

Assignment 6

CODE:-

```
#include<stdio.h>

#include<stdlib.h>

#include<string.h>


struct stuff
{
    char product[20];
    int price;
    int id;
    int quantity;
    struct stuff *link;
};


struct stuff *create(struct stuff *start);
void display (struct stuff *start);
struct stuff *insert(struct stuff *start);
struct stuff *del(struct stuff *start,char eng_stuff[]);
struct stuff *sort (struct stuff *start);
void *search(struct stuff *start,char eng_stuff[]);
struct stuff *modify(struct stuff *start,char eng_stuff[]);
int i,n;
void main()
{   char eng_stuff[20];
    int ch;
    struct stuff *start=NULL;
    while(1)
    {
        printf("Enter choice-\n 1) to create\n 2) to display\n 3) to insert\n 4) to delete\n 5) to sort by
price\n 6) to search\n 7) to modify:");
```

```
scanf("%d",&ch);
switch (ch)
{
case 1:
    start=create(start);
    break;
case 2:
    display(start);
    break;
case 3:
    start=insert(start);
    break;
case 4:
    printf("Enter eng_stuff you want to delete:");
    scanf("%s",eng_stuff);
    start=del(start,eng_stuff);
    break;
case 5:
    sort(start);
    break;
case 6:
    printf("Enter eng_stuff you want to search:");
    scanf("%s",eng_stuff);
    search(start,eng_stuff);
    break;
case 7:
    printf("Enter eng_stuff you want to modify:");
    scanf("%s",eng_stuff);
    start=modify(start,eng_stuff);
    break;
default:
```

```

        printf("error!");

    }

}}

struct stuff *create(struct stuff* start)
{
    struct stuff *p,*temp;

    printf("Enter no of stuffs:");
    scanf("%d",&n);

    temp=start;
    for(i=0;i<n;i++)
    { temp=(struct stuff*)malloc(sizeof(struct stuff)*1);
        printf("enter stuff name:");
        scanf("%s",(temp->product));
        printf("enter price:");
        scanf("%d",&(temp->price));
        printf("enter the quantity:");
        scanf("%d",&temp->quantity);
        printf("enter the id:");
        scanf("%d",&temp->id);

    }

    if(start==NULL)
    {
        start=temp;
        temp->link=NULL;

    }
}

```

```

else
{
    p=start;
    while(p->link!=NULL)
    p=p->link;
    p->link=temp;
    temp->link=NULL;
}
}
return start;
}

```

```

void display (struct stuff *start)

```

```

{ struct stuff *p;

```

```

    if(start==NULL)

```

```

    {

```

```

        printf("\n*Order List is Empty*\n\n");

```

```

        return;

```

```

    }

```

```

    p=start;

```

```

    printf("\neng_stuff\tPrice\tquantity\tid");

```

```

    while(p!=NULL)

```

```

    {

```

```

        printf("\n%s\t%d\t%d\t\t%d\n",p->product,p->price,p->quantity,p->id);

```

```

        p=p->link;

```

```

    }

```

```

    printf("\n");

```

```
}
```

```
struct stuff *insert(struct stuff *start)
{ int pos;
  struct stuff *p,*temp;
  temp=(struct stuff*)malloc(sizeof(struct stuff)*1);
  printf("Enter location you want to insert:");
  scanf("%d",&pos);
```

```

  if(pos==1)
  { printf("enter stuff name:");
    scanf("%s",(temp->product));
    printf("enter price:");
    scanf("%d",&(temp->price));
    printf("enter the id");
    scanf("%d",&(temp->id));
    printf("enter the quantity");
    scanf("%d",&(temp->quantity));
    temp->link=start;
    start=temp;
```

```

    return start;
```

```
}
```

```

  p=start;
  for(i=1;i<pos-1 && p!=NULL;i++)
    p=p->link;
  if(p==NULL)
    printf("\n*less no of eng_stuffs*\n\n");
```

```

  else
```

```
  { printf("enter stuff name:");
```

```

scanf("%s",(temp->product));
printf("enter price:");
scanf("%d",&(temp->price));
printf("enter the id");
scanf("%d",&(temp->id));
printf("%enter the quantity");
scanf("%d",&(temp->quantity));
temp->link=p->link;
p->link=temp;
}
return start;
}

struct stuff *del(struct stuff *start,char eng_stuff[])
{
    struct stuff *temp,*p;

    if(strcmp(start->product,eng_stuff)==0)
    {
        temp=start;
        start=temp->link;
        free(temp);
        return start;
    }

    p=start;
    while(p->link!=NULL)
    { if(strcmp(p->link->product,eng_stuff)==0)
        {
            temp=p->link;
            p->link=temp->link;
            free(temp);

```

```

        return start;
    }
    p=p->link;
}
}

struct stuff *sort (struct stuff *start)
{
    struct stuff *p,*q,*temp;
    temp=(struct stuff*)malloc(sizeof(struct stuff)*1);

    for(p=start;p->link!=NULL;p=p->link)
    {
        for(q=p->link;q!=NULL;q=q->link)
        {

            if(p->price > q->price)
            {
                temp->price=p->price;
                p->price=q->price;
                q->price=temp->price;
                strcpy(temp->product,p->product);
                strcpy(p->product,q->product);
                strcpy(q->product,temp->product);

            }
        }
    }
}

void *search(struct stuff *start,char eng_stuff[])
{

```

```
struct stuff *p;
```

```
int flag=0;
```

```
if(strcmp(start->product,eng_stuff)==0)
```

```
{
```

```
printf("\neng_stuff FOUND\n");
```

```
printf("\neng_stuff\t\tPrice\t\tquantity\tid");
```

```
printf("\n%s\t %d\t%d\t%d\n",start->product,start->price,start->quantity,start->id);
```

```
flag++;
```

```
}
```

```
p=start;
```

```
while(p->link!=NULL)
```

```
{
```

```
if(strcmp(p->link->product,eng_stuff)==0)
```

```
{
```

```
printf("\n*eng_stuff FOUND\n");
```

```
printf("\neng_stuff\t\tPrice\t\tquantity\tid");
```

```
printf("\n%s\t %d\t%d\t%d\n",p->link->product,p->link->price,p->link->id,p->link->quantity);
```

```
}
```

```
p=p->link;
```

```
flag++;
```

```
}
```

```
if(flag==0)
```

```
printf("\n\nnot found\n\n");
```

```
}
```

```
struct stuff *modify(struct stuff *start,char eng_stuff[])
```

```
{
```



```
    struct stuff *p;
int choice;
char ans1,ans2;
int flag=0,chw=0;
```

```
if(strcmp(start->product,eng_stuff)==0)
{
    printf("\neng_stuff FOUND\n");
    printf("\neng_stuff\tPrice\tid\tquantity");
    printf("\n%s\t %d\t%d\t%d",start->product,start->price,start->id,start->quantity);
```

```
do
{
```

```
    printf("enter:\n 1) to modify Product name\n 2) to modify Product price\n 3) to modify id\n 4) to modify quantity\n");
```

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

```
    switch (choice)
```

```
{
```

```
case 1:
```

```
    printf("enter new product name:");
```

```
    scanf("%s",(start->product));
```

```
    break;
```

```
case 2:
```

```
    printf("enter new product price:");
```

```
    scanf("%d",&(start->price));
```

```
    break;
```

```
case 3:
```

```
printf("enter id of candidate:");
```

```
scanf("%d",&(start->id));
```

```
case 4:
```

```
printf("enter candidate quantity:");
```

```
scanf("%d",&(start->quantity));
```

```
}
```

```
printf("press 1 to continue modifying:");
```

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

```
}while(chw==1);
```

```
    flag++;
```

```
    return start;
```

```
}
```

```
p=start;
```

```
while(p->link!=NULL)
```

```
{
```

```
    if(strcmp(p->link->product,eng_stuff)==0)
```

```
{
```

```
    printf("\n*eng_stuff FOUND\n");
```

```
    printf("\neng_stuff\tPrice\nid\nquantity");
```

```
    printf("\n%s\t %d\n%d\n%d",p->link->product,p->link->price,p->link->id,p->link->quantity);
```

```
do
```

```
{
```

```
    printf("enter:\n 1) to modify Product name\n 2) to modify Product price:");
```

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

```
switch (choice)
```

```
{
```

```

    case 1:
        printf("enter new product name:");
        scanf("%s", (p->link->product));
        break;
    case 2:
        printf("enter new product price:");
        scanf("%d", &(p->link->price));
        break;
    case 3:
        printf("enter stuff id:");
        scanf("%d", &(start->link->id));
        break;
    case 4:
        printf("enter new product name:");
        scanf("%d", &(start->link->quantity));
        break;

    }printf("press 1 to continue modifying:");
        scanf("%d", &chw);
        }while(chw==1);

    }
    p=p->link;
    flag++;
    return start;
}

if(flag==0)
    printf("\n\nnot found\n\n");
}

```

OUTPUT:-

Enter choice-

- 1) to create
- 2) to display
- 3) to insert
- 4) to delete
- 5) to sort by price
- 6) to search
- 7) to modify:1

Enter no of stuffs:2

enter stuff name:computer

enter price:20000

enter the quantity:23

enter the id:1234

enter stuff name:keyboard

enter price:12000

enter the quantity:12

enter the id:1231

Enter choice-

- 1) to create
- 2) to display
- 3) to insert
- 4) to delete
- 5) to sort by price
- 6) to search
- 7) to modify:2

eng_stuff	Price	quantity	id
computer	20000	23	1234

keyboard	12000	12	1231
----------	-------	----	------

Enter choice-

- 1) to create
- 2) to display
- 3) to insert
- 4) to delete
- 5) to sort by price
- 6) to search
- 7) to modify:5

Enter choice-

- 1) to create
- 2) to display
- 3) to insert
- 4) to delete
- 5) to sort by price
- 6) to search
- 7) to modify:2

eng_stuff	Price	quantity	id
keyboard	12000	23	1234

computer	20000	12	1231
----------	-------	----	------

Enter choice-

- 1) to create
- 2) to display
- 3) to insert
- 4) to delete
- 5) to sort by price

6) to search

7) to modify:3

Enter location you want to insert:2

enter stuff name:motherboard

enter price:21000

enter the id1233

1.662290e+183nter the quantity21

Enter choice-

1) to create

2) to display

3) to insert

4) to delete

5) to sort by price

6) to search

7) to modify:2

eng_stuff	Price	quantity	id
keyboard	12000	23	1234
motherboard	21000	21	1233
computer	20000	12	1231

Enter choice-

1) to create

2) to display

3) to insert

4) to delete

5) to sort by price

6) to search

7) to modify:4

Enter eng_stuff you want to delete:motherboard

Enter choice-

- 1) to create
- 2) to display
- 3) to insert
- 4) to delete
- 5) to sort by price
- 6) to search
- 7) to modify:2

eng_stuff	Price	quantity	id
keyboard	12000	23	1234
computer	20000	12	1231

Enter choice-

- 1) to create
- 2) to display
- 3) to insert
- 4) to delete
- 5) to sort by price
- 6) to search
- 7) to modify:6

Enter eng_stuff you want to search:computer

*eng_stuff FOUND

eng_stuff	Price	quantity	id
computer	20000	1231	12

Enter choice-

- 1) to create

- 2) to display
- 3) to insert
- 4) to delete
- 5) to sort by price
- 6) to search
- 7) to modify:7

Enter eng_stuff you want to modify:keyboard

eng_stuff FOUND

eng_stuff	Price	id	quantity
keyboard	12000	1234	23

enter:

- 1) to modify Product name
- 2) to modify Product price
- 3)to modify id
- 4)to modify quantity

2

enter new product price:23000

press 1 to continue modifying:1

enter:

- 1) to modify Product name
- 2) to modify Product price
- 3)to modify id
- 4)to modify quantity

1

enter new product name:laptop

press 1 to continue modifying:0

Enter choice-

- 1) to create
- 2) to display
- 3) to insert

- 4) to delete
- 5) to sort by price
- 6) to search
- 7) to modify:2

eng_stuff	Price	quantity	id
laptop	23000	23	1234

computer	20000	12	1231
----------	-------	----	------

Enter choice-

- 1) to create
- 2) to display
- 3) to insert
- 4) to delete
- 5) to sort by price
- 6) to search
- 7) to modify:0

error!Enter choice-

- 1) to create
- 2) to display
- 3) to insert
- 4) to delete
- 5) to sort by price
- 6) to search
- 7) to modify: