DS LAB-PROG 5-SINGLY LINKED LIST

Program and output

Mallika Prasad

1BM19CS081

18.11.2020

Program-

```
#include<stdio.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\n");
  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 delete_front(NODE first)
{
NODE temp;
if(first==NULL)
{
printf("list is empty cannot delete\n");
return first;
}
temp=first;
temp=temp->link;
```

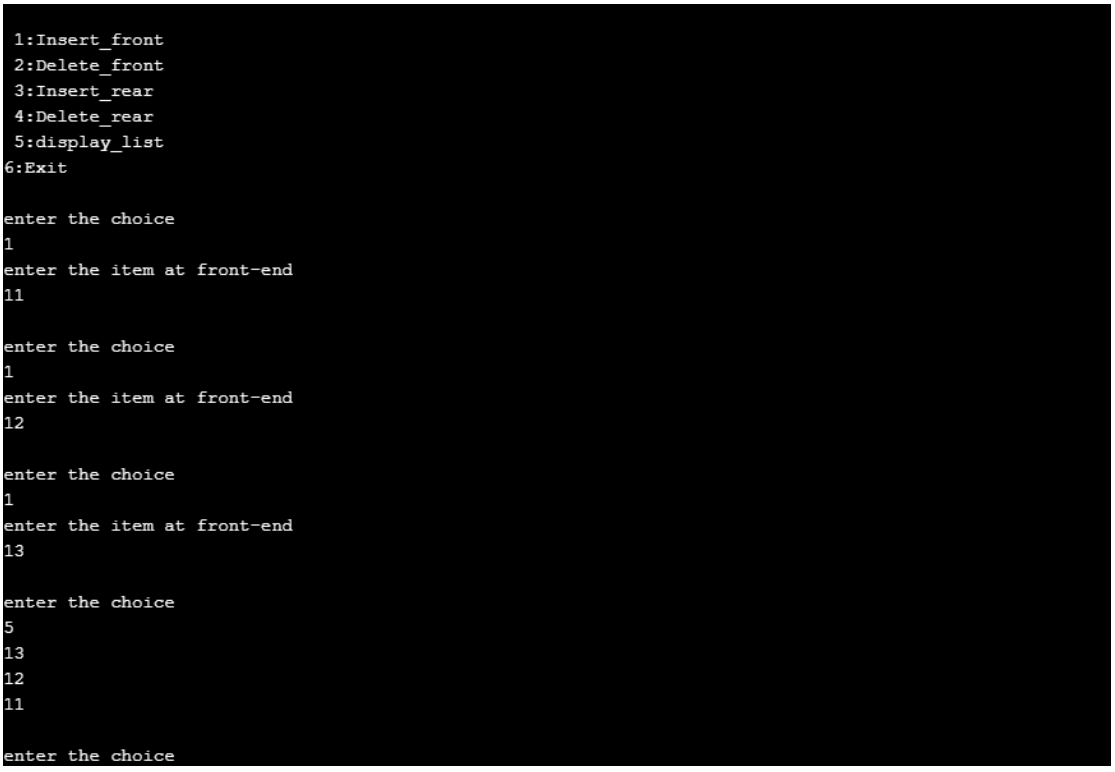
```
printf("item deleted at front-end is=%d\n",first->info);
free(first);
return temp;
}
NODE insert_rear(NODE first,int item)
{
NODE temp,cur;
temp=getnode();
temp->info=item;
temp->link=NULL;
if(first==NULL)
 return temp;
cur=first;
while(cur->link!=NULL)
 cur=cur->link;
cur->link=temp;
return first;
}
NODE 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("iten deleted at rear-end is %d",cur->info);
free(cur);
prev->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);
  }
}
int main()
{
int item,choice;
NODE first=NULL;
printf("\n 1:Insert_front\n 2:Delete_front\n 3:Insert_rear\n 4:Delete_rear\n
5:display_list\n6:Exit\n");
do
{
printf("\nenter the choice\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;
```

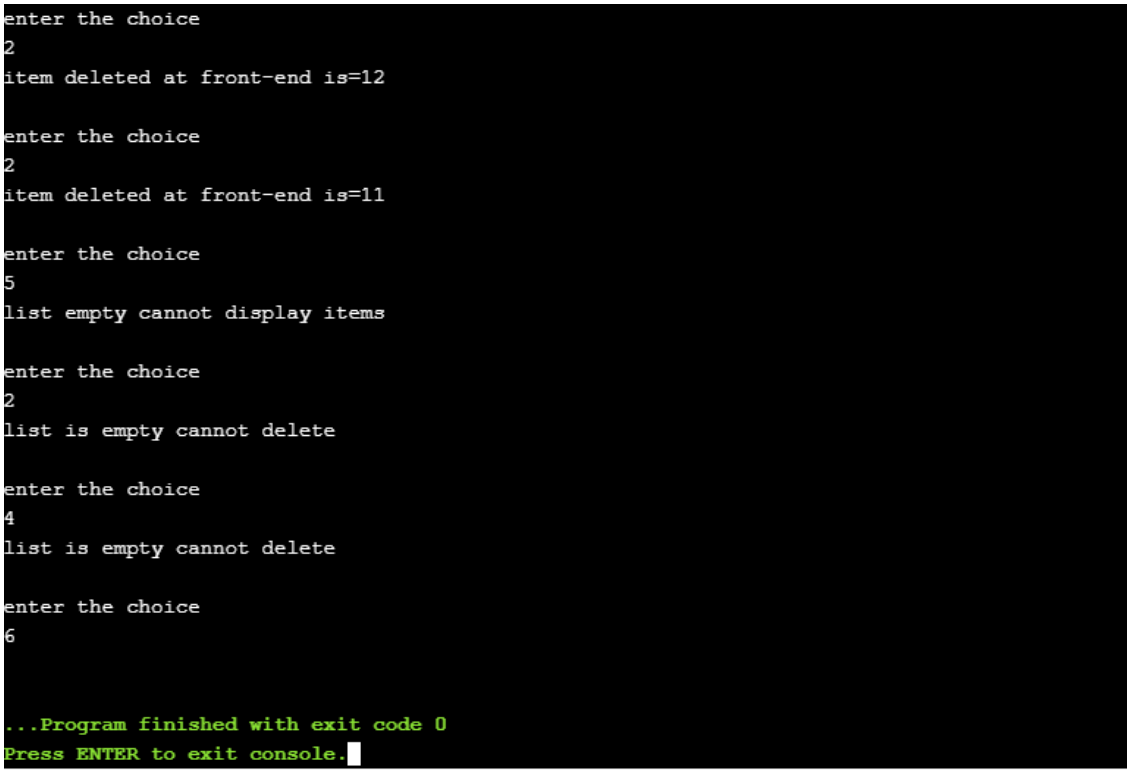
```
 case 5:display(first);
            break;
            case 6:break;
 default:break;
 }
}while(choice!=6);
return 0;
}
```

Output-



```
enter the choice
3
enter the item at rear-end
14

enter the choice
3
enter the item at rear-end
15

enter the choice
3
enter the item at rear-end
16

enter the choice
5
13
12
11
14
15
16

enter the choice
2
item deleted at front-end is=13

enter the choice
4
iten deleted at rear-end is 16
```

```
iten deleted at rear-end is 16
enter the choice
4
iten deleted at rear-end is 15
enter the choice
4
iten deleted at rear-end is 14
enter the choice
5
12
11

enter the choice
2
item deleted at front-end is=12

enter the choice
2
item deleted at front-end is=11

enter the choice
5
list empty cannot display items

enter the choice
2
list is empty cannot delete

enter the choice
4
```


enter the choice
2
item deleted at front-end is=12
enter the choice
2
item deleted at front-end is=11
enter the choice
5
list empty cannot display items
enter the choice
2
list is empty cannot delete
enter the choice
4
list is empty cannot delete
enter the choice
6
...Program finished with exit code 0
Press ENTER to exit console.