if(cnt_op == 1){
        printf("Enter phone number: ");
        scanf(" %[^\n]",info[*count].phoneNo);
    }else if(cnt_op == 2){
        printf("Enter the email ID: ");
        scanf(" %[^\n]",info[*count].email);
    }
    (*count)++;
}

void displayDetails(Item *items,SupplierContact *info,int count){
    for(int i=0;i<count;i++){
        printf("\nItem %d details:",i+1);
        printf("\nItem ID:%d",items[i].itemID);
        printf("\nItem name : %s",items[i].itemName);
        printf("\nItem price : %.2f $",items[i].price);
        printf("\nItem quantity : %d",items[i].quantity);
        printf("\nTotal vale : %.2f $",items[i].totalValue);
        printf("\nContact Info:\n");
        if(info[i].phoneNo != NULL)
        {
            printf("\nPhone NO: %s",info[i].phoneNo);
        }
        else
        {
            printf("\nEmail ID: %s",info[i].email);
        }
    }
}

void serchID(Item *items,SupplierContact *info,int count,int ID){
    for(int i=0;i<count;i++){
        if(ID == items[i].itemID){
        printf("%d ID found\n",ID);
        printf("\nItem %d details:",i+1);
        printf("\nItem ID:%d",items[i].itemID);
        printf("\nItem name : %s",items[i].itemName);
        printf("\nItem price : %.2f $",items[i].price);
        printf("\nItem quantity : %d",items[i].quantity);
        printf("\nTotal vale : %.2f $",items[i].totalValue);
```

```
        printf("\nContact Info:\n");
        if(info[i].phoneNo != NULL)
        {
           printf("\nPhone NO: %s",info[i].phoneNo);
        }
        else
        {
           printf("\nEmail ID: %s",info[i].email);
        }
           return;
        }
     }

        printf("%d ID not found\n",ID);
        return;

}

void updateDetails(Item *items,SupplierContact *info,int count,int ID){
   for(int i=0;i<count;i++){
      if(ID == items[i].itemID){
         printf("Enter the option to be updated:\n1->To update price\n2->To update quantity\n");
         int op;
         scanf("%d",&op);
         if(op == 1){
            printf("Enter price to be updated:");
            float Price;
            scanf("%f",&Price);
            items[i].price = Price;

         }else if(op == 2){
            printf("Enter the quantity:");
            int quant;
            scanf("%d",&quant);
            items[i].quantity = quant;
         }
         items[i].totalValue = (items[i].price) * (items[i].quantity);
         printf("Updated Successfully...\n");
         return;
      }

   }printf("%d ID not found\n",ID);
   return;
}

void Sorting(Item *items,SupplierContact *info,int count){
   for(int i=0;i<count-1;i++){
      for(int j=0;j<count-i-1;j++){
         if(items[j].totalValue < items[j+1].totalValue){
            Item tempitem = items[j];
            items[j] = items[j+1];
            items[j+1] = tempitem;

            SupplierContact tempInfo = info[j];
            info[j] = info[j+1];
            info[j+1] = tempInfo;
         }
```

```
        }
    }
    displayDetails(items,info,count);
    return;

}
```