Name S Skanda
USN : 1BM19CS137
Batch 2
Date: 19/10/2020

1) Circular Queue

```
#include <stdio.h>
#include <process.h>
#define qs 5
int item, front=0, rear = -1, q[10], count=0;
void insert()
{  if(count==qs)
   { printf("The queue is full\n");
     return;
   }
   rear=(rear+1)%qs;
   q[rear]=item;
   count+=1;
}
int delete()
{ if (count==0)
   {  front=0; rear=-1;
      return -1;
   }
   item = q[front];
   front=(front+1)%qs;
   count-=1;
   return item;
}
```

```
void display()
{ int i, f=front;
  if (count==0)
  { printf("The queue is empty");
    return;
  }
  for(i=1; i<=count; i++)
  { printf("%d " q[f]);
    f=(f+1)%qs;
  }
}
int main()
{ int choice()
  for(;;)
  { printf("Menu\n 1.INSERT\n2 Delete\n3. Display 4 Exit")
    printf("Enter your choice");
    switch(choice)
    { case 1: printf("Enter item to be inserted:");
              scanf("%d", &item);
              insert();
              break;
      case 2; item = delete();
              if (item==-1)
                printf("The queue is empty\n");
              else
                printf("item deleted =%d\n", item);
              break;
      case 3: display();
              break;
      default: exit(0);
    }
  }
}
```

Output

MENU
1. INSERT
2. Delete
3. Display
4. Exit
Enter your choice: 2
item deleted=15

MENU
1. Insert
2. Delete
3. Display
4. Exit
Enter your choice: 3
The queue items are:
16 7 18 14

Name S Skanda
USN 1BM19CS137

Date: 14/10/2020
Batch 2

2) Dqueue

```
#include <stdio.h>
#include <process.h>
# define qsize 5
int f=0, r=-1, item, q[10];

int isfull() { return (r==qsize-1)?1:0; }
int isempty() { return (f>r)?1:0; }
void insert_rear()
{ if (isfull())
    { printf("queue overflow \n");
      return;
    r=r+1; q[r]=item;
}
void delete_front()
{ if (isempty())
    { printf("The 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--;
    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("item deleted is %d \n", q[(r)--];
  if(isempty())
  { printf("Queue is empty \n");
    return;
  }
  printf("item deleted is %d \n", q[(r)--]);
  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 ", q[i]);
}
```

```
int main()
{ for(;;)
  { printf("\n1.insert_rear \n2.insert_front
        \n3.delete_rear \n4.delete_front \n5 display
        \n6.exit\n");
    printf("Enter choice :");
    scanf("%d",&ch);
    switch(ch)
    { case1: printf("Enter the item :");
             scanf("%d",&item);
             insert_rear();
             break;
      case 2: printf("Enter the item :");
              scanf("%d",&item);
              insert_front();
              break;
      case 3: delete_rear();
              break;
      case 4: delete_front(); break;
      case 5: display();
              break;
      default : exit(0);
    }
  }
  return 0;
}
```

Output

1 insert_rear
2 insert_front
3 delete_rear
4 delete_front
5 display
6 exit
Enter choice: 3
Item deleted is 5