DEQUEUE

```
#include<stdio.h>
#define qsize 3
int f=0,r=-1,ch;
int 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("queue 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)--]);
  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\t",q[i]);
 }
void main()
 {
  int n=1;
  while(n!=0)
   {
           printf("\n1.insert_rear\n2.insert_front\n3.delete_rear\n4.delete_front\n5.display\n6.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;
            case 6:n=0;
                  break;
            default:exit(0);
           }
          }
}
```

```
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
1
enter the item
11

1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
1
enter the item
22

1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
```

```
enter choice
1
enter the item
33

1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
1
enter the item
44
queue overflow

1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
5
11      22      33
1.insert_rear
```

```
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
4
item deleted is 11

1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
4
item deleted is 22

1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
2
```

```
enter the item
44

1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
5
44      33
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
3
item deleted is 33

1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
```

```
6.exit
enter choice
3
item deleted is 44

1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
3
queue is empty

1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
6


...Program finished with exit code 40
Press ENTER to exit console.
```