CANTEEN MANAGEMENT SYSTEM

DATA STRUCTURE USED: Linked List
SOURCE CODE:

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

struct node
{
   char foodname[50];
   int quantity;
   float price;
   int data;
   struct node *prev;
   struct node *next;
};

struct node *headc = NULL,*newnode,*tailc = NULL;
struct node *heada = NULL, *taila = NULL;
struct node *head_s;

void adminmenu()
{
   printf("1. View total sales\n");
   printf("2. Add new items in the order menu\n");
   printf("3. Delete items from the order menu\n");
   printf("4. Display order menu\n");
   printf("5. Back to Menu\n");
   printf("Enter Your Choice");
}

void customermenu()
{
   printf("1. Place your order\n");
   printf("2. View your ordered items\n");
   printf("3. Delete an item from order\n");
   printf("4. Display final bill\n");
   printf("5. Back to menu \n");
   printf("Enter Your Choice: ");
}

struct node* createadmin(struct node *head,int data, char foodname[25], float price)
{
   newnode = (struct node*)malloc(sizeof(struct node));

   newnode->data = data;
   newnode->price = price;
   newnode-> quantity = 0;
   strcpy(newnode->foodname,foodname);
   newnode->next = NULL;
```

```
    newnode->prev = NULL;

    struct node *temp = head;

    if(temp==NULL)
       heada = taila = newnode;
    else
    {
       while(temp->next!=NULL)
          temp=temp->next;

       temp->next=newnode;
       newnode->prev = taila;
       taila = newnode;
    }

    return heada;
}

struct node* createcustomer(struct node *head,int data,int quantity)
{
    newnode = (struct node*)malloc(sizeof(struct node));

    struct node *temp1 = heada;
    int flag = 0;
    while(temp1!=NULL)
    {
       if(temp1->data==data)
       {
          flag = 1;
          break;
       }
       temp1 = temp1->next;
    }

    if(flag==1)
    {
       newnode->data = data;
       newnode->price = quantity*(temp1->price);
       newnode-> quantity = quantity;
       strcpy(newnode->foodname,temp1->foodname);
       newnode->next = NULL;
       newnode->prev = NULL;

       struct node *temp = head;

       if(temp==NULL)
          headc = tailc = newnode;
       else
       {
          while(temp->next!=NULL)
```

```
            temp=temp->next;

          temp->next=newnode;
          newnode->prev = tailc;
          tailc = newnode;
       }

   }
   else
   {
      printf("\n This item is not present in the menu! \n");
   }
   return headc;
}
```

//displaying the order menu for admin

```
void displayList(struct node *head)
{
   struct node *temp1 = head;
   if(temp1==NULL)
   {
      printf("\nList is empty!!\n\n");
   }
   else
   {
      printf("\n");
      while(temp1!=NULL)
      {
         if(temp1->quantity==0)
            printf("%d\t%s\t%0.2f\n",temp1->data,temp1->foodname,temp1->price);
         else
         {
            printf("%d\t%s\t%d\t%0.2f\n",temp1->data,temp1->foodname,temp1->quantity,temp1-
>price);
         }

         temp1 = temp1->next;
      }
      printf("\n");
   }

}
```

// for total sales and for traversal of linked list we use structure as a function.

```
struct node* totalsales(int data,int quantity)
{
   newnode = (struct node*)malloc(sizeof(struct node));
   int flag = 0;

   struct node *temp1 = heada;
```

```
    while(temp1->data!=data)
    {
       temp1 = temp1->next;
    }

    newnode->data = data;
    newnode->price = quantity*(temp1->price);
    newnode-> quantity = quantity;
    strcpy(newnode->foodname,temp1->foodname);
    newnode->next = NULL;
    newnode->prev = NULL;

    struct node *temp = head_s;

    if(temp==NULL)
       head_s = newnode;
    else
    {
       while(temp->next!=NULL)
       {
          if(temp->data==data)
          {
             flag = 1;
             break;
          }
          temp=temp->next;
       }

       if(flag==1)
       {
          temp->quantity += newnode-> quantity;
          temp->price += newnode->price;
       }
       else
       {
          temp->next=newnode;
       }
    }

    return head_s;
}

void calculatetotsales()
{
    struct node *temp = headc;
    while(temp!=NULL)
    {
       head_s = totalsales(temp->data, temp->quantity);
       temp=temp->next;
    }
}
```

```
struct node* delete(int data,struct node *head, struct node* tail)
{
   if(head==NULL)
   {
      printf("\nList is empty\n");
   }
   else
   {
      struct node* temp;
      if(data==head->data)
      {
         temp = head;
         head = head->next;
         if (head != NULL)
            head->prev = NULL;
         free(temp);
      }
      else if(data==tail->data)
      {
         temp = tail;
         tail = tail->prev;
         tail->next = NULL;
         free(temp);
      }
      else
      {
         temp = head;
         while(data!=temp->data)
         {
            temp = temp->next;
         }
         (temp->prev)->next = temp->next;
         (temp->next)->prev = temp->prev;
         free(temp);
      }
   }
   return head;
}
// delete an item in the order menu

int deleteadmin()
{
   printf("\nEnter serial no. of the food item which is to be deleted: ");
   int num;
   scanf("%d",&num);

   struct node* temp=heada;
   while(temp!=NULL)
   {
      if (temp->data == num)
```

```
        {
            heada = delete(num, heada, taila);
            return 1;
        }
        temp=temp->next;
    }

    return 0;
}
// delete item for customers which they have ordered.

int deletecustomer()
{
    printf("\nEnter serial no. of the food item which is to be deleted: ");
    int num;
    scanf("%d",&num);

    struct node* temp=headc;
    while(temp!=NULL)
    {
        if (temp->data == num)
        {
            headc = delete(num, headc, tailc);
            return 1;
        }
        temp=temp->next;
    }

    return 0;
}
// displaying bill for the customer

void displaybill()
{
    displayList(headc);
    struct node *temp = headc;
    float total_price = 0;
    while (temp!=NULL)
    {
        total_price +=temp->price;
        temp = temp->next;
    }

    printf("Total price: %0.02f\n",total_price);

}

struct node* deleteList(struct node* head)
{
    if(head==NULL)
    {
```

```
        return NULL;
    }
    else
    {
        struct node* temp = head;
        while(temp->next!=0)
        {
            temp = temp->next;
            free(temp->prev);
        }
        free(temp);
        head = NULL;
    }

    return head;
}

void admin()
{
    printf("\n");
    printf(" ADMIN SECTION\n");
    while(1)
    {
        adminmenu();

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

        if(opt==5)
            break;

        switch (opt)
        {
            case 1:
                displayList(head_s);
                break;
            case 2:

                printf("\nEnter serial no. of the food item: ");
                int num,flag = 0;
                char name[50];
                float price;
                scanf("%d",&num);

                struct node *temp = heada;

                while(temp!=NULL)
                {
                    if(temp->data==num)
                    {
                        printf("\nFood item with given serial number already exists!!\n\n");
```

```
                flag = 1;
                break;
            }
            temp = temp->next;
        }

        if(flag==1)
            break;

        printf("Enter food item name: ");
        scanf("%s",name);
        printf("Enter price: ");
        scanf("%f",&price);
        heada = createadmin(heada, num, name, price);
        printf("\nNew food item added to the list!!\n\n");
        break;
    case 3:
        if(deleteadmin())
        {
            printf("\n ###Updated list of food items menu ###\n");
            displayList(heada);
        }
        else
            printf("\nFood item with given serial number doesn't exist!\n\n");

        break;
    case 4:
        printf("\n ### Order menu ###\n");
        displayList(heada);
        break;
    case 5:
        mainmenu();

    default:
        printf("\nEnter a valid option\n");
        break;
    }
  }
}

void customer()
{
  int flag=0,j=1;
  char ch;
  printf("\n");
  printf("\CUSTOMER SECTION\n");
  while(1)
  {
    customermenu();

    int opt;
```

```
        scanf("%d",&opt);

        if(opt==5)
           break;

        switch (opt)
        {
           case 1:
              displayList(heada);
              printf("Enter number corresponding to the item you want to order: ");
              int n;
              scanf("%d",&n);
              printf("Enter quantity: ");
              int quantity;
              scanf("%d",&quantity);
              headc = createcustomer(headc, n, quantity);
              break;
           case 2:
              printf("\n ### List of ordered items ###\n");
              displayList(headc);
              break;
           case 3:
              if(deletecustomer())
              {
                 printf("\n ### Updated list of your ordered food items ###\n");
                 displayList(headc);
              }
              else
                 printf("Food item with given serial number doesn't exist!!\n");
              break;
           case 4:
              calculatetotsales();
              printf("\n### Final Bill ###\n");
              displaybill();
              headc = deleteList(headc);
              //flag=1;
              break;

           case 5:
              mainmenu();
              break;

           default:
              printf("Enter a  valid option\n");
              break;
        }
        if(flag==1)
           break;
     }
  }
```

```
void mainmenu()
{
   printf("\n");
   printf("MIT CANTEEN MANAGEMENT SYSTEM \n");
   printf("CREATED BY: PRIYA\t REG.NO:2021506047 \n");
   printf("1. ADMIN \n");
   printf("2. CUSTOMER \n");
   printf("3. EXIT \n\n");
   printf("Enter Your Choice");
}

int main()
{
   heada = createadmin(heada,1,"Chicken roll",18);
   heada = createadmin(heada,2,"Egg puffs",13);
   heada = createadmin(heada,3,"Veg puffs",12);
   heada = createadmin(heada,4,"Tea",5);
   heada = createadmin(heada,5,"Coffee",10);

   while(1)
   {
      mainmenu();
      int choice;
      scanf("%d",&choice);

      if(choice==3)
      {
         printf("\nThank you!!");
         break;
      }

      switch (choice)
      {
         case 1:
            admin();
            break;
         case 2:
            customer();
            break;
         case 3:
            break;

         default:
            printf("Enter a valid option\n");
            break;
      }
   }
}
```

OUTPUT:

Choice 1

```
MIT CANTEEN MANAGEMENT SYSTEM
CREATED BY: PRIYA        REG.NO:2021506047
1. ADMIN
2. CUSTOMER
3. EXIT

Enter Your Choice1
```

```
 ADMIN SECTION
1. View total sales
2. Add new items in the order menu
3. Delete items from the order menu
4. Display order menu
5. Back to Menu
Enter Your Choice
```

Choice 2

```
CUSTOMER SECTION
1. Place your order
2. View your ordered items
3. Delete an item from order
4. Display final bill
5. Back to menu
Enter Your Choice:
```

```
Enter Your Choice: 1

1       Chicken roll    18.00
2       Egg puffs       13.00
3       Veg puffs       12.00
4       Tea     5.00
5       Coffee  10.00
```

Choice 3

```
3. EXIT

Enter Your Choice3

Thank you!!
```