```
1 p/*WAP Implement Single Link List with following operations
🗏 🔗 Func
            a) a) Sort the linked list. b) Reverse the linked list. c) Concatenation of two linked lists*/
  cor 🤏
             #include<stdio.h>
  #include<conio.h>
             #include<stdlib.h>

    get

            #includecess.h>
  ø ins
             struct node
  日
            int info;

ø orc

             struct node *link;
  11
Struc
             typedef struct node *NODE;
🗏 🦻 noc
             NODE getnode()
        14
   🧓 iı
        15
             NODE X;
   o li
            x=(NODE) malloc(sizeof(struct node));
🗏 🦻 Type
             if (x==NULL)
        18
  ₱ NC
        19
             printf("mem full\n");
             exit(0);
        21
        22
             return x;
        23
        24
             NODE insert rear (NODE first, int item)
        25
        26
             NODE temp, cur;
        27
             temp=getnode();
        28
             temp->info=item;
             temp->link=NULL;
        29
        30
             if (first==NULL)
        31
             return temp;
             cur=first;
        33
             while (cur->link!=NULL)
        34
             cur=cur->link;
        35
            cur->link=temp;
< >
          col: 94
                sel: 0
                                                           filetype: C scope: unknown
line: 2 / 146
                        INS
                             TAB
                                  mode: CRLF
                                             encoding: UTF-8
```

```
35
             cur->link=temp;
  return first;
        37

    get

             void display(NODE first)
  ø ins
        39
  ma
             NODE temp;
             if (first==NULL)

ø orc

            printf("list empty");
  for(temp=first;temp!=NULL;temp=temp->link)
Struc
        44
🗏 🦻 noc
        45
             printf("%d\n", temp->info);
        46
   🦆 jı
        47
   % li
             NODE concat (NODE first, NODE second)
🗆 👂 Type
        49
        50
             NODE cur;
  » NC
        51
             if (first==NULL)
             return second;
             if (second==NULL)
             return first;
        55
             cur=first;
             while (cur->link!=NULL)
        57
             cur=cur->link;
             cur->link=second;
        59
             return first;
        60
        61
             NODE reverse (NODE first)
        62
        63
             NODE cur, temp;
             cur=NULL;
        65
             while (first!=NULL)
        66
            temp=first;
< >
                                              encoding: UTF-8
ine: 2 / 146
          col: 94
                 sel: 0
                        INS
                             TAB
                                  mode: CRLF
                                                            filetype: C scope: unknown
```

while(cur->link!=NULL)

cur=cur->link;

🛮 🔗 Func

```
cur=NULL;
  OCOr
             while(first!=NULL)
  66
        67
             temp=first;

    get

             first=first->link;
  ø ins
             temp->link=cur;

    ma

        70
             cur=temp;
        71

    orc

             return cur;
  73
Struc
             NODE order list(int item, NODE first)
        75
🗏 🦻 noc
        76
             NODE temp, prev, cur;
   ii 🏺
             temp=getnode();
   % li
             temp->info=item;
🗎 🦻 Type
             temp->link=NULL;
             if(first==NULL) return temp;
  ♠ NC
        81
             if(item<first->info)
        82
             temp->link=first;
        84
             return temp;
        85
        86
             prev=NULL;
        87
             cur=first;
        88
             while (cur!=NULL&&item>cur->info)
        89
        90
             prev=cur;
        91
             cur=cur->link;
        92
        93
             prev->link=temp;
        94
             temp->link=cur;
        95
             return first;
        96
             int main()
< >
                                              encoding: UTF-8
line: 2 / 146
          col: 94
                 sel: 0
                        INS
                             TAB
                                   mode: CRLF
                                                             filetype: C scope: unknown
```

NODE cur, temp;

🛮 🔗 Func

```
temp->link=cur;
🗎 🔗 Func
             return first;
  Or 🗞
  97
             int main()

    get

             int item, choice, i, n;
  NODE first=NULL, a, b;
       101
             for(;;)

    orc 102

             printf("1.insert front\n2.concat\n3.reverse\n4.order list\n5.dislay\n6.exit\n");

    rev

             printf("enter the choice\n");
Struc 105
             scanf("%d", &choice);
■ > not 106
             switch (choice)
       107
            申{
   ii 🏺
       108
             case 1:printf("enter the item\n");
       109
             scanf("%d", &item);
■ > Type 110
             first=insert rear(first,item);
  NC 111
             break;
             case 2:printf("enter the no of nodes in 1\n");
       112
       113
             scanf ("%d", &n);
       114
             a=NULL;
       115
             for (i=0; i<n; i++)
       116
       117
             printf("enter the item\n");
       118
             scanf("%d", &item);
       119
             a=insert rear(a,item);
       120
       121
             printf("enter the no of nodes in 2\n");
       122
             scanf ("%d", &n);
       123
             b=NULL;
       124
             for(i=0;i<n;i++)
       125
       126
             printf("enter the item\n");
       127
             scanf("%d", &item);
       128
            b=insert rear(b.item);
< >
          col: 94
                 sel: 0
                             TAB
                                  mode: CRLF
                                              encoding: UTF-8
                                                            filetype: C
                        INS
                                                                     scope: unknown
```

ine: 2 / 146

```
a=NULL;
  Ø cor
             for(i=0;i<n;i++)
  🤣 dis
      116
             printf("enter the item\n");

    get

             scanf("%d", &item);
             a=insert rear(a,item);
  120

    orc 121

             printf("enter the no of nodes in 2\n");
             scanf ("%d", &n);

    rev

             b=NULL;

    Struc 124
    124
             for(i=0;i<n;i++)
∃ 👂 noc 125
             printf("enter the item\n");
       126
             scanf("%d", &item);
       128
             b=insert rear(b,item);
■ > Type 129
  NC 130
             a=concat(a,b);
       131
             display(a);
       132
             break;
             case 3:first=reverse(first);
       133
       134
             display(first);
       135
             break;
       136
             case 4:printf("enter the item to be inserted in ordered list\n");
       137
             scanf("%d", &item);
       138
             first=order list(item, first);
       139
             break;
       140
             case 5:display(first);
       141
             break;
       142
             default:exit(0);
       143
       144
       145
       146
< >
ine: 2 / 146
          col: 94
                 sel: 0
                        INS
                             TAB
                                   mode: CRLF
                                              encoding: UTF-8
                                                            filetype: C scope: unknown
```

scanf ("%d", &n);

```
1.insert_front
2.concat
3.reverse
4.order list
5.dislay
6.exit
enter the choice
enter the item
1.insert_front
2.concat
3.reverse
4.order list
5.dislay
6.exit
enter the choice
enter the item
20
1.insert_front
2.concat
3.reverse
4.order list
5.dislay
6.exit
enter the choice
10
20
1.insert_front
2.concat
3.reverse
4.order list
5.dislay
6.exit
enter the choice
enter the no of nodes in 1
enter the item
```

```
6.exit
enter the choice
enter the no of nodes in 1
enter the item
11
enter the item
22
enter the no of nodes in 2
enter the item
33
enter the item
enter the item
55
11
22
55
1.insert_front
2.concat
3.reverse
4.order list
5.dislay
6.exit
enter the choice
10
20
1.insert_front
2.concat
3.reverse
4.order list
5.dislay
6.exit
enter the choice
```

5.dislay

```
3.reverse
4.order list
5.dislay
6.exit
enter the choice
enter the item
30
1.insert_front
2.concat
3.reverse
4.order list
5.dislay
6.exit
enter the choice
enter the item
1.insert_front
2.concat
3.reverse
4.order list
5.dislay
6.exit
enter the choice
10
20
30
1.insert_front
2.concat
3.reverse
4.order list
5.dislay
6.exit
enter the choice
20
```

```
6.exit
enter the choice
30
20
10
1.insert_front
2.concat
3.reverse
4.order list
5.dislay
6.exit
enter the choice
enter the item to be inserted in ordered_list
1.insert_front
2.concat
3.reverse
4.order list
5.dislay
6.exit
enter the choice
enter the item to be inserted in ordered_list
1.insert_front
2.concat
3.reverse
4.order list
5.dislay
6.exit
enter the choice
enter the item to be inserted in ordered_list
1.insert_front
2.concat
3.reverse
```

5.dislay

```
6.exit
enter the choice
enter the item to be inserted in ordered_list
1.insert_front
2.concat
3.reverse
4.order list
5.dislay
6.exit
enter the choice
enter the item to be inserted in ordered_list
1.insert_front
2.concat
3.reverse
4.order list
5.dislay
6.exit
enter the choice
11
23
10
41
54
1.insert_front
2.concat
3.reverse
4.order list
5.dislay
6.exit
enter the choice
11
23
```