C-Lab Project(B.tech CSE)

Name: KALASH KANDARI

SAP ID:590026638

Batch: 28

Course: B.tech CSE

Project Name:

Shopping List Project

Project Details/Overview

Shopping List Management System:

This project is a basic command-line application in C designed to manage a list of items and their quantities.

C shop.c > main()

```
-: WELCOME TO THE SHOPPING LIST :-

Enter 1 to add item in shopping list
Enter 2 to remove item from shopping list
Enter 3 to edit item of shopping list
Enter 4 to show shopping list
Enter 5 to exit
```

Core Functionality:

- **Add Item:** Allows the user to input an item name and its quantity and adds it to the list.

```
void addItem(void){
    char ch;
    repeat:
    printf("\nEnter item name : ");
    fflush(stdin);
    gets(itemName[count]);
    printf("Enter item quantity : ");
    scanf("%d",&itemQuantity[count]);
    count++;
    printf("\nPress 'Y' to add more items otherwise press 'N' : ");
    ch = getchar();
    if (ch=='y'||ch=='Y')
    {
        goto repeat;
    }
```

int count

- **Remove Item:** Allows the user to search for an item by name and delete it from the list.

```c
void editItem(void){
    char name[50];
    int quantity,check = 0;
    printf("Enter item name : ");
    fflush(stdin);
    gets(name);
    printf("Enter new quantity : ");
    scanf("%d",&quantity);
    for (int i = 0; i < count; i++)
    {
        if (strcmp(name,itemName[i]) == 0)
        {
            itemQuantity[i] = quantity;
            printf("\nItem quantity is updated\n");
            check = 1;
        }
    }
    if (check==0)
    {
        printf("\nItem name is not find\n");
    }
}
```

```c
void removeItem(void){
    char name[50];
    int check = 0;
    printf("Enter item name : ");
    fflush(stdin);
    gets(name);
    for (int i = 0; i < count; i++)
    {
        if (strcmp(name,itemName[i]) == 0)
        {
            for (int j = i; j < count; j++)
            {
                strcpy(itemName[j],itemName[j+1]);
                itemQuantity[j] = itemQuantity[j+1];
            }
            count--;
            printf("\nItem has been deleted\n");
            return;
            check = 1;
        }
    }
```

- **Edit Item:** Allows the user to search for an item by name and update its quantity.
- **Show List:** Displays all current items and their quantities.

```
void showShoppingList(){
    if (count == 0)
    {
        printf("\nShopping list is empty\n");
        return;
    }
    printf("\n Item name\tItem Quantity\n\n");
    for (int i = 0; i < count; i++)
    {
        printf("%s\t\t%d\n",itemName[i],itemQuantity[i]);
    }
}
```

- **Exit:** Terminates the program.

C Types Used:

Typ	Example	Description
cha	char itemName[20][5	Used for individual characters (like user input choice 'y'/'n') and to form **strings** (item names).
int	int itemQuantity[20	Used for storing **integer values**, specifically item quantities, loop counters (i, j), the total item count (count), user menu choice (choice), and status/flag variables (check).
void	void addItem(vo	Used as the return type for functions that **do not return a value** and for function arguments when a function **takes no arguments**.

Functions and Their Roles:

Function Nan	Returr Type	Argume	Description
addItem	void	void	Prompts the user for an item name and quantity, then **appends** the item to the global lists. Uses a goto statement to allow adding multiple items sequentially.

removeItem	void	void	Prompts for an item name, searches the list, and **removes** the item by shifting all subsequent elements up by one index. It decrements the global item count.

Function Name	**Returr Type**	**Argume**	**Description**
editItem	void	void	Prompts for an item name and a new quantity, then **updates** the quantity for the matching item in the list.
showShoppingList	void	void	**Displays** the contents of the item name and quantity lists in a formatted table, unless the list is empty.
main	int	*None*	The **entry point** of the program. It displays the main menu and uses a do-while loop and a switch statement to handle user interaction and call other functions until the user chooses to exit (option 5).

```
Enter your choice: 1

Enter item name : milk 500ml
Enter item quantity : 2

Press 'Y' to add more items otherwise press 'N' : y

Enter item name : bread
Enter item quantity : 5
```

```
Enter 1 to add item in shopping list
Enter 2 to remove item from shopping list
Enter 3 to edit item of shopping list
Enter 4 to show shopping list
Enter 5 to exit
Enter your choice: 2
Enter item name : milk 500ml

Item has been deleted

Enter 1 to add item in shopping list
Enter 2 to remove item from shopping list
Enter 3 to edit item of shopping list
Enter 4 to show shopping list
Enter 5 to exit
```

```
  Item name      Item Quantity

 milk 500ml              2
 egg             12
 bread           2
```

Global Variables and Arrays:

These variables are declared outside all functions and are accessible and modifiable by all functions in the program.

- **char itemName[20][50]**: A **2D character array** (array of strings) to store up to **20 item names**, each with a maximum length of 49 characters (plus the null terminator).
- **int itemQuantity[20]**: An **integer array** to store the quantity for each corresponding item, holding up to **20 integer values**.

- **int count = 0**: An **integer variable** that tracks the **current number of items** in the shopping list. It acts as the next available index for adding items and the upper bound for loops when iterating through the list.

```
char itemName[20][50];
int itemQuantity[20];
int count = 0;
```

Key Standard Library Functions Used:

The program uses functions from the <stdio.h> and <string.h> libraries:

- **<stdio.h> (Standard I/O):**
    - printf(): Output to the console. ○ scanf(): Read formatted input from the console (used for int quantity and choice).
    - gets(): Read a line of text/string from the console (used for item names). **Note:** This function is generally considered **unsafe** and deprecated; fgets() is preferred in modern C.
    - getchar(): Read a single character from the console.
    - fflush(stdin): Attempts to clear the input buffer. **Note:** Its behavior with stdin is **undefined** according to the C standard; it's often used in older/non-standard compilers to handle buffer issues after mixing scanf and gets/getchar.
- **<string.h> (String Handling):**
    - **strcmp(str1, str2)**: Compares two strings. Returns **0** if they are identical, which is used to find a matching item name.
    - **strcpy(dest, src)**: Copies the string from the source (src) to the destination (dest). Used in removeItem to shift item names and in editItem to update the name.

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

Then to print the whole code in the Shop.txt File:

We use the command **FILE** with using a pointer

```
FILE *fptr;
fptr = fopen("Shop.txt", "w");
fprintf(fptr,"\n Item name\tItem Quantity\n\n");
for (int i = 0; i < count; i++)
{
    fprintf(fptr, "%s\t\t%d\n",itemName[i],itemQuantity[i]);
}
return 0;
```

```c
#include <stdio.h>
#include <string.h>
char itemName[20][50];
int itemQuantity[20];
int count = 0;
void addItem(void){
    char ch;
    repeat:
    printf("\nEnter item name : ");
    fflush(stdin);
    gets(itemName[count]);
    printf("Enter item quantity : ");
    scanf("%d",&itemQuantity[count]);
    count++;
    printf("\nPress 'Y' to add more items otherwise press 'N' : ");
    ch = getchar();
    if (ch=='y'||ch=='Y')
    {
        goto repeat;
    }
}
void removeItem(void){
    char name[50];
    int check = 0;
    printf("Enter item name : ");
    fflush(stdin);
    gets(name);
    for (int i = 0; i < count; i++)
    {
        if (strcmp(name,itemName[i]) == 0)
        {
            for (int j = i; j < count; j++)
            {
                strcpy(itemName[j],itemName[j+1]);
                itemQuantity[j] = itemQuantity[j+1];
            }
            count--;
            printf("\nItem has been deleted\n");
            return;
            check = 1;
        }
    }
    if (check==0)
    {
```

```
        printf("\nItem name is not find\n");
    }
}
void editItem(void){
    char name[50];
    int quantity,check = 0;
    printf("Enter item name : ");
    fflush(stdin);
    gets(name);
    printf("Enter new quantity : ");
    scanf("%d",&quantity);
    for (int i = 0; i < count; i++)
    {
        if (strcmp(name,itemName[i]) == 0)
        {
            itemQuantity[i] = quantity;
            printf("\nItem quantity is updated\n");
            check = 1;
        }
    }
    if (check==0)
    {
        printf("\nItem name is not find\n");
    }
}
void showShoppingList(){
    if (count == 0)
    {
        printf("\nShopping list is empty\n");
        return;
    }
    printf("\n Item name\tItem Quantity\n\n");
    for (int i = 0; i < count; i++)
    {
        printf("%s\t\t%d\n",itemName[i],itemQuantity[i]);
    }
}
int main()
{
    int choice;
    printf("\n-: WELCOME TO THE SHOPPING LIST :-\n");
```

```c
do
{
    printf("\nEnter 1 to add item in shopping list\n");
    printf("Enter 2 to remove item from shopping list\n");
    printf("Enter 3 to edit item of shopping list\n");
    printf("Enter 4 to show shopping list\n");
    printf("Enter 5 to exit\n");
repeat:
    printf("Enter your choice: ");
    fflush(stdin);
    scanf("%d", &choice);
    switch (choice)
    {
    case 1:
        addItem();
        break;
    case 2:
        removeItem();
        break;
    case 3:
        editItem();
        break;
    case 4:
        showShoppingList();
        break;
    case 5:
        break;
    default:
        printf("\nInvalid choice try again\n");
        goto repeat;
    }
} while (choice != 5);
return 0;
```

```c
FILE *fptr;
fptr = fopen("Shop.txt", "w");
fprintf(fptr,"\n Item name\tItem Quantity\n\n");
for (int i = 0; i < count; i++)
{
    fprintf(fptr, "%s\t\t%d\n",itemName[i],itemQuantity[i]);
}
return 0;
```