

```
#include<stdio.h>

#include<stdlib.h>

#include<process.h>

#define que_size 5

int item,front=0,rear=-1,q[que_size],count=0;

void insertrear()

{

    if(count==que_size)

    {

        printf("queue overflow\n");

        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-=1;

    return item;

}

void displayq()
```

```

{
    int i,f;
    if(count==0)
    {
        printf("queue is empty\n");
        return;
    }
    f=front;
    printf("contents of queue \n");
    for(i=0;i<count;i++)
    {
        printf("%d\n",q[f]);
        f=(f+1)%que_size;
    }
}

void main()
{
    int choice;
    for(;;)
    {
        printf("\n1.Insert rear\n2.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;  
case 4: exit(0);  
default:printf("WRONG CHOICE!");  
    }  
}  
return 0;  
}
```

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

```
#define QUESIZE 3
```

```
int item, front = 0, rear = -1, q[QUESIZE],  
count = 0;
```

```
void insertrear()
```

```
{
```

```
if (count == QUESIZE)
```

```
{ printf("Queue Overflow\n");  
return;
```

```
}
```

```

rear = (rear + 1) % QUEUE_SIZE; q[rear] = item;
count++;
return;
}

```

```

int deletefront()
{
    if (count == 0) return -1;
    item = q[front];
    front = (front + 1) % QUEUE_SIZE;
    count--;
    return item;
}

```

```

void display()
{
    int i, f = front;
    if (count == 0)
        printf("Queue empty");
        return;
    }
    printf("Contents: \n");
    for (int i = 0; i <= count; i++)
    {
        printf("%d \n", q[f]);
        f = (f + 1) % QUEUE_SIZE;
    }
}

```

```

void main()
{
    int choice;
}

```

```

clrscr();
for(ii).
{
printf("\nEnter 1 for insert 2 for delete 3 for display");
printf("\nEnter choice ");
scanf("%d", &choice);
switch (choice)
{
case 1: printf("\nEnter element ");
scanf("%d", &item);
insertrear();
break;
case 2: item = deletefront();
if (item == -1)
printf("\nQueue is empty");
else
printf("\nItem deleted");
break;
case 3: display();
break;
default: exit(0);
}
}
getch();
}

```

The screenshot shows the Visual Studio Code interface with a C++ project named 'cqueue.c - Prg data - Visual Studio Code'. The Explorer pane on the left shows the project structure, including a 'Stack implementation' folder and a 'cqueue.c' file. The main editor displays the C++ code for a queue implementation, which includes functions for inserting, deleting, and displaying elements. The Terminal pane at the bottom shows the output of the program, which prompts the user to enter a choice (1 for insert, 2 for delete, 3 for display) and then the corresponding item. The output shows the program running successfully and displaying the results of the operations.

```

cqueue.c - Prg data - Visual Studio Code
2: Code
PS D:\Prg data\C&C++\Stack implementation> cd "d:\Prg data\C&C++\Stack implementation" ; if ($?) { gcc cqueue.c -o cqueue } ; if ($?) { .\cqueue }

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

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

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

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

1.Insert rear
2.Delete front
3.Display
4.exit
Enter the choice : 2
item deleted is 1

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

```