

create a singly linked list and implement insert front, insert rear, insert at a particular position and display function

main.c

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 struct node
4 {
5     int info;
6     struct node *link;
7 };
8
9 typedef struct node *NODE;
10 NODE getnode()
11 {
12     NODE x;
13     x=(NODE)malloc(sizeof(struct node));
14     if(x==NULL)
15     {
16         printf("Memory is full\n");
17         exit(0);
18     }
19     return x;
20 }
21
22 void freenode(NODE x)
23 {
24     free(x);
25 }
26
27 NODE insert_front(NODE first,int item)
```

input

main.c

```
27 NODE insert_front(NODE first,int item)
28 {
29     NODE temp;
30     temp=getnode();
31     temp->info=item;
32     temp->link=NULL;
33     if(first==NULL)
34     {
35         return temp;
36     }
37     temp->link=first;
38     first=temp;
39     return first;
40 }
41
42 NODE insert_rear(NODE first ,int item)
43 {
44     NODE temp,cur;
45     temp=getnode();
46     temp->info=item;
47     temp->link=NULL;
48     if(first==NULL)
49         return temp;
50     cur=first;
51     while(cur->link!=NULL)
52         cur=cur->link;
53     cur->link=temp;
```

input

main.c

```
53     cur->link=temp;
54     return first;
55 }
56
57 NODE insert_pos(int item, int pos ,NODE first)
58 {
59     NODE temp;
60     NODE prev,cur;
61     int count;
62     temp=getnode();
63     temp->info=item;
64     temp->link=NULL;
65     if(first==NULL && pos==1)
66         return temp;
67     if(first==NULL)
68     {
69         printf("Invalid Position\n");
70         return first;
71     }
72     if(pos==1)
73     {
74         temp->link=first;
75         return temp;
76     }
77     count=1;
78     prev=NULL;
79     cur=first;
```

input

main.c

```
77 count=1;
78 prev=NULL;
79 cur=first;
80 while(cur!=NULL && count!=pos)
81 {
82     prev=cur;
83     cur=cur->link;
84     count++;
85 }
86 if(count==pos)
87 {
88     prev->link=temp;
89     temp->link=cur;
90     return first;
91 }
92 printf("IP\n");
93 return first;
94 }
95
96 void display(NODE first)
97 {
98     NODE temp;
99     if(first==NULL)
100         printf("List is EMPTY , Cannot Display Items\n");
101     printf("\n*****\n");
102     for(temp=first;temp!=NULL;temp=temp->link)
103     {
```

input

main.c

```
96 void display(NODE first)
97 {
98     NODE temp;
99     if(first==NULL)
100         printf("List is EMPTY , Cannot Display Items\n");
101     printf("\n*****\n");
102     for(temp=first;temp!=NULL;temp=temp->link)
103     {
104         printf("%d\n",temp->info);
105     }
106     printf("\n*****\n");
107 }
108
109
110 void main()
111 {
112     int item,choice,pos;
113     NODE first=NULL;
114     for(;;)
115     {
116         printf("Enter the choice: ");
117         scanf("%d",&choice);
118         switch(choice)
119         {
120             case 1:printf("Enter the item at front-end: ");
121                     scanf("%d",&item);
122                     first=insert front(first,item);
```

input

main.c

```
112 int item,choice,pos;
113 NODE first=NULL;
114 for(;;)
115 {
116     printf("Enter the choice: ");
117     scanf("%d",&choice);
118     switch(choice)
119     {
120         case 1:printf("Enter the item at front-end: ");
121                 scanf("%d",&item);
122                 first=insert_front(first,item);
123                 break;
124         case 2:printf("Enter the item at rear-end: ");
125                 scanf("%d",&item);
126                 first=insert_rear(first,item);
127                 break;
128         case 3:printf("Enter the position: ");
129                 scanf("%d",&pos);
130                 first=insert_pos(item,pos,first);
131                 break;
132         case 4:display(first);
133                 break;
134         default:exit(0);
135                 break;
136     }
137 }
138 }
```

input

Enter the choice: 1

onlinegdb.com/online\_c\_compiler

Run Debug Stop Share Save {} Beautify

Language C

input

```
Enter the choice: 1
Enter the item at front-end: 10
Enter the choice: 2
Enter the item at rear-end: 23
Enter the choice: 1
Enter the item at front-end: 20
Enter the choice: 2
Enter the item at rear-end: 45
Enter the choice: 4
```

\*\*\*\*\*

```
20
10
23
45
```

\*\*\*\*\*

```
Enter the choice: 3
Enter the position: 2
Enter the choice: 2
Enter the item at rear-end: 256
Enter the choice: 3
Enter the position: 2
Enter the choice: 4
```

\*\*\*\*\*

```
20
256
```



```
Enter the choice: 4
*****
20
10
23
45
*****
Enter the choice: 3
Enter the position: 2
Enter the choice: 2
Enter the item at rear-end: 256
Enter the choice: 3
Enter the position: 2
Enter the choice: 4
*****
20
256
45
10
23
45
256
*****
```

