Lab-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("Memory 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 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_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
free(first);
return temp;
}
NODE delete_rear(NODE first)
{
  NODE cur, Prev;
  if (first==NULL)
  {
   Printf("List is empty cannot delete\n");
   return first;
  }
  if (first->link=NULL)
  {
   Print f("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);
Rev->link=NULL;
return first;
}
void display (NODE first)
{
  NODE temp;
  if (first==NULL)
  {
    Printf("List empty cannot display items\n");
    return;
  }
  Printf("Contents of list:\n");
  for(temp=first; temp!=NULL; temp=temp->link)
  {
    Printf("%d\n", temp->info);
  }
}
int main()
{
  int item, choice;
  NODE first=NULL, a, b;
```

```
for(;;)
{
Printf("\n1:Stack\n2:Queue\n3:Exit\n");
Printf("Enter the choice\n");
Scanf("%d",&choice);
Switch(choice)
{
  case 1: Printf("Stack\n");
          for(;;)
        {
         Printf("\n1:Insert_rear\n2:Delete_rear
             \n3:Display_list\n4:Exit\n");
         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: display(first);
                 break;
        default: exit(0);
                 break;
   }
}
```

```
case 2: Printf("QUEUE\n");
        for(;;)
        {
          Printf("\n1: Insert_rear\n2: Delete_front\n
                  3: Display_List\n4: Exit\n");
          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_front(first);
                    break;

            case 3: display (first);
                    break;

            default: exit(0);
                     break;
          }
        }

case 3: exit(0);
default: Printf("Invalid choice\n");
}
}
return 0;
}
```