PAGE NO.
DATE / /

Data Structures - Lab 4

1. Circular Queue

```
#include <stdio.h>
#include <stdlib.h>
#include <process.h>
#define que_size 3
int item, front = 0, rear = -1, q[que_size], count = 0;
Void insertrear()
{
  if (count == que_size)
{
printf("queue overflow.");
return;
}
rear = (rear + 1) % que_size;
    q[rear] = item;
    count ++;
  }
  int deletefront()
{
if (count == 0)
   return -1;
item = q[front];
front = (front + 1) % que_size;
   count = count - 1;
   return item;
  }
  void displayq()
{
  int i, f;
   if (count == 0)
```

```
{
  printf ("queue is empty");
  return;
}
  f = front;
printf ("contents of queue \n");
  for (i=0; i<count; i++)
{
printf ("%d \n", q[f]);
  f = (f+1) % que_size;
}
}
  Void main ()
{
int choice;
  for (;;)
{
printf ("\n 1. Insert rear \n 2. delete front \n 3. Display
        \n 4. exit \n");
  printf ("enter the choice:");
  scanf ("%d", &choice);
 Switch (choice)
{
Case 1: printf ("enter the item to be inserted:");
  scanf ("%d", &item); insertrear();
      break;
 Case 2: item = delete front ();
    if (item == -1)
  printf ("queue is empty \n");
else printf ("item deleted is %d \n", item);
 break; case 3: display q ();
  break; default: exit (0); } }
    getch (); }
```

PAGE NO.
DATE / /

Double ended Queue

```
#include <stdio.h>
# include <stdlib.h>
# include <process.h>
# define que_size 5
int f=0, r=-1, ch;
int item, q[10];
int isfull()
{
return (r==q size-1)? 1:0;
}
int isempty()
{
return (f>r)?1:0;
}
void insest_rear()
{
if (isfull())
{
printf("queue Overflow \n");
return;
}
r=r+1;
q[r]=item; }
void delete_front()
{
if (is empty())
{ printf("queue is empty \n");
return; }
printf("item deleted is %d \n", q[(f)++]);
if (f>r)
```

```
{ f=0; r=-1;
} }
void insert_front()
{ if (f!=0)
  { f=f-1;
    q[f] = item;
    return;
  }
  else if ((f==0)&&(r==-1))
  { q[++(r)] = item;
    return;
  }
  else printf("Insertion not possible \n");
}
void delete_rear()
  {
  if (isempty())
  { printf("queue is empty \n");
return;}
printf("item deleted is %d \n", q[(r)--1]);
    if (f>r)
  { f=0; r=-1; } }
void display()
{ int i;
  if (isempty())
{ printf("queue empty \n");
    return; }
    for(i=f; i<=r; i++)
    printf("%d \n", q[i]);
}
  void main()
{
```

PAGE NO.
DATE / /

```
for (;;)
{
printf ("1. insert_rear \n 2. insert front \n 3.
delete_rear \n 4. delete front \n 5 display \n 6. exit
\n ");
printf (" enter choice \n ");
    scanf (" %d ", &ch);
  switch (ch)
{
  case 1: printf (" enter the item \n ");
      scanf (" %d ", & item);
       insert_rear ();
      break;
  case 2: printf (" enter the item \n ");
      scanf (" %d ", & item);
       insert_front ();
      break;
  case 3 : delete_rear ();
      break;
  case 4: delete_front ();
      break;
  case 5: display ();
      break;
  default : exit (0);
  }
  }
  }
```