```c
#include<stdio.h>
#include<stdlib.h>
struct arrqueue
{
    int *a;
    int m;
    int f,r;
};
typedef struct arrqueue *queue;
typedef int position;
typedef int element;

queue makenullqueue(int sz)
{
    queue q;
    q=(queue)malloc(sizeof(struct arrqueue));
    q->a=(int *)malloc(sizeof(int)*sz);
    q->m=sz;
    q->f=0;
    q->r=0;
    return q;
}
void enqueue(queue q,element e)
{
    if(!isfull(q))
    {
        q->a[rearpos(q)]=e;
        q->r++;
    }
    else
        printf("\nQueue is Full");

}
element dequeue(queue q)
{
    position i;
    element e;
    if(!isempty(q))
    {
```

```
        e=q->a[frontpos(q)];

for(i=frontpos(q);i<rearpos(q);i=nextpos(q,i))
            q->a[i]=q->a[i+1];
        q->r--;
        return e;
    }
    else
        printf("\nQueue is Empty");
    return -1;
}

int isempty(queue q)
{
    if(q->f==q->r)
        return 1;
    return 0;
}
int isfull(queue q)
{
    if(q->r==q->m)
        return 1;
    return 0;
}

void printqueue(queue q)
{
    position i;
    printf("\nElements in the Queue are:\n");

for(i=frontpos(q);i<rearpos(q);i=nextpos(q,i))
        printf("%d ",q->a[i]);
    printf("\n");
}

position frontpos(queue q)
{
    return 0;
}
```

```
position rearpos(queue q)
{
    return q->r;
}
position nextpos(queue q,position p)
{
    return p+1;
}
int elelength(element ele)
{
    int c=0;
    while(ele>0)
    {
        ele=ele/10;
        c++;
    }
    return c;
}
int max(int a,int b)
{
    if(a>b)
        return a;
    return b;
}
int getdigit(int ele,int i)
{
    int j,n;
    for(j=0;j<i;j++)
    {
        n=ele%10;
        ele=ele/10;
    }
    return n;
}
void radixsort()
{
    int no,ele,i,j,md=0,k;
    queue q,qb[10];
    q=makenullqueue(20);
```

```
    for(i=0;i<10;i++)
        qb[i]=makenullqueue(20);
    printf("\nEnter number of elements:");
    scanf("%d",&no);
    printf("\nEnter elements:\n");
    for(i=0;i<no;i++)
    {
        scanf("%d",&ele);
        enqueue(q,ele);
        md=max(md,elelength(ele));
//      printqueue(q);
    }
    for(i=1;i<=md;i++)
    {
        while(!isempty(q))
        {
            ele=dequeue(q);
            k=getdigit(ele,i);
            enqueue(qb[k],ele);
        }
        for(j=0;j<10;j++)
        {
            while(!isempty(qb[j]))
            {
                ele=dequeue(qb[j]);
                enqueue(q,ele);
            }
        }
    }
    printqueue(q);
}
int main()
{
    radixsort();
    return 0;
}
```