CIRCULAR QUEUE

By V. Kenny Philip

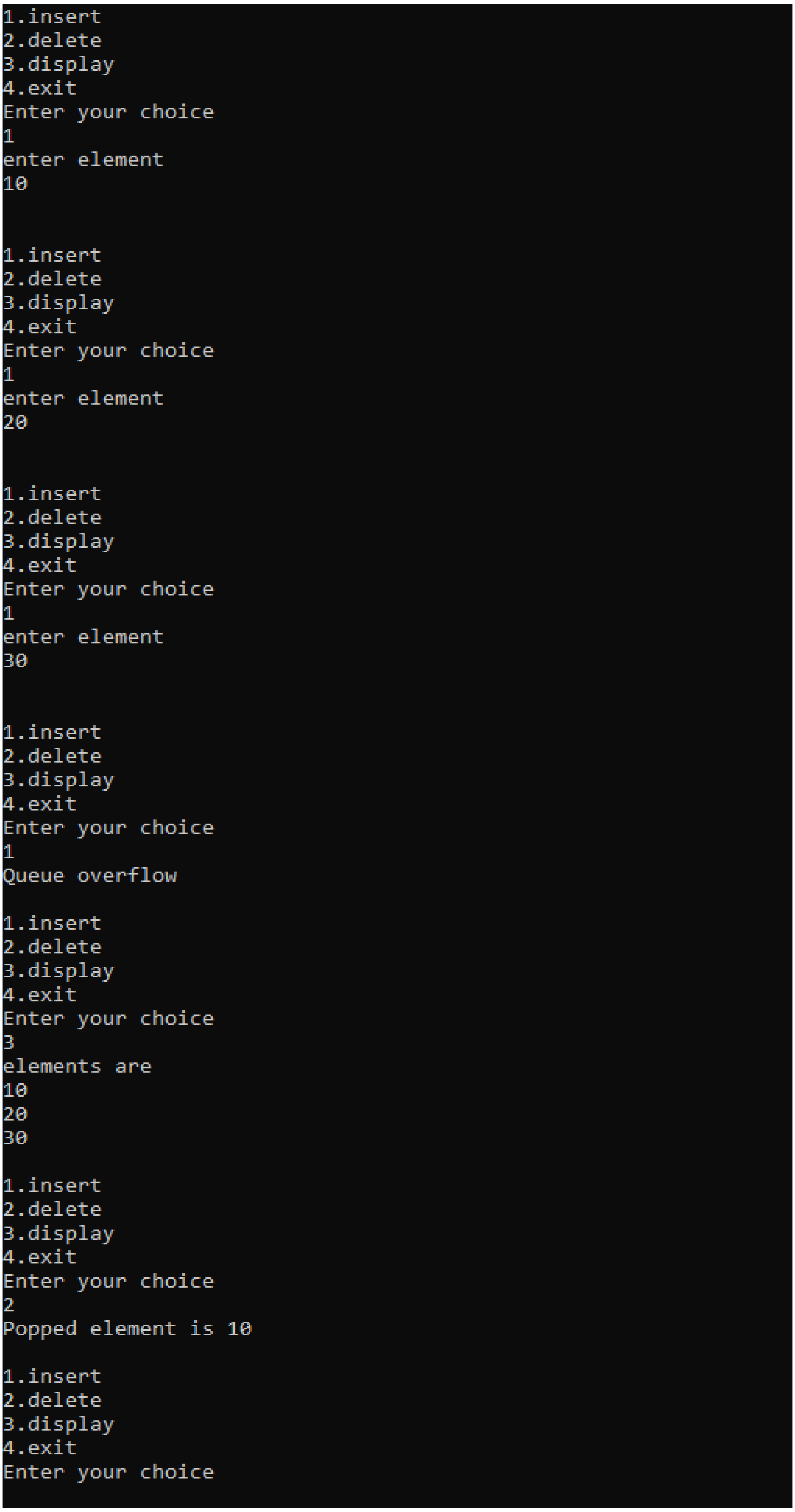
```
#include<stdio.h>
#include<stdlib.h>
int size=3,f=-1,r=-1,item;
int Q[3];
void insert();
void delete1();
void display();
int main()
{
while(1)
{
int ch;
printf("\n1.insert \n2.delete\n3.display\n4.exit");
printf("\nEnter your choice\n");
  scanf("%d",&ch);
  switch(ch)
  {
   case 1:insert();
          break;
   case 2:delete1();
          break;
   case 3:display();
          break;
   case 4:exit(0);
   default:printf("\nInvalid choice");
  }
printf("\n");
 }
return 0;
```

```
}
void insert()
{
if((f==0 && r==size-1) || f==r+1)
{
printf("Queue overflow");
return;
}
printf("enter element\n");
scanf("%d",&item);
if(f==-1 && r==-1)
{
f++;
Q[++r]=item;
}
else if(f>0 && r==size-1)
{
r=(r+1)%size;
Q[r]=item;
}
else
Q[++r]=item;
return;
}
void delete1()
{
if(f==-1 && r==-1)
{
printf("\nQueue is underflow");
return;
}
```

```
if(f==r)
{
   printf("%d",Q[r]);
   f=-1,r=-1;
   return;
}
if(f<r)
{
printf("Popped element is %d",Q[f++]);
return;
}
if(r<f)
{
if(f==size-1)
{
f=(f+1)%size;
}
  printf("Popped element is %d",Q[f++]);
}
}
void display()
{
if(f==-1 && r==-1)
{
printf("\nQueue is underflow");
return;
}
printf("elements are");
if(f<r)
{
  for(int i=f;i<=r;i++)
```

```
   {
       printf("\n%d",Q[i]);
   }
}
else
{
   for(int i=f;i<size;i++)
   {
        printf("\n%d",Q[i]);
   }
   for(int i=0;i<=r;i++)
    printf("\n%d",Q[i]);
}
return;
}
```

OUTPUT



```
1.insert
2.delete
3.display
4.exit
Enter your choice
1
enter element
10


1.insert
2.delete
3.display
4.exit
Enter your choice
1
enter element
20


1.insert
2.delete
3.display
4.exit
Enter your choice
1
enter element
30


1.insert
2.delete
3.display
4.exit
Enter your choice
1
Queue overflow

1.insert
2.delete
3.display
4.exit
Enter your choice
3
elements are
10
20
30

1.insert
2.delete
3.display
4.exit
Enter your choice
2
Popped element is 10

1.insert
2.delete
3.display
4.exit
Enter your choice
```