

Practise program:-

Write a program to create a singly linked list and implement :-

- a) insert front ; b) delete rear ; c) display , d) count items, e)
- Search item if present

A).

```
#include <stdio.h>
#include <conio.h>
#include <malloc.h>
#include <process.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 \n");
```

```
exit(0);
```

```
}
```

```
return x;
```

```
};
```

```
int freenode(NODE x) {
```

```
free(x);
```

```
return 0;
```

```
}
```

```
NODE insert_front(NODE first, int item) {
```

```
NODE temp;
```

```
temp = getnode();
```

```
temp->info = item;
```

```
temp->link = NULL;
```

```
return temp;
temp->link = first;
first = temp;
return first;
```

```
}
```

```
NODE delete_start(NODE first)
```

```
NODE cur, precur;
```

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

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

```
return first;
```

```
}
```

```
if(first->link == NULL){
```

```
printf("item deleted is : %d", first->info);
```

```
free(first);
```

```
return NULL;
```

```
}
```

```
precur = NULL;
```

```
cur = first;
```

```
while(cur->link != NULL){
```

```
precur = cur;
```

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

```
}
```

```
printf("item deleted at rear-end is : %d", cur->info);
```

```
free(cur);
```

```
precur->link = NULL;
```

```
return first;
```

```
}
```

```
void display(NODE first)
```

```
NODE temp;
```

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

```
printf("list empty cannot display items \n");
```

```
for(temp = first; temp != NULL; temp = temp->link){
```

```
printf("%d \n", temp->info);
```

```
}
```

```
void Search(int item1, NODE first) {
    NODE cur;
    if (first == NULL) {
        printf("list is empty\n");
        return;
    }
    cur = first;
    while (cur != NULL) {
        if (item1 == cur->info) printf("search successful\n");
        cur = cur->link;
    }
    if (cur == NULL) printf("search is unsuccessful\n");
}
```

```
NODE asc(NODE first) {
    NODE prev = first;
    NODE cur = NULL;
    int temp;
    if (first == NULL)
        return 0;
    else {
        while (prev != NULL) {
            cur = prev->link;
            while (cur != NULL) {
                if (prev->info > cur->info) {
                    temp = prev->info;
                    prev->info = cur->info;
                    cur->info = temp;
                }
                cur = cur->link;
            }
            prev = prev->link;
        }
        return first;
    }
}
```

```
NODE des(NODE first){  
    NODE prev = first;  
    NODE cur = NULL;  
    int temp;  
    if (first == NULL){  
        return 0;  
    }  
    else{  
        while (prev != NULL){  
            cur = prev -> link;  
            while (cur != NULL){  
                if (prev -> info < cur -> info){  
                    temp = prev -> info;  
                    prev -> info = cur -> info;  
                    cur -> info = temp;  
                }  
                cur = cur -> link;  
            }  
            prev = prev -> link;  
        }  
        return first;  
    }  
}
```

```
int length(NODE first){  
    NODE cur;  
    int count = 0;  
    if (first == NULL)  
        return 0;  
    cur = first;  
    while (cur != NULL)  
        count++;  
    cur = cur -> link;  
    return count;  
}
```

```
int main(){  
    int item, choice, pos, option, item1, count;  
    system("cls");
```

NODE first = NULL;

~~for~~ :

for(;;) {

printf("In 1: insert-front In 2: delete-rear In 3: sort In 4:

Search In 5: display list In 6: length of list In 7: exit \n"),

scanf("%d", &choice);

scanf("r.d", &choice);

switch(choice) {

case 1: printf("enter the item at front-end In ");

scanf("r.d", &item);

first = insert_front(first, item);

break;

case 2: ~~first~~ first = delete_rear(first);

break;

case 3: printf("Press 1 for ascending sort & 2 for descending
sort : \n");

scanf("r.d", &option);

if(option == 1) {

first = asc(first);

~~display(first);~~

~~if(option == 2) {~~

first = desc(first);

display(first);

}

break;

case 4: printf("enter the item to be searched ! \n");

scanf("r.d", &item);

search(item, first);

break;

case 5: display(first);

break;

case 6: count = length(first);

printf("length(items) in the list is %d \n", count);

break;

default : exit(0);
break;

}

getch();

getchar();

}

C:\Users\sohan\Desktop\C Programs\Data Structures Lab\linked list sort search display\search_sort_display_linked_list.cpp - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

(globals)

```
[*] search_sort_display_linked_list.cpp
1 #include <stdio.h>
2 #include <conio.h>
3 #include <malloc.h>
4 #include <process.h>
5 struct node
6 {
7     int info;
8     struct node *link;
9 };
10 typedef struct node *NODE;
11 NODE get_node()
12 {
13     NODE x;
14     x=(NODE) malloc(sizeof(struct node));
15     if(x==NULL)
16     {
17         printf("mem full\n");
18         exit(0);
19     }
20     return x;
21 }
22 int freenode(NODE x)
23 {
24     free(x);
25     return 0;
26 }
27 NODE insert_front(NODE first, int item)
28 {
29     NODE temp;
30     temp=get_node();
31     temp->info=item;
32     temp->link=NULL;
33     if(first==NULL)
34         return temp;
35     temp->link=first;
36     first=temp;
37     return first;
38 }
```

Compiler Resources Compile Log Debug Find Results

Line: 153 Col: 1 Sel: 0 Lines: 188 Length: 3132 Insert Done parsing in 0.016 seconds

Search

Windows Start File Explorer Mail Settings Task View Chrome Edge Word DEV

16:33 ENG 07-12-2020

C:\Users\sohan\Desktop\C Programs\Data Structures Lab\linked list sort search display\search_sort_display_linked_list.cpp - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

(globals)

[*] search_sort_display_linked_list.cpp

```
38 L }
39 NODE delete_rear(NODE first)
40 {
41     NODE cur, prev;
42     if(first==NULL)
43     {
44         printf("list is empty cannot delete\n");
45         return first;
46     }
47     if(first->link==NULL)
48     {
49         printf("item deleted is %d\n", first->info);
50         free(first);
51         return NULL;
52     }
53     prev=NULL;
54     cur=first;
55     while(cur->link!=NULL)
56     {
57         prev=cur;
58         cur=cur->link;
59     }
60     printf("item deleted at rear-end is %d", cur->info);
61     free(cur);
62     prev->link=NULL;
63
64     return first;
65 }
66 void display(NODE first)
67 {
68     NODE temp;
69     if(first==NULL)
70     {
71         printf("list empty cannot display items\n");
72     }
73     for(temp=first; temp!=NULL; temp=temp->link)
74     {
75         printf("%d\n", temp->info);
76     }
77 }
```

Compiler Resources Compile Log Debug Find Results

Line: 153 Col: 1 Sel: 0 Lines: 188 Length: 3132 Insert Done parsing in 0.016 seconds

Search

Windows Start File Explorer Mail Settings Task View Chrome Edge Word DEV

16:34 ENG 07-12-2020

C:\Users\sohan\Desktop\C Programs\Data Structures Lab\linked list sort search display\search_sort_display_linked_list.cpp - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

(globals)

```
[*] search_sort_display_linked_list.cpp
74 }
75 }
76 void search(int item, NODE first){
77     NODE cur;
78     if(first==NULL){
79         printf("list is empty \n");
80         return;
81     }
82     cur=first;
83     while(cur!=NULL){
84         if(item==cur->info) printf("search successful "); break;
85         cur=cur->link;
86     }
87     if(cur==NULL){
88         printf("search is unsuccessful \n");
89     }
90 }
91 }
92 NODE asc(NODE first){
93     NODE prev=first;
94     NODE cur=NULL;
95     int temp;
96     if(first==NULL){
97         return 0;
98     }
99     else{
100    while(prev!=NULL){
101        cur=prev->link;
102        while(cur!=NULL){
103            if(prev->info > cur->info){
104                temp=prev->info;
105                prev->info=cur->info;
106                cur->info=temp;
107            }
108            cur=cur->link;
109        }
110        prev=prev->link;
111    }
112 }
```

Compiler Resources Compile Log Debug Find Results

Line: 153 Col: 1 Sel: 0 Lines: 188 Length: 3132 Insert Done parsing in 0.016 seconds

Search

16:34 ENG 07-12-2020

C:\Users\sohan\Desktop\C Programs\Data Structures Lab\linked list sort search display\search_sort_display_linked_list.cpp - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

(globals)

```
[*] search_sort_display_linked_list.cpp
111 }
112 }
113 return first;
114 }
115 NODE des(NODE first){
116     NODE prev=first;
117     NODE cur=NULL;
118     int temp;
119     if(first==NULL){
120         return 0;
121     }
122     else{
123         while(prev!=NULL){
124             cur=prev->link;
125             while(cur!=NULL){
126                 if(prev->info < cur->info){
127                     temp=prev->info;
128                     prev->info=cur->info;
129                     cur->info=temp;
130                 }
131                 cur=cur->link;
132             }
133             prev=prev->link;
134         }
135     return first;
136 }
137 int length(NODE first){
138     NODE cur;
139     int count=0;
140     if(first==NULL)
141         return 0;
142     cur=first;
143     while(cur!=NULL){
144         count++;
145         cur=cur->link;
146     }
147     return count;
148 }
```

Compiler Resources Compile Log Debug Find Results

Line: 153 Col: 1 Sel: 0 Lines: 188 Length: 3132 Insert Done parsing in 0.016 seconds

Search

16:34 ENG 07-12-2020

C:\Users\sohan\Desktop\C Programs\Data Structures Lab\linked list sort search display\search_sort_display_linked_list.cpp - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

(globals)

```
[*] search_sort_display_linked_list.cpp
147     return count;
148 }
149 int main(){
150     int item_choice, pos, option, item1, count;
151     system("cls");
152     NODE first=NULL;
153     for(;;){
154         printf("\n 1:insert_front \n 2: delete_rear \n 3: sort \n 4: search \n 5: display_list \n 6:length of the list \n 7: exit \n");
155         printf("enter the choice \n");
156         scanf ("%d", &choice);
157         switch(choice){
158             case 1: printf(" enter the item at front-end \n");
159             scanf ("%d", &item);
160             first=insert_front(first,item);
161             break;
162             case 2: first=delete_rear(first);
163             break;
164             case 3: printf("press 1 for ascending sort and 2 for descending sort: \n");
165             scanf ("%d", &option);
166             if(option==1){
167                 first=asc(first);
168                 display(first);}
169             if(option==2){
170                 first=des(first);
171                 display(first);}
172             break;
173             case 4: printf("enter the item to be searched: \n");
174             scanf ("%d", &item1);
175             search(item1,first);
176             break;
177             case 5: display(first);
178             break;
179             case 6: count=length(first);
180             printf("length(items) in the list is %d \n", count);
181             break;
182             default: exit(0);
183             break;
184         }
185     }
}
```

Compiler Resources Compile Log Debug Find Results

Line: 153 Col: 1 Sel: 0 Lines: 188 Length: 3132 Insert Done parsing in 0.016 seconds

Search

Windows Start File Explorer Mail Settings Task View Chrome Edge Word DEV

16:34 ENG 07-12-2020



[*] search_sort_display_linked_list.cpp

```
151 system("cls");
152 NODE first=NULL;
153 for(;;){
154     printf("\n 1:insert_front \n 2: delete_rear \n 3: sort \n 4: search \n 5: display_list \n 6:length of the list \n 7: exit \n");
155     printf("enter the choice \n");
156     scanf ("%d", &choice);
157     switch(choice){
158         case 1: printf("enter the item at front-end \n");
159         scanf ("%d", &item);
160         first=insert_front(first,item);
161         break;
162         case 2: first=delete_rear(first);
163         break;
164         case 3: printf("press 1 for ascending sort and 2 for descending sort: \n");
165         scanf ("%d", &option);
166         if(option==1){
167             first=asc(first);
168             display(first);}
169         if(option==2){
170             first=des(first);
171             display(first);}
172         break;
173         case 4: printf("enter the item to be searched: \n");
174         scanf ("%d", &item1);
175         search(item1,first);
176         break;
177         case 5: display(first);
178         break;
179         case 6: count=length(first);
180         printf("length(items) in the list is %d \n", count);
181         break;
182         default: exit(0);
183         break;
184     }
185     getch();
186     return 0;
187 }
```

Compiler Resources Compile Log Debug Find Results

Line: 153 Col: 1 Sel: 0 Lines: 188 Length: 3132 Insert

Done parsing in 0.016 seconds



Search



16:34 ENG 07-12-2020

C:\Users\sohan\Desktop\C Programs\Data Structures Lab\linked list sort search display\search_sort_display_linked_list.exe

```
1:insert_front
2: delete_rear
3: sort
4: search
5: display_list
6:length of the list
7: exit
enter the choice
1
enter the item at front-end
23

1:insert_front
2: delete_rear
3: sort
4: search
5: display_list
6:length of the list
7: exit
enter the choice
1
enter the item at front-end
65

1:insert_front
2: delete_rear
3: sort
4: search
5: display_list
6:length of the list
7: exit
enter the choice
1
enter the item at front-end
77

1:insert_front
2: delete_rear
3: sort
4: search
5: display_list
6:length of the list
7: exit
enter the choice
1
enter the item at front-end
3

1:insert_front
```



16:38 ENG 07-12-2020 3

C:\Users\sohan\Desktop\C Programs\Data Structures Lab\linked list sort search display\search_sort_display_linked_list.exe

```
6:length of the list
7: exit
enter the choice
1
enter the item at front-end
3

1:insert_front
2: delete_rear
3: sort
4: search
5: display_list
6:length of the list
7: exit
enter the choice
1
enter the item at front-end
100
```

```
1:insert_front
2: delete_rear
3: sort
4: search
5: display_list
6:length of the list
7: exit
enter the choice
4
```

enter the item to be searched:

100

search successful

```
1:insert_front
2: delete_rear
3: sort
4: search
5: display_list
6:length of the list
7: exit
enter the choice
```

2

item deleted at rear-end is 23

```
1:insert_front
2: delete_rear
3: sort
4: search
5: display_list
6:length of the list
7: exit
enter the choice
```

5



C:\Users\sohan\Desktop\C Programs\Data Structures Lab\linked list sort search display\search_sort_display_linked_list.exe

6:length of the list

7: exit

enter the choice

5

100

3

77

65

1:insert_front

2: delete_rear

3: sort

4: search

5: display_list

6:length of the list

7: exit

enter the choice

6

length(items) in the list is 4

1:insert_front

2: delete_rear

3: sort

4: search

5: display_list

6:length of the list

7: exit

enter the choice

3

press 1 for ascending sort and 2 for descending sort:

2

100

77

65

3

1:insert_front

2: delete_rear

3: sort

4: search

5: display_list

6:length of the list

7: exit

enter the choice

7

Process exited after 49.53 seconds with return value 0

Press any key to continue . . .



Search



16:39
ENG
07-12-2020