Ordinary Queue

```
#include <stdio.h>
#include <stdlib.h>
#define QUE_SIZE 3
int item, front = 0, rear = -1, q[10];
void insertrear()
{if (rear == QUE_SIZE - 1)
{
 printf("queue overflow \n");
 return;
}
 rear = rear+1
 q[rear] = item
}int deletefront ()
{if (front > rear)
{ front = 0;
rear = -1;
return -1;
} return q[front++];
} void displayQ()
{int i;
if (front > rear)
{
 printf("Queue is empty \n");
 return;
}
 printf("Contents of queue \n");
 for(i= front; i<= rear; i++)
{
 printf("%d \n", q[i]);
}}
 int main()
{
```

```
int choice ;
for(;;)
{
printf("1:insertrear 2:delete front 3. display 4.exit \n");
printf("Enter the choice \n");
scanf("%d", &choice);
switch(choice)
{
case 1: printf("Enter the item to be inserted \n");
scanf("%d", &item)
insertrear();
break;
case 2: item= delete front();
if(item == -1)
printf("Queue is empty \n");
else
printf("Item deleted = %d \n", item);
break;
case 3: displayQ();
break;
default: exit(0);
}
}
}
```

```c
#include<stdio.h>
#include<stdlib.h>
#define QUE_SIZE 3
int item,front=0,rear=-1,q[10];
void insertrear()
{if(rear==QUE_SIZE-1)
{
  printf("queue overflow\n");
  return;
}
rear=rear+1;
q[rear]=item;
}int deletefront()
{if (front>rear)
{front=0;
rear=-1;
return -1;
}return q[front++];
}void displayQ()
{int i;
if (front>rear)
{
  printf("queue is empty\n");
  return;
}
printf("contents of queue\n");
for(i=front;i<=rear;i++)
{
  printf("%d\n",q[i]);
```

ixoticCadetblueMass#main.c

ticCadetblueMass

Run

main.c

```
}}
int main()
{
  int choice;
  for(;;)
  {
    printf("1:insertrear 2:deletefront 3:display 4:exit\n");
    printf("enter the choice\n");
    scanf("%d",&choice);
    switch(choice)
    {
      case 1:printf("enter the item to be inserted\n");
      scanf("%d",&item);
      insertrear ();
      break;
      case 2:item=deletefront();
      if(item==-1)
      printf("queue is empty\n");
      else
      printf("item deleted=%d\n",item);
      break;
      case 3:displayQ();
      break;
      default:exit (0);

    }

  }
}
```

```
clang-7 -pthread -lm -o main main.c
./main
1:insertrear 2:deletefront 3:display 4:exit
enter the choice
1
enter the item to be inserted
32
1:insertrear 2:deletefront 3:display 4:exit
enter the choice
1
enter the item to be inserted
34
1:insertrear 2:deletefront 3:display 4:exit
enter the choice
3
contents of queue
32
34
1:insertrear 2:deletefront 3:display 4:exit
enter the choice
2
item deleted=32
1:insertrear 2:deletefront 3:display 4:exit
enter the choice
2
item deleted=34
1:insertrear 2:deletefront 3:display 4:exit
enter the choice
2
queue is empty
1:insertrear 2:deletefront 3:display 4:exit
enter the choice
```

Circular Queue

```
#include <stdio.h>
#include <stdlib.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;
}
}
int main()
{
 int choice;
for(;;)
{
printf("\n1. Insertrear \n 2. Delete front \n3. Display \n4. 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=deletefront();
        if(item==-1)
        printf("queue is empty\n");
        else
        printf("Item deleted is %d \n", item);
        break;
case 3: displayq();
        break;
   default: exit(0);
}
}
```

main.c

```c
#include<stdio.h>
#include<stdlib.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");
```

main.c

```
30          return;
31        }
32        f=front;
33        printf("contents of queue \n");
34        for(i=0;i<=count;i++)
35        {
36          printf("%d\n",q[f]);
37          f=(f+1)%que_size;
38        }
39    }
40    int main()
41    {
42      int choice;
43      for(;;)
44      {
45        printf("\n1.Insert rear \n2.Delete front \n3.Display
          \n4.exit \n ");
6         printf("Enter the choice : ");
7         scanf("%d",&choice);
8         switch(choice)
9         {
          case 1:printf("Enter the item to be inserted :");
                 scanf("%d",&item);
                 insertrear();
                 break;
          case 2:item=deletefront();
                 if(item==-1)
                 printf("queue is empty\n");
                 else
```

```
            printf("queue is empty\n");
            else
            printf("item deleted is %d \n",item);
            break;
        case 3:displayq();
            break;
        default:exit(0);
    }
  }
}
```

```
$ clang-7 -pthread -lm -o main main.c
$ ./main

1.Insert rear
2.Delete front
3.Display
4.exit
 Enter the choice : 3
queue is empty
1.Insert rear
2.Delete front
3.Display
4.exit
 Enter the choice : 1
Enter the item to be inserted :21

1.Insert rear
2.Delete front
3.Display
4.exit
 Enter the choice : 3
contents of queue
21
0

1.Insert rear
2.Delete front
3.Display
4.exit
 Enter the choice :
```