

## Ques Program 6 :-

- Ques Write a program implement Singly linked list with following question operation :
- Create a linked list.
  - Deletion of first element, Specified element and last element in the list.
  - Display the contents of the linked list.

```
#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 = (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 insertFront (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 delete_start (NODE first) {`

`NODE temp;`

`if (first == NULL) {`

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

`return first;`

`}`

`NODE delete_end (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);`

`prev->link = NULL;`

`return first;`

`}`

`void display (NODE first) {`

`NODE temp;`

`if (first == NULL)`

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

{}

```
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;
```

```
free(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;
```

{}

```
if (count == pos) {
```

```
printf ("invalid position specified\n");
```

```
return first;
```

{}

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

```
greenode (cur);
```

```
return first;
```

{

```
int main()
```

```
int item, choice, pos;
```

```
NODE first = NULL;
```

```
system("cls");
```

```
for(;;)
```

```
printf("In 1: Insert-front In 2: Delete-front In 3: Insert-rear In  
4: Delete-rear In 5: Delete at specified position. In 6: Display-list  
In 7: Exit In ");
```

```
printf("enter the choice In ");
```

```
scanf("%d", &choice);
```

```
switch(choice){
```

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

```
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 In");
```

```
scanf("%d", &item);
```

```
first=insert-rear(first, item);
```

```
break;
```

```
case 4: first=delete-rear(first);
```

```
break;
```

```
case 5: printf("enter the position of the item to be deleted : In");
```

```
scanf("%d", &pos);
```

```
first=delete-pos(pos, first);
```

```
break;
```

```
case 6: display(first);
```

```
break;
```

```
default: exit(0); break; if getch(); return 0; }
```

C:\Users\sohan\Desktop\C Programs\Data Structures Lab\singly linked list delete\singly\_linked\_lists\_deletion.cpp - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

(globals)

```
[*] singly_linked_lists_deletion.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;
}
```

Compiler Resources Compile Log Debug Find Results

Line: 149 Col: 15 Sel: 0 Lines: 182 Length: 3003 Insert Done parsing in 0.015 seconds

Search

Windows Start Task View File Explorer Mail Settings Taskbar Icons Word DEV

21:49 ENG 05-12-2020 3

C:\Users\sohan\Desktop\C Programs\Data Structures Lab\singly linked list delete\singly\_linked\_lists\_deletion.cpp - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

[\*] singly\_linked\_lists\_deletion.cpp

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

Compiler Resources Compile Log Debug Find Results

Line: 149 Col: 15 Sel: 0 Lines: 182 Length: 3003 Insert Done parsing in 0.015 seconds

21:49 05-12-2020

C:\Users\sohan\Desktop\C Programs\Data Structures Lab\singly linked list delete\singly\_linked\_lists\_deletion.cpp - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

[\*] singly\_linked\_lists\_deletion.cpp

```
73     return first;
74 }
75 if(first->link==NULL)
76 {
77     printf("item deleted is %d\n", first->info);
78     free(first);
79     return NULL;
80 }
81 prev=NULL;
82 cur=first;
83 while(cur->link!=NULL)
84 {
85     prev=cur;
86     cur=cur->link;
87 }
88 printf("item deleted at rear-end is %d", cur->info);
89 free(cur);
90 prev->link=NULL;
91
92 return first;
93 }
94 void display(NODE first)
95 {
96     NODE temp;
97     if(first==NULL)
98         printf("list empty cannot display items\n");
99     for(temp=first; temp!=NULL; temp=temp->link)
100    {
101        printf("%d\n", temp->info);
102    }
103 }
104 NODE delete_pos(int pos, NODE first)
105 {
106     NODE cur;
107     NODE prev;
108     int count;
109     if(first==NULL || pos<=0)
```

Compiler Resources Compile Log Debug Find Results

Line: 149 Col: 15 Sel: 0 Lines: 182 Length: 3003 Insert Done parsing in 0.015 seconds

21:49 05-12-2020 ENG

C:\Users\sohan\Desktop\C Programs\Data Structures Lab\singly linked list delete\singly\_linked\_lists\_deletion.cpp - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

[\*] singly\_linked\_lists\_deletion.cpp

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

Compiler Resources Compile Log Debug Find Results

Line: 149 Col: 15 Sel: 0 Lines: 182 Length: 3003 Insert Done parsing in 0.015 seconds

Search

21:49 05-12-2020 ENG

C:\Users\sohan\Desktop\C Programs\Data Structures Lab\singly linked list delete\singly\_linked\_lists\_deletion.cpp - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

(globals)

```
[*] singly_linked_lists_deletion.cpp
143     return first;
144 }
145 int main()
146 {
147     int item_choice, pos;
148     NODE first=NULL;
149     system("cls");
150     for(;;)
151     {
152         printf("\n 1:Insert_front\n 2:Delete_front\n 3:Insert_rear\n 4:Delete_rear\n 5.Delete at specified position \n 6:Display_list\n 7:Ex
153         printf("enter the choice\n");
154         scanf ("%d", &choice);
155         switch(choice)
156         {
157             case 1:printf("enter the item at front-end\n");
158             scanf ("%d", &item);
159             first=insert_front(first,item);
160             break;
161             case 2:first=delete_front(first);
162             break;
163             case 3:printf("enter the item at rear-end\n");
164             scanf ("%d", &item);
165             first=insert_rear(first,item);
166             break;
167             case 4:first=delete_rear(first);
168             break;
169             case 5:printf("enter the position of the item to be deleted: \n");
170             scanf ("%d", &pos);
171             first=delete_pos(pos,first);
172             break;
173
174             case 6:display(first);
175             break;
176             default:exit(0);
177             break;
178         }
179     }
```

Compiler Resources Compile Log Debug Find Results

Line: 149 Col: 15 Sel: 0 Lines: 182 Length: 3003 Insert Done parsing in 0.015 seconds

Search

21:49 05-12-2020 ENG

C:\Users\sohan\Desktop\C Programs\Data Structures Lab\singly linked list delete\singly\_linked\_lists\_deletion.cpp - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

(globals)

```
[*] singly_linked_lists_deletion.cpp
147 int item_choice, pos;
148 NODE first=NULL;
149 system("cls");
150 for(;;)
151 {
152     printf("\n 1: Insert_front\n 2: Delete_front\n 3: Insert_rear\n 4: Delete_rear\n 5. Delete at specified position \n 6: Display_list\n 7: Exit");
153     printf(" enter the choice\n");
154     scanf("%d", &choice);
155     switch(choice)
156     {
157         case 1: printf("enter the item at front-end\n");
158         scanf ("%d", &item);
159         first=insert_front(first, item);
160         break;
161         case 2: first=delete_front(first);
162         break;
163         case 3: printf("enter the item at rear-end\n");
164         scanf ("%d", &item);
165         first=insert_rear(first, item);
166         break;
167         case 4: first=delete_rear(first);
168         break;
169         case 5: printf("enter the position of the item to be deleted: \n");
170             scanf ("%d", &pos);
171             first=delete_pos(pos, first);
172             break;
173
174         case 6: display(first);
175         break;
176         default: exit(0);
177         break;
178     }
179 }
180 getch();
181 return 0;
182 }
```

Compiler Resources Compile Log Debug Find Results

Line: 149 Col: 15 Sel: 0 Lines: 182 Length: 3003 Insert Done parsing in 0.015 seconds

Search

21:49 05-12-2020 ENG

C:\Users\sohan\Desktop\C Programs\Data Structures Lab\singly linked list delete\singly\_linked\_lists\_deletion.exe

```
1:Insert_front
2:Delete_front
3:Insert_rear
4:Delete_rear
5.Delete at specified position
6:Display_list
6:Exit
enter the choice
1
enter the item at front-end
56

1:Insert_front
2:Delete front
3:Insert_rear
4:Delete_rear
5.Delete at specified position
6:Display_list
6:Exit
enter the choice
3
enter the item at rear-end
69

1:Insert_front
2:Delete_front
3:Insert_rear
4:Delete_rear
5.Delete at specified position
6:Display_list
6:Exit
enter the choice
1
enter the item at front-end
4

1:Insert_front
2:Delete_front
3:Insert_rear
4:Delete_rear
5.Delete at specified position
6:Display_list
6:Exit
enter the choice
3
enter the item at rear-end
76

1:Insert_front
```



C:\Users\sohan\Desktop\C Programs\Data Structures Lab\singly linked list delete\singly\_linked\_lists\_deletion.exe

enter the item at rear-end

76

```
1:Insert_front  
2:Delete_front  
3:Insert_rear  
4:Delete_rear  
5.Delete at specified position  
6:Display_list  
6:Exit
```

enter the choice

5

enter the position of the item to be deleted:

3

```
1:Insert_front  
2:Delete_front  
3:Insert_rear  
4:Delete_rear  
5.Delete at specified position  
6:Display_list  
6:Exit
```

enter the choice

1

enter the item at front-end

8

```
1:Insert_front  
2:Delete_front  
3:Insert_rear  
4:Delete_rear  
5.Delete at specified position  
6:Display_list  
6:Exit
```

enter the choice

3

enter the item at rear-end

9

```
1:Insert_front  
2:Delete_front  
3:Insert_rear  
4:Delete_rear  
5.Delete at specified position  
6:Display_list  
6:Exit
```

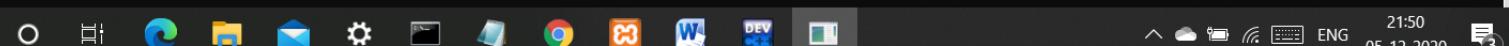
enter the choice

2

item deleted at front-end is=8



Search



21:50  
ENG  
05-12-2020

C:\Users\sohan\Desktop\C Programs\Data Structures Lab\singly linked list delete\singly\_linked\_lists\_deletion.exe

enter the item at rear-end

9

```
1:Insert_front  
2:Delete_front  
3:Insert_rear  
4:Delete_rear  
5.Delete at specified position  
6:Display_list  
6:Exit  
enter the choice  
2  
item deleted at front-end is=8
```

```
1:Insert_front  
2:Delete_front  
3:Insert_rear  
4:Delete_rear  
5.Delete at specified position  
6:Display_list  
6:Exit  
enter the choice  
4  
item deleted at rear-end is 9  
1:Insert_front  
2:Delete_front  
3:Insert_rear  
4:Delete_rear  
5.Delete at specified position  
6:Display_list  
6:Exit  
enter the choice  
6  
4  
56  
76
```

```
1:Insert_front  
2:Delete_front  
3:Insert_rear  
4:Delete_rear  
5.Delete at specified position  
6:Display_list  
6:Exit  
enter the choice  
7
```

Process exited after 280 seconds with return value 0

Press any key to continue . . .

