

Lab Program :-

- 7) Write a program implement Single link list with following operations  
 a) Sort the linked list. b) ~~Reverse~~ Reverse linked list  
 c) Concatenation of two linked lists.

A) #include <stdio.h>

#include <conio.h>

#include <malloc.h>

#include <stdlib.h>

Struct node {

int info;

Struct node \*link;

}

typedef Struct node \*NODE;

NODE getnode()

{ NODE x;

x = (NODE) malloc(sizeof(Struct node));

if (x == NULL)

printf ("mem full in");

exit(0);

}

return x;

}

void freenode(NODE x)

{ free(x);

}

NODE insert-front(NODE first, int item)

{ NODE temp;

temp = getnode();

temp->info = item;

temp->link = NULL;

if (first == NULL)

return temp;

temp->link = first;

first = temp;  
return first;

} NODE IF (NODE second, int item) {  
NODE temp;  
temp = getnode();  
temp->info = item;  
temp->link = NULL;  
if (second == NULL)  
return temp;  
temp->link = second;  
second = temp;  
return second;

} NODE delete-front (NODE first) {

NODE temp;  
if (first == NULL){  
printf ("list is empty cannot delete \n");  
return first;

} temp = first;  
temp = temp->link;  
printf ("item deleted at front-end is = %d \n", first->info);  
free(first);  
return temp;

} NODE insert-rear (NODE first, int item) {

NODE temp, cur;  
temp = getnode();  
temp->info = item;  
temp->link = NULL;  
if (first == NULL)  
return temp;  
cur = first;

while ( cur -> link != NULL )

    cur = cur -> link;

    cur -> link = temp;

return first;

}

node LR ( node second, int item ) {

    node temp, cur;

    temp = getnode();

    temp -> info = item;

    temp -> link = NULL;

    if ( second == NULL )

        return temp;

        cur = second;

    while ( cur -> link != NULL )

        cur = cur -> link;

        cur -> link = temp;

    return second;

}

node delete -> rear ( node first ) {

    node cur, prev;

    if ( first == NULL ) {

        printf (" list is empty cannot delete \n " );

        return first;

}

    if ( first -> link == NULL ) {

        printf (" item deleted is % d \ n ", first -> info );

        free ( first );

        return NULL;

}

    prev = NULL;

    cur = first;

    while ( cur -> link != NULL ) {

        prev = cur;

        cur = cur -> link;

}

```

        printf (" item deleted at rear end is %d ", cur->info );
        free (cur) ;
        forev->link = NULL ;
        return first ;
    }
}

```

```

NODE insert_pos (int item, int pos, NODE first) {
    NODE temp ;

```

```

    NODE forev, cur ;

```

```

    int count ;

```

```

    temp = getnode () ;

```

```

    temp->info = item ;

```

```

    temp->link = NULL ;

```

```

    if (first == NULL && pos == 1)

```

```

        return temp ;

```

```

    if (first == NULL) {

```

```

        printf (" Invalid pos \n ") ;

```

```

        return first ;
    }
}

```

```

if (pos == 1) {

```

```

    temp->link = first ;

```

```

    return temp ;
}
}

```

```

count = 1 ;

```

```

forev = NULL ;

```

```

cur = first ;

```

```

while (cur != NULL && count != pos) {

```

```

    forev = cur ;

```

```

    cur = cur->link ;

```

```

    count++ ;
}
}

```

```

if (count == pos) {

```

```

    forev->link = temp ;

```

```

    temp->link = cur ;

```

```

    return first ;
}
}

```

printf ("Invalid position in");  
return first;

}  
NODE delete\_pos (int pos, NODE first) {

NODE cur;

NODE prev;

int count;

if (first == NULL || pos < 0) {

printf ("invalid position\n");

return NULL;

}  
if (pos == 1) {

cur = first;

first = first -> link;

freemode (cur);

return first;

}  
prev = NULL;

cur = first;

count = 1;

while (cur != NULL) {

if (count == pos)

break;

prev = cur;

cur = cur -> link;

count++;

}  
if (count != pos) {

printf ("invalid position\n");

return first;

}  
prev -> link = cur -> link;

freemode (cur);

return first;

```
NODE reverse(NODE first){
```

```
    NODE cur, temp;
```

```
    cur = NULL;
```

```
    while (first != NULL){
```

```
        temp = first;
```

```
        first = first -> link;
```

```
        temp->link = cur;
```

```
        cur = temp;
```

```
}
```

{ (cur = first) };

```
    return cur;
```

```
}
```

{ (cur = first) };

```
NODE asc(NODE first){
```

```
    NODE forev = first;
```

```
    NODE cur = NULL;
```

```
    int temp;
```

```
if (first == NULL){
```

```
    return 0;
```

```
}
```

{ (cur = first) };

```
else {
```

```
    while (forev != NULL){
```

```
        cur = forev -> link;
```

```
        while (cur != NULL){
```

```
            if (forev -> info > cur -> info){
```

```
                temp = forev -> info;
```

```
                forev -> info = cur -> info;
```

```
                cur -> info = temp;
```

```
}
```

{ (cur -> info) };

```
            cur -> info = cur -> link;
```

```
}
```

{ (forev -> link) };

```
}
```

{ (return first) };

```
}
```

```
NODE desc(NODE first){
```

```
    NODE forev = first;
```

```
None cur = NULL;  
int temp;  
if (first == NULL){  
    return 0;  
}  
else{  
    while (forev != NULL){  
        cur = forev->link;  
        while (cur != NULL){  
            if (forev->info < cur->info){  
                temp = forev->info;  
                forev->info = cur->info;  
                cur->info = temp;  
            }  
            cur = cur->link;  
        }  
        forev = forev->link;  
    }  
    return first;  
}
```

```
NODE concate(NODE first, NODE second){
```

```
    NODE cur;  
    if (first == NULL)  
        return second;  
    if (second == NULL)  
        return first;  
    cur = first;  
    while (cur->link != NULL){  
        cur = cur->link;  
    }
```

```
    cur->link = second;  
    return first;
```

```
void display(NODE first){  
    NODE temp;  
    if (first == NULL)
```

```

if (list == NULL) printf("list empty cannot display items\n");
for (temp = first; temp != NULL; temp = temp->link) {
    printf("%d\n", temp->info);
}

```

83

```

int main() {
    int item, choice, pos, element, option, choice2, item1, num;
    system("cls");
    Node first = NULL;
    Node second = NULL;
    for ( ; ) {
        printf("\n 1: insert-front\n 2: delete-front\n 3: insert-rear\n"
               "4: delete-rear\n 5: random-position\n 6: reverse\n 7: sort\n 8: "
               "concat\n 9: display\n 10: Exit\n");
        scanf(" %d", &choice);
        switch (choice) {
            case 1: printf("enter the item at front-end\n");
                      scanf(" %d", &item);
                      first = insert_front(first, item);
                      break;
            case 2: first = delete_front(first);
                      break;
            case 3: printf("enter the item at rear-end\n");
                      scanf(" %d", &item);
                      first = insert_rear(first, item);
                      break;
            case 4: first = delete_rear(first);
                      break;
            case 5: printf("press 1 to insert or 2 to delete at any "
                           "desired position\n");
                      scanf(" %d", &element);
                      if (element == 1) {
                          printf("enter the position to insert\n");
                          scanf(" %d", &pos);
                      }

```

```
printf ("enter the item to insert : \n");
scanf ("%d", &item);
first = insert_pos(item, pos, first);
if (element == 2) {
    printf ("enter position to delete \n");
    scanf ("%d", &pos);
    first = delete_pos(pos, first);
}
break;
case 6: first = reverse(first);
break;
case 7: printf ("press 1 for ascending : - sort &
                2 for descending sort : \n");
scanf ("%d", &option);
if (option == 1)
    first = asc(first);
if (option == 2)
    first = desc(first);
break;
case 8: printf ("Create a second list \n");
printf ("enter the number of elements in second list : \n");
scanf ("%d", &num);
for (int i = 1; i <= num; i++) {
    printf ("In press 1 to insert front & 2 to insert rear in ");
    scanf ("%d", &choice2);
    if (choice2 == 1)
        printf ("enter the item at the front end \n");
    scanf ("%d", &item1);
}
if (choice2 == 2)
    printf ("enter the item at rear end \n");
    scanf ("%d", &item);
second = IR(second, item);
}
```

G Proposed Alternate Assessment Tool Plan (if applicable)

Question paper format

One Question to be asked for 20 Marks

One Question to be asked for 20 Marks

Two Questions to be asked for 20 Marks each

classmate

Date \_\_\_\_\_

Page \_\_\_\_\_

first = concat(first, second);

break;

case 9 : display(first);

break;

default : exit(0);

break;

}

getch();

return 0;

}

Linked\_list\_sort\_rev\_concat - [Linked\_list\_sort\_rev\_concat.dev] - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

(globals)

Linked\_list\_sort\_rev\_con.cpp

```
1 #include <stdio.h>
2 #include <conio.h>
3 #include <malloc.h>
4 #include <stdlib.h>
5 struct node{
6     int info;
7     struct node *link;
8 };
9 typedef struct node *NODE;
10 NODE getnode(){
11     NODE x;
12     x=(NODE) malloc(sizeof(struct node));
13     if(x==NULL)
14     {
15         printf("mem full \n");
16         exit(0);
17     }
18     return x;
19 }
20 void freenode(NODE x){
21     free(x);
22 }
23 NODE insert_front(NODE first,int item){
24     NODE temp;
25     temp=getnode();
26     temp->info=item;
27     temp->link=NULL;
28     if(first==NULL)
29         return temp;
30     temp->link=first;
31     first=temp;
32     return first;
33 }
34 NODE IF(NODE second,int item){
35     NODE temp;
36     temp=getnode();
37     temp->info=item;
```

Compiler Resources Compile Log Debug Find Results

Line: 218 Col: 16 Sel: 0 Lines: 323 Length: 5892 Insert Done parsing in 0.063 seconds

Search

09:18 06-12-2020 ENG

Linked\_list\_sort\_rev\_concat - [Linked\_list\_sort\_rev\_concat.dev] - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

(globals)

```
37 |     temp->info=item;
38 |     temp->link=NULL;
39 |     if(second==NULL)
40 |         return temp;
41 |     temp->link==second;
42 |     second=temp;
43 |     return second;
44 |
45 | NODE delete_front(NODE first){
46 | NODE temp;
47 | if(first==NULL){
48 | printf("list is empty cannot delete \n");
49 | return first;
50 | }
51 | temp=first;
52 | temp=temp->link;
53 | printf("item deleted at front-end is= %d \n",first->info);
54 | free(first);
55 | return temp;
56 |
57 | NODE insert_rear(NODE first,int item){
58 | NODE temp,cur;
59 | temp=getnode();
60 | temp->info=item;
61 | temp->link=NULL;
62 | if(first==NULL)
63 |     return temp;
64 | cur=first;
65 | while(cur->link!=NULL)
66 |     cur=cur->link;
67 | cur->link=temp;
68 | return first;
69 |
70 | NODE LR(NODE second,int item){
71 | NODE temp,cur;
72 | temp=getnode();
73 | temp->info=item;
```

Compiler Resources Compile Log Debug Find Results

Line: 218 Col: 16 Sel: 0 Lines: 323 Length: 5892 Insert Done parsing in 0.063 seconds

Search

09:18 06-12-2020 ENG

Linked\_list\_sort\_rev\_concat - [Linked\_list\_sort\_rev\_concat.dev] - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

(globals)

Linked\_list\_sort\_rev\_con.cpp

```
73 |     temp->info=item;
74 |     temp->link=NULL;
75 |     if(second==NULL)
76 |         return temp;
77 |     cur=second;
78 |     while(cur->link!=NULL)
79 |         cur=cur->link;
80 |         cur->link=temp;
81 |         return second;
82 |
83 | NODE delete_rear(NODE first){
84 |     NODE cur,prev;
85 |     if(first==NULL){
86 |         printf("list is empty cannot delete \n ");
87 |         return first;
88 |     }
89 |     if(first->link==NULL){
90 |         printf("item deleted is %d \n ",first->info);
91 |         free(first);
92 |         return NULL;
93 |     }
94 |     prev=NULL;
95 |     cur=first;
96 |     while(cur->link!=NULL)
97 |     {
98 |         prev=cur;
99 |         cur=cur->link;
100 |     }
101 |     printf("item deleted at rear end is %d",cur->info);
102 |     free(cur);
103 |     prev->link=NULL;
104 |     return first;
105 | }
106 | NODE insert_pos(int item,int pos,NODE first){
107 |     NODE temp;
108 |     NODE prev,cur;
109 |     int count;
```

Compiler Resources Compile Log Debug Find Results

Line: 218 Col: 16 Sel: 0 Lines: 323 Length: 5892 Insert Done parsing in 0.063 seconds

Search

09:18 06-12-2020 ENG

Linked\_list\_sort\_rev\_concat - [Linked\_list\_sort\_rev\_concat.dev] - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

(globals)

```
109 int count;
110 NODE temp=getnode();
111 temp->info=item;
112 temp->link=NULL;
113 if(first==NULL&&pos==1)
114 return temp;
115 if(first==NULL){
116 printf("invalid pos \n");
117 return first;
118 }
119 if(pos==1){
120 temp->link=first;
121 return temp;
122 }
123 count=1;
124 prev=NULL;
125 cur=first;
126 while(cur!=NULL&&count!=pos){
127 prev=cur;
128 cur=cur->link;
129 count++;
130 }
131 if(count==pos){
132 prev->link=temp;
133 temp->link=cur;
134 return first;
135 }
136 printf("invalid position \n");
137 return first;
138 }
139 NODE delete_pos(int pos, NODE first){
140 NODE cur;
141 NODE prev;
142 int count;
143 if(first==NULL||pos<=0){
144 printf("invalid position \n");
145 return NULL;
146 }
```

Compiler Resources Compile Log Debug Find Results

Line: 218 Col: 16 Sel: 0 Lines: 323 Length: 5892 Insert Done parsing in 0.063 seconds

Search

09:18 06-12-2020 ENG

Linked\_list\_sort\_rev\_concat - [Linked\_list\_sort\_rev\_concat.dev] - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

(globals)

```
145     return NULL;
146 }
147 if (pos==1){
148     cur=first;
149     first=first->link;
150     freenode(cur);
151     return first;
152 }
153 prev=NULL;
154 cur=first;
155 count=1;
156 while(curl=NULL){
157     if(count==pos)
158         break;
159     prev=cur;
160     cur=cur->link;
161     count++;
162 }
163 if(count!=pos){
164     printf("invalid position \n");
165     return first;
166 }
167 if(count!=pos){
168     printf("invalid position specified \n");
169     return first;
170 }
171 prev->link=cur->link;
172 freenode(cur);
173 return first;
174 }
175 NODE reverse(NODE first){
176     NODE cur,temp;
177     cur=NULL;
178     while(first!=NULL){
179         temp=first;
180         first=first->link;
181         temp->link=cur;
```

Compiler Resources Compile Log Debug Find Results

Line: 218 Col: 16 Sel: 0 Lines: 323 Length: 5892 Insert Done parsing in 0.063 seconds

Search

09:18 06-12-2020 ENG

Linked\_list\_sort\_rev\_concat - [Linked\_list\_sort\_rev\_concat.dev] - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

(globals)

```
181     temp->link=cur;
182     cur=temp;
183 }
184 return cur;
185 }
186 NODE asc(NODE first){
187     NODE prev=first;
188     NODE cur=NULL;
189     int temp;
190     if(first==NULL){
191         return 0;
192     }
193     else{
194         while(prev!=NULL){
195             cur=prev->link;
196             while(cur!=NULL){
197                 if(prev->info > cur->info){
198                     temp=prev->info;
199                     prev->info=cur->info;
200                     cur->info=temp;
201                 }
202                 cur=cur->link;
203             }
204             prev=prev->link;
205         }
206     }
207     return first;
208 }
209 NODE des(NODE first){
210     NODE prev=first;
211     NODE cur=NULL;
212     int temp;
213     if(first==NULL){
214         return 0;
215     }
216     else{
217         while(prev!=NULL){
```

Compiler Resources Compile Log Debug Find Results

Line: 218 Col: 16 Sel: 0 Lines: 323 Length: 5892 Insert Done parsing in 0.063 seconds

Search

09:19 06-12-2020 ENG

Linked\_list\_sort\_rev\_concat - [Linked\_list\_sort\_rev\_concat.dev] - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

(globals)

```
216 } else{
217     while(prev!=NULL){
218         cur=prev->link;
219         while(cur!=NULL){
220             if(prev->info < cur->info){
221                 temp=prev->info;
222                 prev->info=cur->info;
223                 cur->info=temp;
224             }
225             cur=cur->link;
226         }
227         prev=prev->link;
228     }
229     return first;
230 }
231
232 NODE concat(NODE first, NODE second){
233     NODE cur;
234     if(first==NULL)
235         return second;
236     if(second==NULL)
237         return first;
238     cur=first;
239     while(cur->link!=NULL){
240         cur=cur->link;
241     }
242     cur->link=second;
243     return first;
244 }
245 void display(NODE first){
246     NODE temp;
247     if(first==NULL)
248         printf("list empty cannot display items \n");
249     for(temp=first;temp!=NULL,temp=temp->link){
250         printf("%d \n",temp->info);
251     }
252 }
```

Compiler Resources Compile Log Debug Find Results

Line: 218 Col: 16 Sel: 0 Lines: 323 Length: 5892 Insert Done parsing in 0.063 seconds

Search

09:19 06-12-2020 ENG

Linked\_list\_sort\_rev\_concat - [Linked\_list\_sort\_rev\_concat.dev] - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

(globals)

```
252 L }
253 int main(){
254     int item choice, pos, element, option, choice2, item1, num
255     system("cls");
256     NODE first=NULL;
257     NODE second=NULL;
258     for(;;){
259         printf("\n 1: insert_front \n 2: delete_front \n 3: insert_rear \n 4: delete_rear \n 5: random_position \n 6: reverse \n 7: sort \n 8: exit");
260         printf("enter the choice \n");
261         scanf ("%d", &choice);
262         switch(choice){
263             case 1: printf ("enter the item at front-end \n");
264             scanf ("%d", &item);
265             first=insert_front(first,item);
266             break;
267             case 2: first=delete_front(first);
268             break;
269             case 3: printf ("enter the item at rear-end \n");
270             scanf ("%d", &item);
271             first=insert_rear(first,item);
272             break;
273             case 4: first=delete_rear(first);
274             break;
275             case 5: printf ("press 1 to insert or 2 to delete at any desired position \n");
276             scanf ("%d", &element);
277             if(element==1){
278                 printf ("enter the position to inset \n");
279                 scanf ("%d", &pos);
280                 printf ("enter the item to inset \n");
281                 scanf ("%d", &item);
282                 first=insert_pos(item,pos,first);}
283             if(element==2){
284                 printf ("enter the position to delete \n");
285                 scanf ("%d", &pos);
286                 first=delete_pos(pos,first);
287             }
288             break;
289         }
290     }
291 }
```

Compiler Resources Compile Log Debug Find Results

Line: 218 Col: 16 Sel: 0 Lines: 323 Length: 5892 Insert Done parsing in 0.063 seconds

Search

09:19 06-12-2020 ENG

Linked\_list\_sort\_rev\_concat - [Linked\_list\_sort\_rev\_concat.dev] - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

(globals)

```
287 }
288 break;
289 case 6: first=reverse(first);
290     break;
291 case 7: printf("press 1 for ascending sort and 2 for descending sort: \n");
292 scanf("%d", &option);
293 if(option==1)
294 first=asc(first);
295 if(option==2)
296 first=des(first);
297 break;
298 case 8: printf("create a second list \n");
299 printf("enter the number of elements in second list \n");
300 scanf("%d", &num);
301 for(int i=1; i<=num; i++){
302 printf("\n press 1 to insert front and 2 to insert rear \n");
303 scanf("%d", &choice2);
304 if(choice2==1){
305 printf("enter the item at front-end \n");
306 scanf("%d", &item1);
307 }
308 if(choice2==2){
309 printf("enter the item at rear-end \n");
310 scanf("%d", &item1);
311 second=L(second, item1);
312 }
313 }
314 first=concat(first, second);
315 break;
316 case 9: display(first);
317 break;
318 default: exit(0);
319 break;
320 }
321 getch();
322 return 0;
323 }
```

Compiler Resources Compile Log Debug Find Results

Line: 218 Col: 16 Sel: 0 Lines: 323 Length: 5892 Insert Done parsing in 0.063 seconds

Search

09:19 06-12-2020 ENG

C:\Users\sohan\Desktop\C Programs\Data Structures Lab\linked list sort rev concatenation\Linked\_list\_sort\_rev\_concat.exe

```
1:insert_front  
2: delete_front  
3: insert_rear  
4: delete_rear  
5: random_position  
6: reverse  
7: sort  
8: concate  
9: display_list  
10: exit  
enter the choice
```

```
1  
enter the item at front-end  
3
```

```
1:insert_front  
2: delete_front  
3: insert_rear  
4: delete_rear  
5: random_position  
6: reverse  
7: sort  
8: concate  
9: display_list  
10: exit  
enter the choice
```

```
1  
enter the item at front-end  
5
```

```
1:insert_front  
2: delete_front  
3: insert_rear  
4: delete_rear  
5: random_position  
6: reverse  
7: sort  
8: concate  
9: display_list  
10: exit  
enter the choice
```

```
3  
enter the item at rear-end  
2
```

```
1:insert_front  
2: delete_front  
3: insert_rear  
4: delete_rear
```



09:19 ENG 06-12-2020 3

C:\Users\sohan\Desktop\C Programs\Data Structures Lab\linked list sort rev concatenation\Linked\_list\_sort\_rev\_concat.exe

3  
enter the item at rear-end  
2

1:insert\_front  
2: delete\_front  
3: insert\_rear  
4: delete\_rear  
5: random\_position  
6: reverse  
7: sort  
8: concat  
9: display\_list  
10: exit

enter the choice  
3

enter the item at rear-end  
4

1:insert\_front  
2: delete\_front  
3: insert\_rear  
4: delete\_rear  
5: random\_position  
6: reverse  
7: sort  
8: concat  
9: display\_list  
10: exit

enter the choice  
6

1:insert\_front  
2: delete\_front  
3: insert\_rear  
4: delete\_rear  
5: random\_position  
6: reverse  
7: sort  
8: concat  
9: display\_list  
10: exit

enter the choice  
9

4

2

3

5

1:insert\_front

Search



09:19 ENG 06-12-2020



C:\Users\sohan\Desktop\C Programs\Data Structures Lab\linked list sort rev concatenation\Linked\_list\_sort\_rev\_concat.exe

```
1:insert_front
2: delete_front
3: insert_rear
4: delete_rear
5: random_position
6: reverse
7: sort
8: concat
9: display_list
10: exit
enter the choice
7
press 1 for ascending sort and 2 for descending sort:
1

1:insert_front
2: delete_front
3: insert_rear
4: delete_rear
5: random_position
6: reverse
7: sort
8: concat
9: display_list
10: exit
enter the choice
9
2
3
4
5

1:insert_front
2: delete_front
3: insert_rear
4: delete_rear
5: random_position
6: reverse
7: sort
8: concat
9: display_list
10: exit
enter the choice
8
create a second list
enter the number of elements in second list
3

press 1 to insert front and 2 to insert rear
```



C:\Users\sohan\Desktop\C Programs\Data Structures Lab\linked list sort rev concatenation\Linked\_list\_sort\_rev\_concat.exe

```
1
enter the item at front-end
33

press 1 to insert front and 2 to insert rear
2
enter the item at rear-end
22

press 1 to insert front and 2 to insert rear
1
enter the item at front-end
11

1:insert_front
2: delete_front
3: insert_rear
4: delete_rear
5: random_position
6: reverse
7: sort
8: concate
9: display_list
10: exit
enter the choice
7
press 1 for ascending sort and 2 for descending sort:
2

1:insert_front
2: delete_front
3: insert_rear
4: delete_rear
5: random_position
6: reverse
7: sort
8: concate
9: display_list
10: exit
enter the choice
9
22
5
4
3
2

1:insert_front
2: delete_front
3: insert_rear
```



09:20  
ENG 06-12-2020 3

C:\Users\sohan\Desktop\C Programs\Data Structures Lab\linked list sort rev concatenation\Linked\_list\_sort\_rev\_concat.exe

```
10: exit
enter the choice
7
press 1 for ascending sort and 2 for descending sort:
2
```

```
1:insert_front
2: delete_front
3: insert_rear
4: delete_rear
5: random_position
6: reverse
7: sort
8: concat
9: display_list
10: exit
```

enter the choice

```
9
22
5
4
3
2
```

```
1:insert_front
2: delete_front
3: insert_rear
4: delete_rear
5: random_position
6: reverse
7: sort
8: concat
9: display_list
10: exit
```

enter the choice

```
10
```

```
-----  
Process exited after 54.1 seconds with return value 0  
Press any key to continue . . .
```



09:20  
ENG 06-12-2020 3