```
        current=next;
    }
    if(str==1)
     head1=prev;
    else if(str==2)
     head2=prev;
     display(str);
}

void display(int str)
{
    struct node *ptr=NULL;
      if(str==1)
       ptr=head1;
      else if(str==2)
       ptr=head2;

    if(ptr==NULL)
    {
        printf("Nothing to print\n");
    }
    else
    {
        while(ptr!=NULL)
       {
        printf("%d ",ptr->data);
        ptr=ptr->next;
       }
    }

}
```

```
clang-7 -pthread -lm -o main main.c
./main

1. Create
2. Display
3. Reverse
4. Concatinate
5. Exit
Enter your choice : 1
Enter the list to be added to:
1
Enter the data : 1
Node created

1. Create
2. Display
3. Reverse
4. Concatinate
5. Exit
Enter your choice : 1
Enter the list to be added to:
1
Enter the data : 2
Node created

1. Create
2. Display
3. Reverse
4. Concatinate
5. Exit
Enter your choice : 1
Enter the list to be added to:
1
```


```
        current=next;
    }
    if(str==1)
     head1=prev;
    else if(str==2)
     head2=prev;
     display(str);
}

void display(int str)
{
    struct node *ptr=NULL;
      if(str==1)
       ptr=head1;
      else if(str==2)
       ptr=head2;

    if(ptr==NULL)
    {
        printf("Nothing to print\n");
    }
    else
    {
        while(ptr!=NULL)
       {
        printf("%d ",ptr->data);
        ptr=ptr->next;
       }
    }

}
```


```
Enter your choice : 1
Enter the list to be added to:
1
Enter the data : 3
Node created

1. Create
2. Display
3. Reverse
4. Concatinate
5. Exit
Enter your choice : 1
Enter the list to be added to:
2
Enter the data : 4
Node created

1. Create
2. Display
3. Reverse
4. Concatinate
5. Exit
Enter your choice : 1
Enter the list to be added to:
2
Enter the data : 5
Node created

1. Create
2. Display
3. Reverse
4. Concatinate
5. Exit
```

```
        current=next;
    }
    if(str==1)
     head1=prev;
    else if(str==2)
     head2=prev;
     display(str);
}

void display(int str)
{
    struct node *ptr=NULL;
      if(str==1)
       ptr=head1;
      else if(str==2)
       ptr=head2;

    if(ptr==NULL)
    {
        printf("Nothing to print\n");
    }
    else
    {
        while(ptr!=NULL)
       {
        printf("%d ",ptr->data);
        ptr=ptr->next;
       }
    }

}
```

```
1. Create
2. Display
3. Reverse
4. Concatinate
5. Exit
Enter your choice : 2
Enter the string to be displayed
1
1 2 3
1. Create
2. Display
3. Reverse
4. Concatinate
5. Exit
Enter your choice : 2
Enter the string to be displayed
2
4 5
1. Create
2. Display
3. Reverse
4. Concatinate
5. Exit
Enter your choice : 3
Enter the list to be reversed
1
3 2 1
1. Create
2. Display
3. Reverse
4. Concatinate
5. Exit
```

```
        current=next;
    }
    if(str==1)
     head1=prev;
    else if(str==2)
     head2=prev;
     display(str);
}

void display(int str)
{
    struct node *ptr=NULL;
      if(str==1)
       ptr=head1;
      else if(str==2)
       ptr=head2;

    if(ptr==NULL)
    {
        printf("Nothing to print\n");
    }
    else
    {
        while(ptr!=NULL)
       {
        printf("%d ",ptr->data);
        ptr=ptr->next;
       }
    }

}
```

```
4 5
1. Create
2. Display
3. Reverse
4. Concatinate
5. Exit
Enter your choice : 3
Enter the list to be reversed
1
3 2 1
1. Create
2. Display
3. Reverse
4. Concatinate
5. Exit
Enter your choice : 3
Enter the list to be reversed
2
5 4
1. Create
2. Display
3. Reverse
4. Concatinate
5. Exit
Enter your choice : 4
3 2 1 5 4
1. Create
2. Display
3. Reverse
4. Concatinate
5. Exit
Enter your choice : 5
```