

```
#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");  
    }  
}
```

```
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: |
```

```

#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;
```

```
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: |
```



```
#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)
    {
```

```

28 void createll2()
29 {
30     int n;
31     printf("Enter the data\n");
32     scanf("%d", &n);
33     newnode=(struct node*) malloc(sizeof(struct node));
34     newnode->data=n;
35     if(head2==NULL)
36     {
37         head2=temp=newnode;
38     }
39     else{
40         temp->next=newnode;
41         temp=newnode;
42         newnode->next=NULL;
43     }
44 }
45
46 void display1()
47 {
48     temp=head;
49     while(temp!=NULL)
50     {
51         printf("%d", temp->data);
52         temp=temp->next;
53     }
54 }
55
56 void display2()
57 {
58     temp=head2;
59     while(temp!=NULL)
60     {
61         printf("%d", temp->data);
62         temp=temp->next;
63     }
64 }

```

```
7 void display2()
8 {
9     temp=head2;
10    while(temp!=NULL)
11    {
12        printf("%d",temp->data);
13        temp=temp->next;
14    }
15 }
16
17
18 void reverse()
19 {
20     struct node*prevnode,*nextnode;
21     temp=nextnode=head;
22     prevnode=NULL;
23     while(temp!=NULL)
24     {
25         temp=temp->next;
26         nextnode->next=prevnode;
27         prevnode=nextnode;
28         nextnode=temp;
29     }
30     head=prevnode;
31 }
32
33 void sorting()
34 {
35     int c;
36     struct node *nextnode;
37     temp=nextnode=head;
38     while(temp->next !=NULL)
39     {
40         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()
```

```
{
```

```

6 void concatenate()
7 {
8     temp=head;
9     while(temp->next!=NULL)
10    {
11        temp=temp->next;
12    }
13    temp->next=head2;
14 }
15
16 void main()
17 {
18     int choice;
19
20     while (1)
21     {
22         printf("\n1. create linked list1\n");
23         printf("2. display1\n");
24         printf("3. create a linkedlist2\n");
25         printf("4. display2\n");
26         printf("5. reverse\n");
27         printf("6. sorting\n");
28         printf("7. concatenation\n");
29         printf("8. exit\n");
30         printf("Enter your choice: ");
31         scanf("%d", &choice);
32
33         switch (choice)
34         {
35             case 1:
36                 createll1();
37                 break;
38
39             case 2:
40                 display1();
41                 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|