```
main.c
                                                                                        1
     #include<stdio.h>
 2
     #include<stdlib.h>
 3
 4
     struct node
 5
 6
        int info;
 7
        struct node *link;
 8
      };
 9
      typedef struct node *NODE;
10
11
      NODE getnode()
12
13
          NODE x;
14
          x = (NODE)malloc(sizeof(struct node));
15
          if(x==NULL)
16
          {
17
               printf("\nMemory is full\n");
18
               exit(0);
19
 20
           return x;
 21
 22
 23
      NODE insert_front(NODE first, int item)
 24
      {
 25
          NODE temp;
          temp=getnode();
 26
 27
          temp->info=item;
           temp->link=NULL;
 28
           if(first==NULL)
 29
 30
              return temp;
 31
 32
           temp->link=first;
 33
           first=temp;
 34
           return first;
 35
 36
 37
      NODE delete_front(NODE first)
 38
      {
 39
 40
           NODE temp;
           if(first==NULL)
 41
 42
           {
               printf("List is empty. Cannot delete\n");
 43
```

```
main.c
                                                                                      42
           {
  43
                printf("List is empty. Cannot delete\n");
  44
                return first:
 45
           }
 46
           temp=first;
 47
           temp = temp->link;
           printf("Item deleted at front end is %d\n",first->info);
 48
 49
           free(first);
 50
           return temp;
 51
 52
 53
       NODE IF(NODE second, int item)
 54
       {
 55
           NODE temp;
 56
           temp=getnode();
 57
           temp->info=item;
 58
           temp->link=NULL;
 59
           if(second==NULL)
 60
               return temp;
 61
           temp->link=second;
 62
           second=temp;
 63
           return second;
 64
 65
 66
      NODE IR(NODE second, int item)
 67
      {
68
          NODE temp, cur;
69
          temp=getnode();
          temp->info=item;
70
          temp->link=NULL;
71
72
          if(second==NULL)
               return temp;
73
          cur=second;
74
          while(cur->link!=NULL)
75
              cur=cur->link;
76
          cur->link=temp;
77
78
          return second;
79
80
     NODE reverse(NODE first)
81
82
83
         NODE cur, temp;
84
          cur=NULL;
```

```
main.c
123
124
125
      NODE descending(NODE first)
126
      {
          NODE prev=first;
127
128
          NODE cur=NULL;
129
         int temp;
         if(first==NULL)
130
131
132
            return 0;
133
         else
134
135
136
             while(prev!= NULL)
137
138
                  cur = prev->link;
139
                 while(cur!= NULL)
140
141
                        if(prev->info < cur->info)
142
143
                            temp = prev->info;
                          prev->info = cur->info;
144
                           cur->info = temp;
145
146
                      cur = cur->link;
147
148
                  prev= prev->link;
149
150
151
           return first;
152
       }
153
154
       NODE concatenate(NODE first, NODE second)
155
       {
156
           NODE cur;
157
            if(first==NULL)
 158
                return second;
 159
            if(second==NULL)
 160
                return first;
 161
            cur=first;
 162
            while(cur->link!=NULL)
 163
            {
 164
```

```
main.c
164
          cur=cur->link;
165
166
          cur->link=second;
167
          return first;
168
169
170
      void display(NODE first)
171
172
173
          NODE temp;
          if(first==NULL)
174
              printf("List is empty. Cannot display items.\n");
175
          printf("List contents are : ");
176
          for(temp=first;temp!=NULL;temp=temp->link)
177
178
            printf("\n%d", temp->info);
179
180
181
      int main()
182
183
          int item, choice, pos, element, option, choice2, item1, num;
184
          NODE first=NULL;
185
          NODE second=NULL;
186
          for(;;)
187
188
               printf("\n\nChoose an option");
189
               printf("\n1:Insert_front \n2:Delete_front \n3:Reverse \n4:Sort
190
               \n5.Concatenate \n6:Display \n7:Exit\n");
               printf("Enter the choice : ");
191
               scanf("%d", &choice);
192
               switch(choice)
193
194
                   case 1: printf("Enter the item at front-end : ");
195
                         scanf("%d", &item);
196
                         first=insert_front(first,item);
197
                         printf("%d inserted at front-end.", first->info);
198
199
                         break:
200
                   case 2: first=delete_front(first);
201
                         break:
202
                   case 3: first=reverse(first);
203
                           printf("List is reversed.");
204
                           break:
205
                   case 4: printf("Press 1 for Ascending-sort and 2 for
```

```
case 4: printf("Press 1 for Ascending-sort and 2 for
205
                   Descending-sort : ");
                           scanf("%d", &option);
206
207
                            if(option==1)
                            {
208
209
                                first=ascending(first);
                                printf("List is sorted in ascending order.");
210
211
212
                            if(option==2)
213
                            {
214
                                first=descending(first);
                                printf("List is sorted in descending order.");
215
216
217
                            break;
218
                   case 5: printf("Create a second list\n");
                            printf("Enter the number of elements in the second list
219
                            : "):
                            scanf("%d", &num);
220
                            for(int i=1;i<=num;i++)</pre>
221
222
                                printf("\nPress 1 to Insert-front and 2 to
223
                                Insert-rear : ");
                                scanf("%d", &choice2);
224
                                if(choice2==1)
225
226
                                    printf("Enter the item at front-end : ");
227
                                  scanf("%d",&item1);
228
                                    second=IF(second, item1);
229
230
                                if(choice2==2)
231
232
                                    printf("Enter the item at rear-end : ");
233
                                  scanf("%d",&item1);
234
                                    second=IR(second,item1);
235
236
237
                            first=concatenate(first, second);
238
                            printf("\nThe two lists are concatenated.");
239
                            break:
240
                   case 6: display(first);
241
242
                          break;
                   default:exit(0);
243
```

```
second=IR(second,item1);
235
236
237
                           first=concatenate(first, second);
238
                           printf("\nThe two lists are concatenated.");
239
                           break;
240
                  case 6: display(first);
241
                         break;
242
                  default:exit(0);
243
                         break;
244
           } }
245
246
```

```
clang-7 -pthread -lm -o main main.c
./main
Choose an option
1:Insert_front
2:Delete_front
3:Reverse
4:Sort
5. Concatenate
6:Display
7:Exit
Enter the choice: 1
Enter the item at front-end: 13
13 inserted at front-end.
Choose an option
1:Insert_front
2:Delete_front
3:Reverse
4:Sort
5. Concatenate
6:Display
7:Exit
Enter the choice: 1
Enter the item at front-end: 15
15 inserted at front-end.
Choose an option
1:Insert_front
2:Delete_front
3:Reverse
4:Sort
5. Concatenate
6:Display
7:Exit
Enter the choice: 3
```

List is reversed.

```
1:Insert_front
2:Delete_front
3:Reverse
4:Sort
5. Concatenate
6:Display
7:Exit
Enter the choice: 4
Press 1 for Ascending-sort and 2 for Descending-sort : 1
List is sorted in ascending order.
Choose an option
1:Insert_front
2:Delete_front
3:Reverse
4:Sort
5. Concatenate
6:Display
7:Exit
Enter the choice: 6
List contents are :
13
15
```

Choose an option

| | S.R. POOTA Classmate | 0 |
|---------|--|------|
| | 1BM19CS135 | |
| th bod. | | |
| 7110 | #include < stdio.h> | |
| | #include < Staller. h> | |
| | Skuct node | |
| | E int rife; interest to grat | |
| | struct node & wik; &; (was = home) | |
| | typedel skuct node * NODE; | |
| | NODE getnode() | |
| | E' NODE X', | |
| | x= (NODE) malloc (size of (8kuct node)); | |
| | if (x== NULL) (not the shore 3000) 97 From | |
| | E penty ("In Memory is full In"); | |
| | exitle); 3 | |
| | setuen x', 3 | |
| | NODE insert front (NODE first, int item) | |
| | { NODE temp; (DOM == brong) } | |
| 2 | temp= get node(); | |
| | temp -> link= NUU', | |
| | of (first = 2 NULL) (Jun = 1 Am) (Lucy) election | |
| | E setuen tenp; 3 | |
| • . | temp > link = first; ignot a tril (= 200) | |
| | first z temp; & brown nails | |
| | setuen fæst; 3 (1008 300m) would 300m | |
| | propE delete front (NODE first) | |
| | Cut-Nucl. | |
| | NODE temp; 3 (Dona 1 too) distant | |
| | J (first == NUU) { | |
| | pent ("list is enpty. Carnot delete \n"); | |
| | setuen first; 3 | |
| | temp = first; | |
| | tenp= temp-> link; | |
| | pentil (" Item deleted at front end is xd in", first) into); | |
| | fee (1003t); | |
| | setuen temp; 3 | |
| | NODE OF (NODE second, int item) | |
| | (1304 = 326) Ji | 13 |
| | | 4450 |

| | S.R. POOTA 1BM1905135 | elassmate Date Page |
|---------|---|-------------------------------|
| | NODE temp; | Ld.oible & statio.b. |
| | temp= getnodel); | Midade Estalla h> |
| | temp -> info = Etem; | shor trust |
| | temp -> link= NULL' | oliv dis 3 |
| | if (second = = NULL) | if ished in whom trucks ! |
| | setur temp' 13001 | |
| | temp > link = Second; | Mehanton 3004 |
| | second = temp; | 'x 3000 " |
| | Setuen second; 3 | arcie) sallon (3901) =x |
| | NODE IR (NODE second, int | iten) (1911-12) 16 |
| | I NODE temp, cue; (al) | in warmen (11") (Hinga 3 |
| | temp=get node(); | E Calling |
| | temp-> info = etem, | F. rick neutro |
| | temp -> link= MUL' to 1000 | Fami took toon man |
| | of (Second=2 NOU) | (not tenp |
| | setuen temp | Claban top - grost. |
| | cutz second; | dein school strid to good |
| | white (cur > link != NULL) | (JUM == +215) B. |
| | cue= cue > link) | & ignoh senting 3 |
| | case -> link = temp, | tent = list; |
| | seturn second; 3' | ignot = title |
| | NODE leverye (NODE first) | |
| | E NODE cue, temp; | gon) those delete front (not) |
| | | 3 |
| | while (first 1= NULL) & | NODE HORD |
| | temp = first, | 6 . / 1 |
| | first = first >link; temp >link = cur; | with Colotte is enet. C |
| | cus = temp; 3 | & tool newles |
| | Return cue; 3 | items and |
| 1 · (do | NODE ascending (NODE fei | white good good |
| | E NODE prev= first; | St) hould matter of Hong |
| | NODE cur: NULL; | (deno) mel |
| | int temp; | I gras outle |
| | y (first == rull) | 1 posts 30019 45 3000 |

| | | classmate | |
|--|----------------|-------------------------------------|---|
| S.R. POOTA | | Date | 2 |
| 1BM1905135 | | Paga | |
| 0 1- | 0 | | |
| & return 0', | 3 | trul centre | |
| else E | | Cus test | |
| while (prev!= N | | (word link - such distribute) | |
| cue= prev -> linh | | f cus - cus - Stable | |
| while (cur!= N | | | |
| E if (prev- | > info as cul. | control have and conic | |
| E temp=prev | > info', | हैं (नेहार्ग mades | |
| prev->info= | cue > into ; | (Leaf Mar) polysis by | - |
| cue > info = | temp; & | good 7004 J | |
| C 1 2 - C 1 - 1 - 1 | . 8 | (wuy = + + raf) to | |
| Prev = brev > 1 | ink is 300 + | print ("Wit is emply Carred ast:"): | |
| 3 getien 1 | cest; 3 | suit (" list contents are: "). | |
| NODE desiend | ula CNODE ! | dis (temp-frest) temple (test) | |
| E NODE PREV | | | |
| NODE CUS: | NUL' SE | opik- quot, "b ra!") Hong | |
| int temp; | , | Ocion da | |
| 1 (150+==NI | out) { - | int them choice por, ele | |
| Return 0', 3 | | NOTE test 300h | |
| else { | | 'Must strend = Dual' | |
| while (prev != | NULL) ? | (*,)&+ | |
| 6.60 T proud 1 | of the star | o sport (" In In Charge o | |
| A il Cure to a | cui) S | partie Trachant 10 8 | |
| of College of | 1000) | 18 04 5 04 05 | |
| of chind he | cur-s info | In 5 Concatents in 631 | |
| temp= prev-) | info | printle " Eater the choice:" | |
| | | Scarl (" t d! & chare); | |
| | | 3 Conner Cohnice | |
| curs cul -> link | , Ito mati | east i porty (" Eater the | |
| pew= pew -> lin | k, } | (mar & (wird" & cken) | |
| 4 | (cost, item) | 1 trul with front ? | |
| Return Liest !! | gentless to | balasai b. () Hasq | |
| NODE concateno | te (NODE. | first, NODE second) | |
| 11 11 11 11 11 11 11 11 11 11 11 11 11 | | Tore 2: first = dolote part (fi | |
| y (first = NULL) | | o o interest | |
| | | com 3 tiet somes (first) | |
| il (second sent | MAST - Comment | or a 1811") thing | |
| 60.00 | Co marine | | |

| | Date |
|--|-------------------------------------|
| S.R. POOTA | Page |
| IBM19CS135 | |
| (Orters) | E is raider in |
| zetuen füst; | F rela |
| cue dist | 2 CHUM-lung I dida |
| while (cue > link!=NUL) | " Hord & 1229 2212) |
| E cue = cue -> link', | Liver Jews) diday |
| 3 | 18 4 4000 1 3 |
| cue > link = second', | sus co office and a |
| Return filest:, 3 | discount in |
| void display (note first) | (0 0 C 20) - 0 0 C 200 |
| £ NODE temp; | of and - of or - Mil |
| . (12. | said - Rud - Rud |
| print ("Ust is empty Carnot | display itens. (n"), |
| 1 2 2 1 1 1 201 1 1 201 1 1 201 1 1 1 201 1 1 1 | 1 1100 |
| for Chemp= first ; temp!= NULL; | temp= temp->link) |
| Teller of the state of the stat | : trul = werg Forant |
| party ("\n'.d", temp > info); | 33 Dod ses From |
| int main () | got do |
| & int item, choice pos, elen | nent, option, choice 2, Hend, nun', |
| NODE first = NULL', | of 10 natibe |
| NODE Second = NULL' | 3 22/9 |
| | 3 (www =) useg) sterior |
| | option"); a) costa = ano |
| ant ("In 1: Inserthant In 2: | Delete jent In 3. Reveer n 4. Soit |
| in 5. Concatenate in 6: DRg | Slow 107.2 Exit 10"1" |
| party (" Enter the choice: ") | doi: |
| scary (". / d', & choice); | obs (- Sep) = 1 |
| gwitch (choice) [| S do and |
| case 1: penty (" Enter the i | tem of John 1: 12: |
| scan fluy, d' 1 item) | an fint-end. |
| first = insert_fent (fi | That was many |
| sent ("Yd inserted | Mat (tem) |
| lenk' | t fentend!; fiest -> info); |
| Jreah; Case 2: first = delete_fent (first | 17: |
| lreak' | |
| 2' 1:01 - 20,0180 (1:00) 1 | A (Probes Morr) |
| ease 3: fiest = severse (fiest); pents ("list is sev | 888 d 471 |
| ha 6 c 33 300 | and · () was a house) li |