Dequeue

```
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#define qsize 7
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 (is full())
  {
   print f("Queue overflow \n");
   return;
  }
 r=r+1;
 q[r]=item;
}
```



```
void delite_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 (is empty())
    {
     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 (is empty())
    {
     printf ("Queue empty \n");
     return;
    }
   for (i=f; i<=r; i++)
    printf ("%d \n", q[i]);
 }

void main()
 {
   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_recent();
                 break;
        case 4 : delete_front();
                 break;
        case 5 : display();
                 break;
        default : exit(0);
    }
}
```