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

int isfull()
  {
   return(r==que_size-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()
```

```
45    }
46  void insert_front()
47    {
48     if(f!=0)
49       {
50        f = f-1;
51        q[f]=item;
52        return;
53       }
54      else if(f==0 && r==-1)
55       {
56        q[++(r)]=item;
57        return;
58       }
59      else
60        printf("insertion not possible\n");
61    }
62  void delete_rear()
63    {
64     if(isempty())
65       {
66        printf("queue is empty\n");
67        return;
68       }
69     printf("item deleted is %d\n",q[(r)--]);
70     if(f>r)
71       {
72        f=0;
73        r=-1;
74       }
75    }
76  void display()
77    {
78     int i;
79     if(isempty())
80       {
81        printf("queue empty\n");
82        return;
83       }
84     for(i=f;i<=r;i++)
85      printf("%d\n",q[i]);
86    }
87  void main()
88   {
89    int n=1;
90    while(n!=0)
```

```
        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()
 {
  int n=1;
  while(n!=0)
   {
    printf("1.insert_rear\n2.insert_front\n3.delete_rear\n4.delete_front\n5.display\n6.exit\n");
    printf("enter choice\n");
    scanf("%d",&choice);
    switch(choice)
      {
       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);
      }
   }
 }
```

```
main.c:113:13: warning: implicit declaration of function 'exit' [-Wimplicit-fu
main.c:113:13: warning: incompatible implicit declaration of built-in function
main.c:113:13: note: include '<stdlib.h>' or provide a declaration of 'exit'
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
1
enter the item
2
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
1
enter the item
3
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
1
enter the item
4
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
1
enter the item
5
1.insert_rear
2.insert_front
```

```
enter choice
1
enter the item
5
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
1
enter the item
6
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
1
enter the item
7
queue overflow
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
5
2
3
4
5
6
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
```

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

10-....pdf

```
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
4
queue empty
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
1
enter the item
4
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
2
enter the item
6
insertion not possible
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
5
4
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
```