```c
#include<stdio.h>
#include<math.h>


struct node
{
    int data;
    struct node *next;
}*front,*rear,*temp;

void insert()
{
     int n;
      struct node *newnode;
    printf("Enter the data\n");
    scanf("%d",&n);
    newnode=(struct node*)malloc(sizeof(struct node));
    newnode->data=n;
    newnode->next=NULL;
    if(front==0&&rear==0)
    {
        front=rear=newnode;
    }
    else{
        rear->next=newnode;
        rear=newnode;
    }
}

void display()
{

    temp=front;
    while(temp!=NULL)
```

```
void display()
{

    temp=front;
    while(temp!=NULL)
    {
        printf("%d",temp->data);
        temp=temp->next;
    }
}


void delete()
{
    temp=front;
    front=temp->next;
    free(temp);
}


void main()
{
     int choice;

    while (1)
    {
        printf("\n1.insert\n");
        printf("2. display\n");
        printf("3. delete\n");
        printf("4. exit\n");

        printf("Enter your choice: ");
        scanf("%d", &choice);
```

```
    {
        printf("\n1.insert\n");
        printf("2. display\n");
        printf("3. delete\n");
        printf("4. exit\n");

        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice)
        {
            case 1:
               insert();
               break;

            case 2:
                display();
                break;

            case 3:
                delete();
                break;


            case 4:
                exit(0);
                break;

            default:
                printf("Invalid choice\n");
        }
    }
}
SZ
```

```
1.insert
2. display
3. delete
4. exit
Enter your choice: 1
Enter the data
2

1.insert
2. display
3. delete
4. exit
Enter your choice: 1
Enter the data
3

1.insert
2. display
3. delete
4. exit
Enter your choice: 1
Enter the data
4

1.insert
2. display
3. delete
4. exit
Enter your choice: 2
234
1.insert
2. display
3. delete
4. exit
Enter your choice: 3

1.insert
2. display
3. delete
4. exit
Enter your choice: 2
34
1.insert
2. display
3. delete
4. exit
Enter your choice:
```

```c
#include<stdio.h>
#include<math.h>

struct node
{
    int data;
    struct node *next;
}*top;

void push()
{
    int n;

    struct node *newnode;
    printf("Enter the data\n");
    scanf("%d",&n);
    newnode=(struct node*)malloc(sizeof(struct node));
    newnode->data=n;
    newnode->next=top;
    top=newnode;

}

void display()
{
    struct node *temp;
    temp=top;
    while(temp!=0)
    {
        printf("%d\t",temp->data);
        temp=temp->next;
    }
}
```

```
    struct node *newnode;
    printf("Enter the data\n");
    scanf("%d",&n);
    newnode=(struct node*)malloc(sizeof(struct node));
    newnode->data=n;
    newnode->next=top;
    top=newnode;

}

void display()
{
    struct node *temp;
    temp=top;
    while(temp!=0)
    {
        printf("%d\t",temp->data);
        temp=temp->next;
    }
}
 void pop()
 {
     struct node *temp,*nextnode;
     temp=top;
     nextnode=temp->next;
     top=nextnode;
     free(temp);



 }

void main()
{
    int choice;
```

```c
void main()
{
    int choice;
    while(1)
    {
    printf("Enter the choice\n");
    scanf("%d",&choice);
    printf("Enter 1.push 2.display   3.pop   4.exit\n");
    switch(choice)
    {
        case 1:push();
        break;
        case 2:display();
        break;
        case 3:pop();
        break;
        case 4:exit(0);
        break;
        default :
            printf("invalid input\n");
    }
}
}
```

```
1.push
2. display
3. pop
4. exit
Enter your choice: 1
Enter the data
2

1.push
2. display
3. pop
4. exit
Enter your choice: 1
Enter the data
3

1.push
2. display
3. pop
4. exit
Enter your choice: 1
Enter the data
4

1.push
2. display
3. pop
4. exit
Enter your choice: 2
432
1.push
2. display
3. pop
4. exit
Enter your choice: 3

1.push
2. display
3. pop
4. exit
Enter your choice: 2
32
1.push
2. display
3. pop
4. exit
Enter your choice:
```

```c
#include<stdio.h>
#include<math.h>

struct node
{
    int data;
    struct node *next;
}*head,*newnode,*temp,*head2;

void createll1()
{
    int n;
    printf("Enter the data\n");
    scanf("%d",&n);
    newnode=(struct node*)malloc(sizeof(struct node));
    newnode->data=n;
    if(head==NULL)
    {
        head=temp=newnode;
    }
    else{
        temp->next=newnode;
        temp=newnode;
        newnode->next=NULL;
    }
}

void createll2()
{
     int n;
    printf("Enter the data\n");
    scanf("%d",&n);
    newnode=(struct node*)malloc(sizeof(struct node));
    newnode->data=n;
    if(head2==NULL)
    {
```

```c
void createll2()
{
     int n;
    printf("Enter the data\n");
    scanf("%d",&n);
    newnode=(struct node*)malloc(sizeof(struct node));
    newnode->data=n;
    if(head2==NULL)
    {
        head2=temp=newnode;
    }
    else{
        temp->next=newnode;
        temp=newnode;
        newnode->next=NULL;
    }
}

void display1()
{
    temp=head;
    while(temp!=NULL)
    {
        printf("%d",temp->data);
        temp=temp->next;
    }
}


void display2()
{
    temp=head2;
    while(temp!=NULL)
    {
        printf("%d",temp->data);
        temp=temp->next;
```

```
void display2()
{
    temp=head2;
    while(temp!=NULL)
    {
        printf("%d",temp->data);
        temp=temp->next;
    }
}


void reverse()
{
    struct node*prevnode,*nextnode;
    temp=nextnode=head;
    prevnode=NULL;
    while(temp!=NULL)
    {
        temp=temp->next;
        nextnode->next=prevnode;
        prevnode=nextnode;
        nextnode=temp;
    }
    head=prevnode;
}

void sorting()
{
    int c;
    struct node *nextnode;
    temp=nextnode=head;
    while(temp->next !=NULL)
    {
        nextnode=temp->next;
```

```
void sorting()
{
    int c;
    struct node *nextnode;
    temp=nextnode=head;
    while(temp->next !=NULL)
    {
        nextnode=temp->next;
        while(nextnode!=NULL)
        {
            if(temp->data>nextnode->data)
            {
                c=temp->data;
                temp->data=nextnode->data;
                nextnode->data=c;
            }
            nextnode=nextnode->next;
        }
        temp=temp->next;
    }
}


void concatenate()
{
    temp=head;
    while(temp->next!=NULL)
    {
        temp=temp->next;
    }
    temp->next=head2;
}

void main()
{
```

```
void concatenate()
{
    temp=head;
    while(temp->next!=NULL)
    {
        temp=temp->next;
    }
    temp->next=head2;
}

void main()
{
     int choice;

    while (1)
    {
        printf("\n1. create linked list1\n");
        printf("2. display1\n");
        printf("3. create a linkedlist2\n");
        printf("4. display2\n");
        printf("5. reverse\n");
        printf("6. sorting\n");
        printf("7. concatenation\n");
        printf("8. exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice)
        {
            case 1:
               createll1();
               break;

            case 2:
                display1();
                break;
```

```
switch (choice)
{
    case 1:
       createll1();
       break;

    case 2:
        display1();
        break;

    case 3:
        createll2();
        break;

    case 4:
        display2();
        break;

    case 5:
       reverse();
       break;

    case 6:
        sorting();
        break;

    case 7:
        concatenate();
        break;

    case 8:
        exit(0);
        break;

    default:
        printf("Invalid choice\n");
```

```
1. create linked list1
2. display1
3. create a linkedlist2
4. display2
5. reverse
6. sorting
7. concatenation
8. exit
Enter your choice: 1
Enter the data
2

1. create linked list1
2. display1
3. create a linkedlist2
4. display2
5. reverse
6. sorting
7. concatenation
8. exit
Enter your choice: 1
Enter the data
3

1. create linked list1
2. display1
3. create a linkedlist2
4. display2
5. reverse
6. sorting
7. concatenation
8. exit
Enter your choice: 1
Enter the data
4

1. create linked list1
2. display1
3. create a linkedlist2
4. display2
5. reverse
6. sorting
7. concatenation
8. exit
Enter your choice: 2
234
1. create linked list1
2. display1
3. create a linkedlist2
4. display2
```

```
Enter your choice: 2
432
1. create linked list1
2. display1
3. create a linkedlist2
4. display2
5. reverse
6. sorting
7. concatenation
8. exit
Enter your choice: 6

1. create linked list1
2. display1
3. create a linkedlist2
4. display2
5. reverse
6. sorting
7. concatenation
8. exit
Enter your choice: 2
234
1. create linked list1
2. display1
3. create a linkedlist2
4. display2
5. reverse
6. sorting
7. concatenation
8. exit
Enter your choice: 7

1. create linked list1
2. display1
3. create a linkedlist2
4. display2
5. reverse
6. sorting
7. concatenation
8. exit
Enter your choice: 2
234547
1. create linked list1
2. display1
3. create a linkedlist2
4. display2
5. reverse
6. sorting
7. concatenation
8. exit
Enter your choice: S
```