C code for largest multiple of 3

```
#include <stdio.h>
#include <stdlib.h>

struct Queue
{
    int *arr;
    int front;
    int rear;
    int size;
};

struct Queue * createQueue(int size)
{
    struct Queue *q =(struct Queue*)malloc(sizeof(struct Queue));
    q->arr = (int *) malloc (sizeof(int)*size);
    q->front=q->rear=-1;
    q->size = size;
    return q;
}

void enqueue(struct Queue* q, int element)
{
  if(q->front == -1)
      q->front = q->rear =0;

  q->arr[q->rear] = element;
  q->rear++;
}

int isEmpty(struct Queue *q)
{
    if(q->front == -1) return 1;
    else return 0;
}

void dequeue(struct Queue *q)
{
    q->front ++;
    if(q->front == q->rear)
```

```
        q->front = q->rear = -1;
}

void printQueue(struct Queue* q)
{
     for(int i=q->front ;i<q->rear; i++)
      printf("%d",q->arr[i]);
}

int swap(int *a, int *b)
{
     int temp = *a;
     *a = *b;
     *b = temp;
}

int partition(int * arr, int l, int r)
{
     int pivot = arr[r];
     int i = l-1;
     for(int j=l; j<r; j++)
     {
          if(arr[j]<pivot)
          {
               i++;
               swap(&arr[i],&arr[j]);
          }
     }
     swap(&arr[i+1],&arr[r]);

     return (i+1);
}

void q_sort(int *arr, int l, int r)
{
     if(l<r)
     {
          int p = partition(arr, l, r);
          q_sort(arr, l , p-1);
          q_sort(arr, p+1, r);
     }
}


void findMaxMultipleof3(int *arr, int size)
```

```
{
    q_sort(arr,0,size-1);
    struct Queue * q0 = createQueue(size);
    struct Queue * q1 = createQueue(size);
    struct Queue * q2 = createQueue(size);
    int i, sum;
    for(i=0,sum=0;i<size;i++)
    {
        sum = sum + arr[i];
        if((arr[i]%3) == 0)
          enqueue(q0,arr[i]);
        else if(arr[i]%3 ==1)
          enqueue(q1,arr[i]);
        else
          enqueue(q2,arr[i]);
    }
    if(sum%3 == 1)
    {
        if(!isEmpty(q1))
        {
            dequeue(q1);
            size --;
        }
        else
        {

            if(!isEmpty(q2))
            {
             dequeue(q2);
             size--;
            }
            else
              printf("Multiple of 3 is not possible");
            if(!isEmpty(q2))
            {
             dequeue(q2);
             size--;
            }
            else
              printf("Multiple of 3 is not possible");
        }
    }
    else if(sum%3 == 2)
    {
      if(!isEmpty(q2))
```

```
        {
            dequeue(q2);
            size --;
        }
        else
        {

            if(!isEmpty(q1))
            {
             dequeue(q1);
             size--;
            }
            else
              printf("Multiple of 3 is not possible");
            if(!isEmpty(q1))
            {
             dequeue(q1);
             size--;
            }
            else
              printf("Multiple of 3 is not possible");
        }

    }
    int temp[size];
    maketemp(temp,q0,q1,q2);
    q_sort(temp,0,size-1);
    printarrreverse(temp,size);
}
void maketemp(int*temp,struct Queue*q0,struct Queue*q1,struct
Queue*q2)
{
    int i=0;
    for(int j=q0->front;j<q0->rear;j++,i++)
    {
        temp[i] = q0->arr[j];
    }
    for(int j=q1->front;j<q1->rear;j++,i++)
    {
        temp[i] = q1->arr[j];
    }
    for(int j=q2->front;j<q2->rear;j++,i++)
    {
        temp[i] = q2->arr[j];
    }
```

```
}

void printarrreverse(int *temp,int size)
{
     for(int i=size-1;i>=0;i--)
        printf("%d",temp[i]);
}
void main()
{
   int arr[]={3,3,6,1,1,1};
   findMaxMultipleof3(arr,6);


}
```