DS LAB-PROG 8-SINGLY LINKED LIST

STACK AND QUEUE

Program and output

Mallika Prasad

1BM19CS081

9.12.2020

Program 8-

```
#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_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)
```

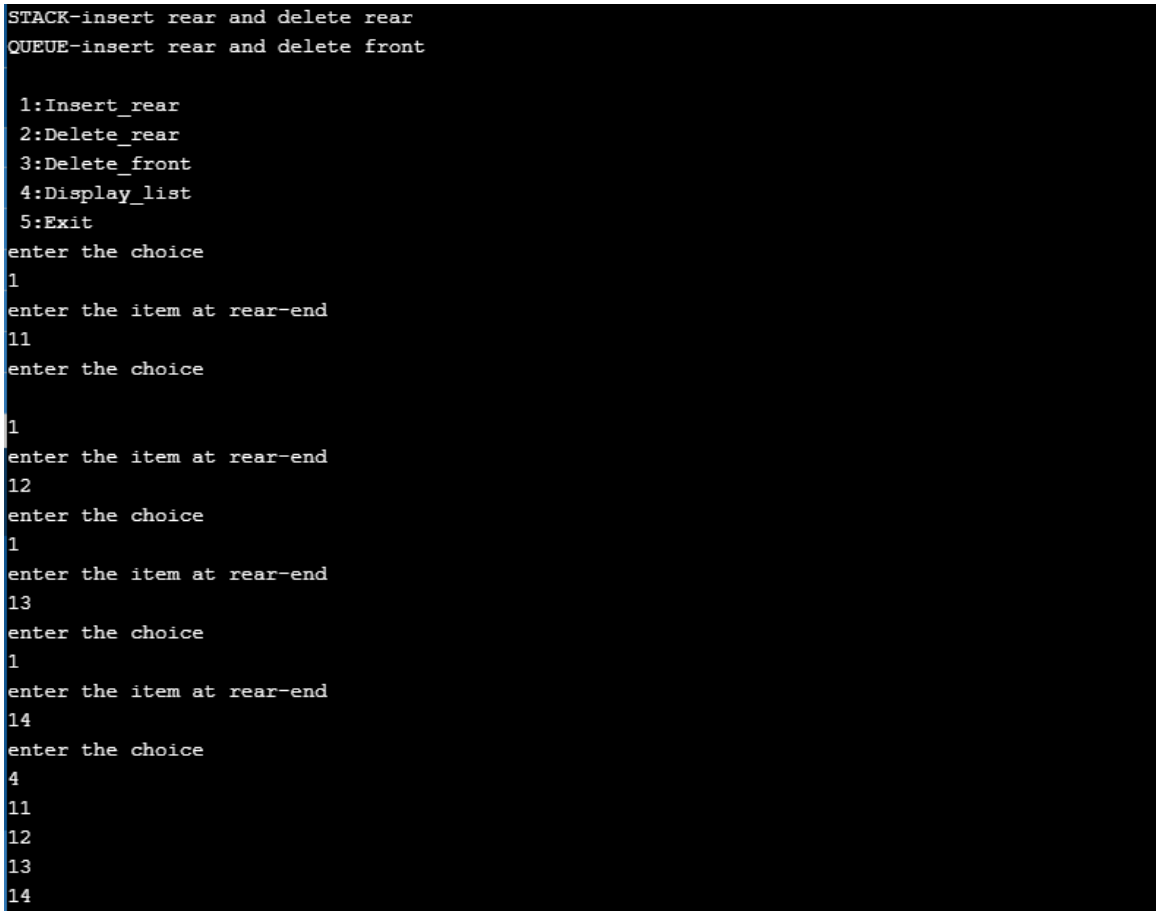
```
{
NODE cur,prev;
if(first==NULL)
{
printf("stack is empty cannot delete\n");
return first;
}
if(first->link==NULL)
{
printf("item deleted is %d\n",first->info);
free(first);
return NULL;
}
prev=NULL;
cur=first;
while(cur->link!=NULL)
{
prev=cur;
cur=cur->link;
}
printf("item deleted at rear-end is %d\n",cur->info);
free(cur);
prev->link=NULL;
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;
}

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("STACK-insert rear and delete rear\nQUEUE-insert rear and delete front\n");
printf("\n 1:Insert_rear\n 2:Delete_rear\n 3:Delete_front\n 4:Display_list\n 5:Exit\n");
do
{
printf("enter the choice\n");
scanf("%d",&choice);
switch(choice)
{
case 1:printf("enter the item at rear-end\n");
scanf("%d",&item);
first=insert_rear(first,item);
break;
case 2:first=delete_rear(first);
break;
case 3:first=delete_front(first);
```

```
break;

case 4:display(first);

break;

case 5:break;

default:printf("invalid choice");

break;

}

}while(choice!=5);

return 0;

}
```

Output-



```
13
14
enter the choice
2
item deleted at rear-end is 14
enter the choice
2
item deleted at rear-end is 13
enter the choice
4
11
12
enter the choice
2
item deleted at rear-end is 12
enter the choice
2
item deleted is 11
enter the choice
2
stack is empty cannot delete
enter the choice
1
enter the item at rear-end
22
enter the choice
1
enter the item at rear-end
33
enter the choice
1
```

```
33
enter the choice
1
enter the item at rear-end
44
enter the choice
1
enter the item at rear-end
55
enter the choice
4
22
33
44
55
enter the choice
3
item deleted at front-end is=22
enter the choice
3
item deleted at front-end is=33
enter the choice
4
44
55
enter the choice
3
item deleted at front-end is=44
enter the choice
3
item deleted at front-end is=55
```

```
4
22
33
44
55
enter the choice
3
item deleted at front-end is=22
enter the choice
3
item deleted at front-end is=33
enter the choice
4
44
55
enter the choice
3
item deleted at front-end is=44
enter the choice
3
item deleted at front-end is=55
enter the choice
3
list is empty cannot delete
enter the choice
5


...Program finished with exit code 0
Press ENTER to exit console.
```