Linear queue

```
# include <stdio.h> .
# define MAX 50.
void insert ();
void delete ();.
void display ();.
int queesize [MAX] .
int rear = -1 .
 int front = -1 ; .
main () .
{ int choice ;
 while (1) .
{ printf (" 1. Insert element \n");
  printf ("2. Delete element \n");
   printf ("3. Display \n");
    printf ("4. Exit \n");
  printf (" enter ur choice :");
   scanf (" %d", &choice);
  switch (choice)
{
   case 1 :
   insert ();
   break;
case 2 :

delete ();
break ;

case 3 :

display ();
break ;

case 4 :

exit (1) ;

deafult : printf ("error \n");

}

}
}

void insert ()

{ int additem ;

if ( rear = = MAX -1)

printf ("Queque overflow \n");

else
```

Date :

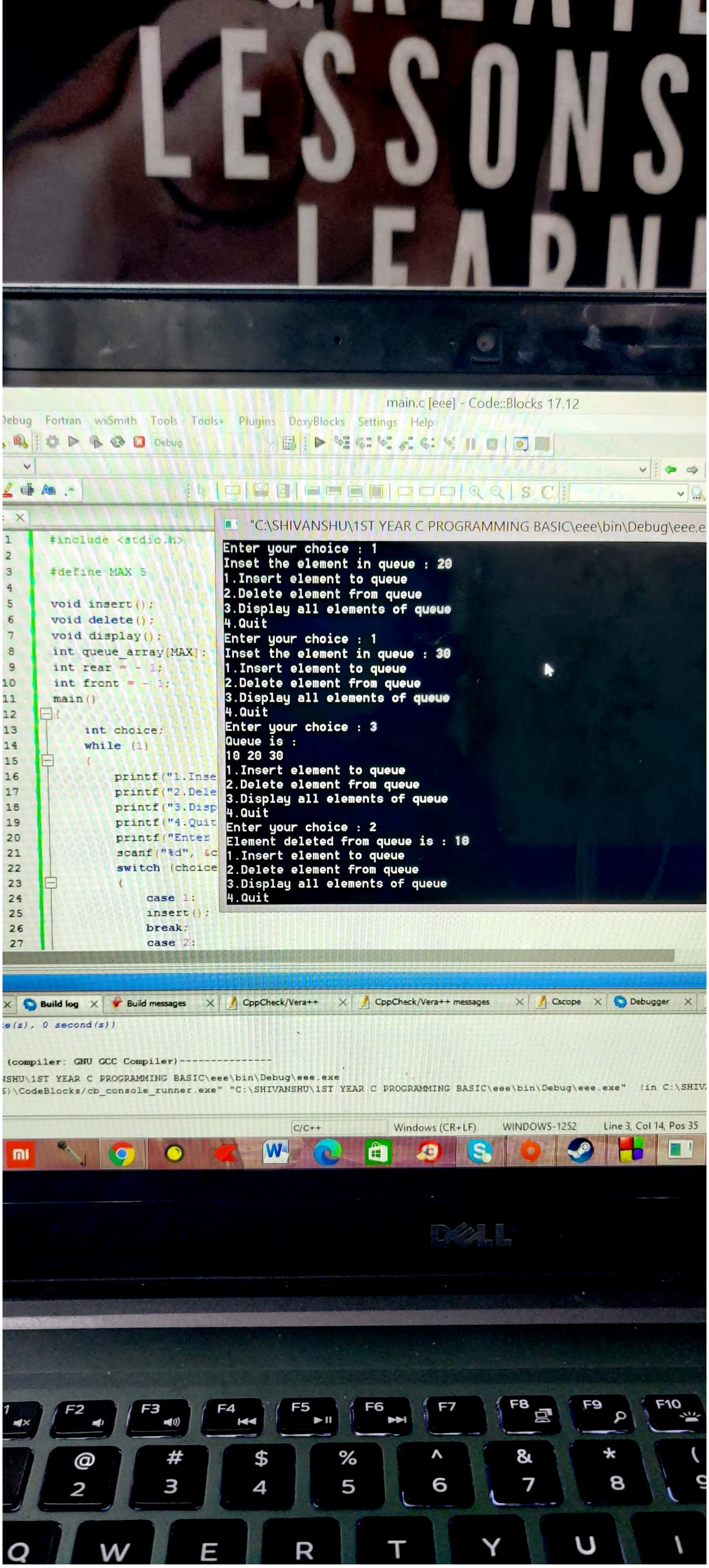
Page No. :

```
{ if (front = = -1)
    front = 0;
   printf ("Insert the element in queue:");
   scanf ("%d", &additem);
    rear = rear +1;
    queue [rear] = additem;
  }
 }
void delete ().
if ( front = = -1)
{ printf ("Queue Underflow \n");
   return;
 }
else
{ printf ("element deleted %d", qu
   front = front +1; }
}
```

gh art and
art only, that

'What is easy is
Seldom excellent

```
void display()
{ int i
if (front = = -1)
   printf ("Queue is empty \n");
else
{ printf ("Queue is : \n");
 for (i = front, i <= rear, i++)
   printf ("%d", queue[i]);
   printf ("\n");
}
}
```



```
#include <stdio.h>

#define MAX 5

void insert();
void delete();
void display();
int queue_array[MAX];
int rear = - 1;
int front = - 1;
main()
{
    int choice;
    while (1)
    {
        printf("1.Inse
        printf("2.Dele
        printf("3.Disp
        printf("4.Quit
        printf("Enter
        scanf("%d", &c
        switch (choice
        {
            case 1:
            insert();
            break;
            case 2:
```

"C:\SHIVANSHU\1ST YEAR C PROGRAMMING BASIC\eee\bin\Debug\eee.e

```
Enter your choice : 1
Inset the element in queue : 20
1.Insert element to queue
2.Delete element from queue
3.Display all elements of queue
4.Quit
Enter your choice : 1
Inset the element in queue : 30
1.Insert element to queue
2.Delete element from queue
3.Display all elements of queue
4.Quit
Enter your choice : 3
Queue is :
10 20 30
1.Insert element to queue
2.Delete element from queue
3.Display all elements of queue
4.Quit
Enter your choice : 2
Element deleted from queue is : 10
1.Insert element to queue
2.Delete element from queue
3.Display all elements of queue
4.Quit
```

```
e(s), 0 second(s))

(compiler: GNU GCC Compiler)---------------
ISHU\1ST YEAR C PROGRAMMING BASIC\eee\bin\Debug\eee.exe
5)\CodeBlocks/cb_console_runner.exe" "C:\SHIVANSHU\1ST YEAR C PROGRAMMING BASIC\eee\bin\Debug\eee.exe"  (in C:\SHIV
```

Scanned by CamScanner

Circular queue

```
#include <stdio.h>
#include <stlib.h>
#include < process.h>
# define quesize 3

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

void insertrear()

{   if ( count == quesize)
    { printf ("quee overflow");
      return ;  }
    rear = (rear+1) % quesize ;
    q[rear] item;
      count ++ ;
  }

int deletfront ()

{   if count(==0) return -1;
      item = q[front];
      front =(front+1) % quesize ,
    count = count-1;
    return item;
   }
  void   display()
  { 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;
  }
}
```

```
void main ()
{ int choice;
  for(;;)
  { printf ("\n1.Insert rear\n2.Delete front
              \n3. Display \n 4. exit \n");

    printf ("Enter option :");
    scanf ("%d", &choice);
    switch(choice)
    { case 1 : printf ("Enter item to be
                       inserted :");
               scanf ("%d", &item);
               insert rear ();
               break;
      case 2 : item= delete front ();
               if (item == -1)
               printf ("queue is empty
               else
printf ("item deleted is %d\n", item);
  break
case 3: display();
break;
default: exit (0);
}
}
}
```

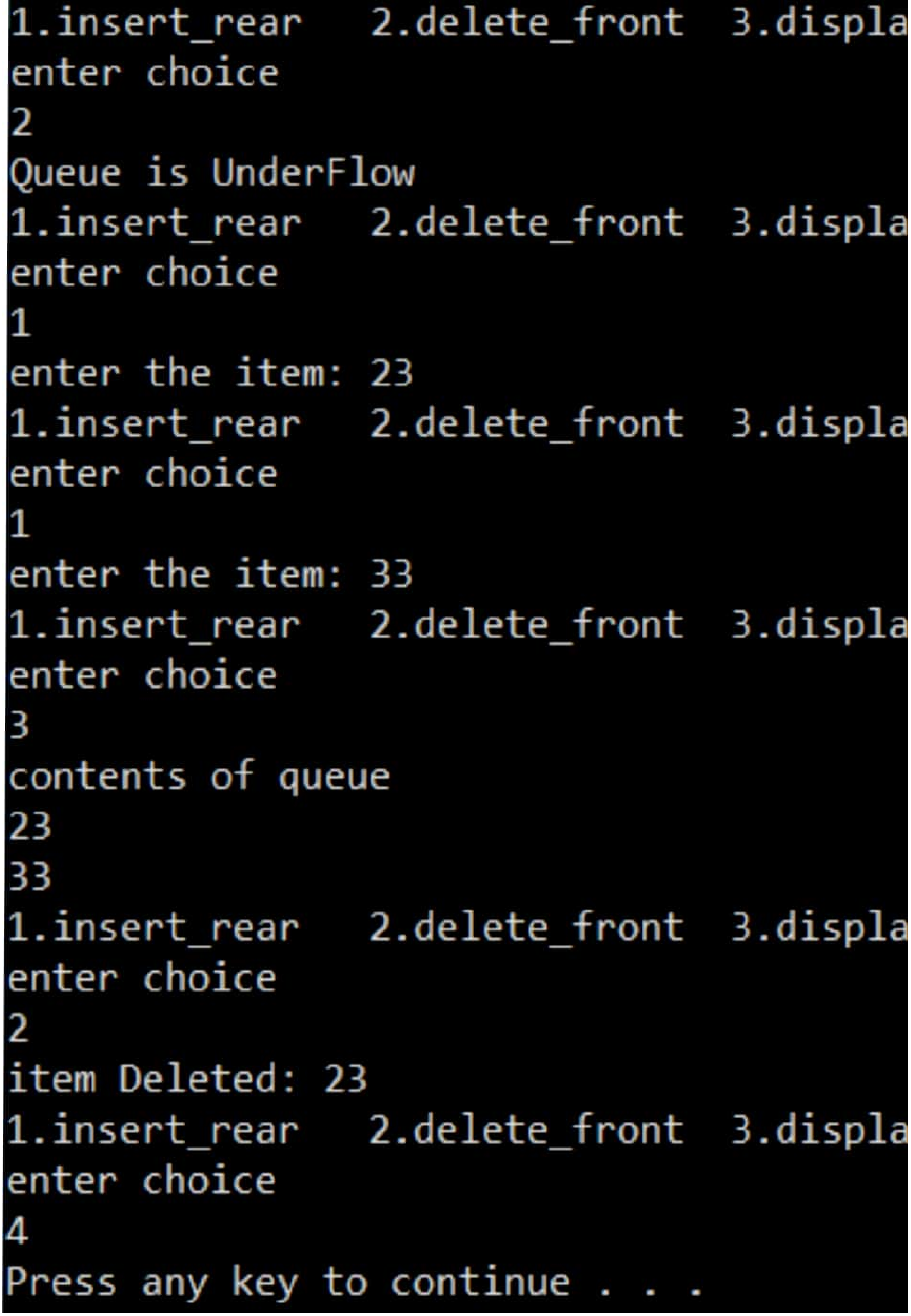
Scanned by CamScanner

d/i/p item);

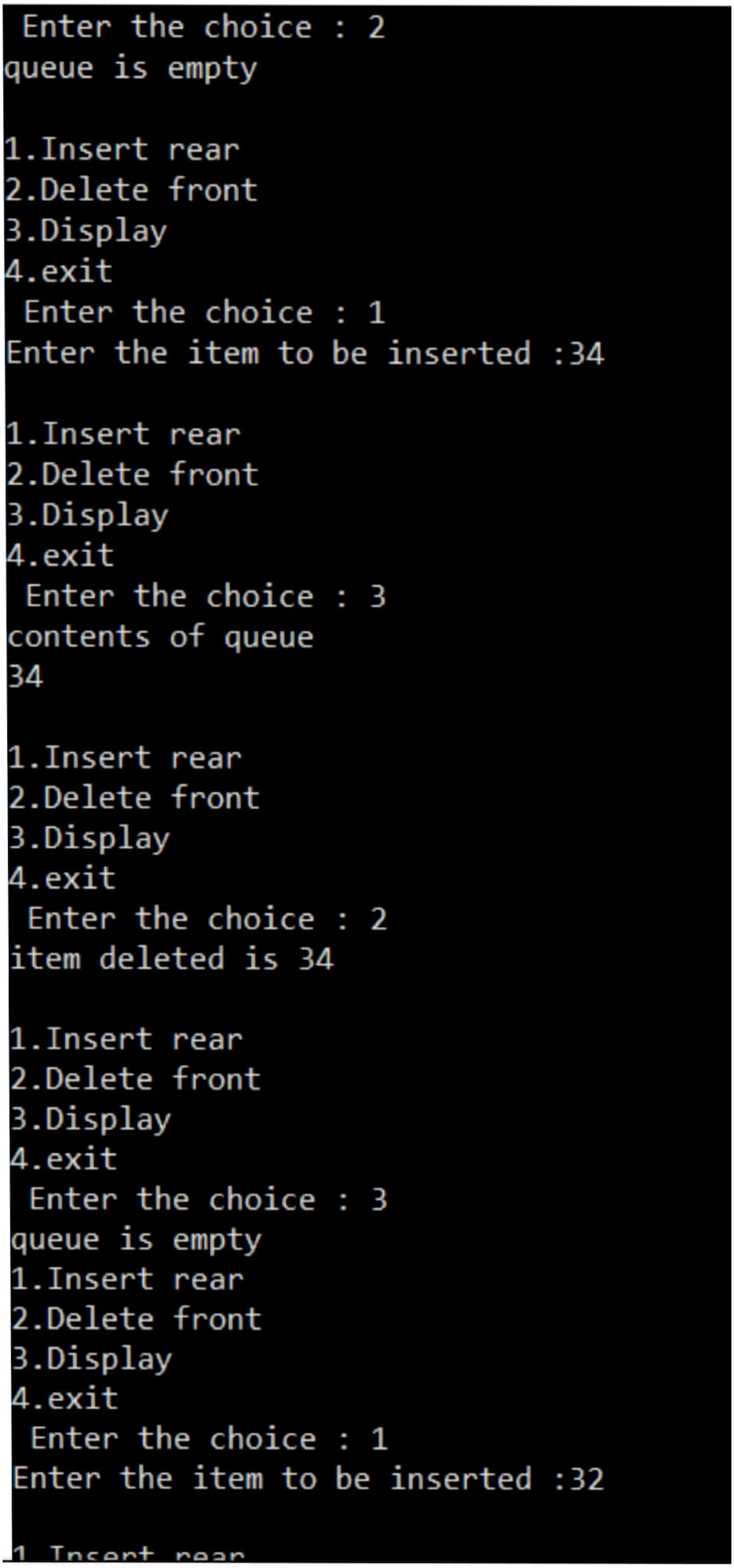
deque

```
#include <stdio.h>
#include <conio.h>
#include <process.h>.
# define  qsize 5
int   f=0,  r=-1, ch;
int   item, q[10];
int  isfull ().
{  return (r==  qsize-1)?1:0;.
}
int  isempty ()
{ return  (f>r) ? 1:0;
  }
void  insertrear ()
 {   if (isfull())
    { printf ("queque overflow");
        return;
    }
   r=r+1;
    q[r]=item; }
```

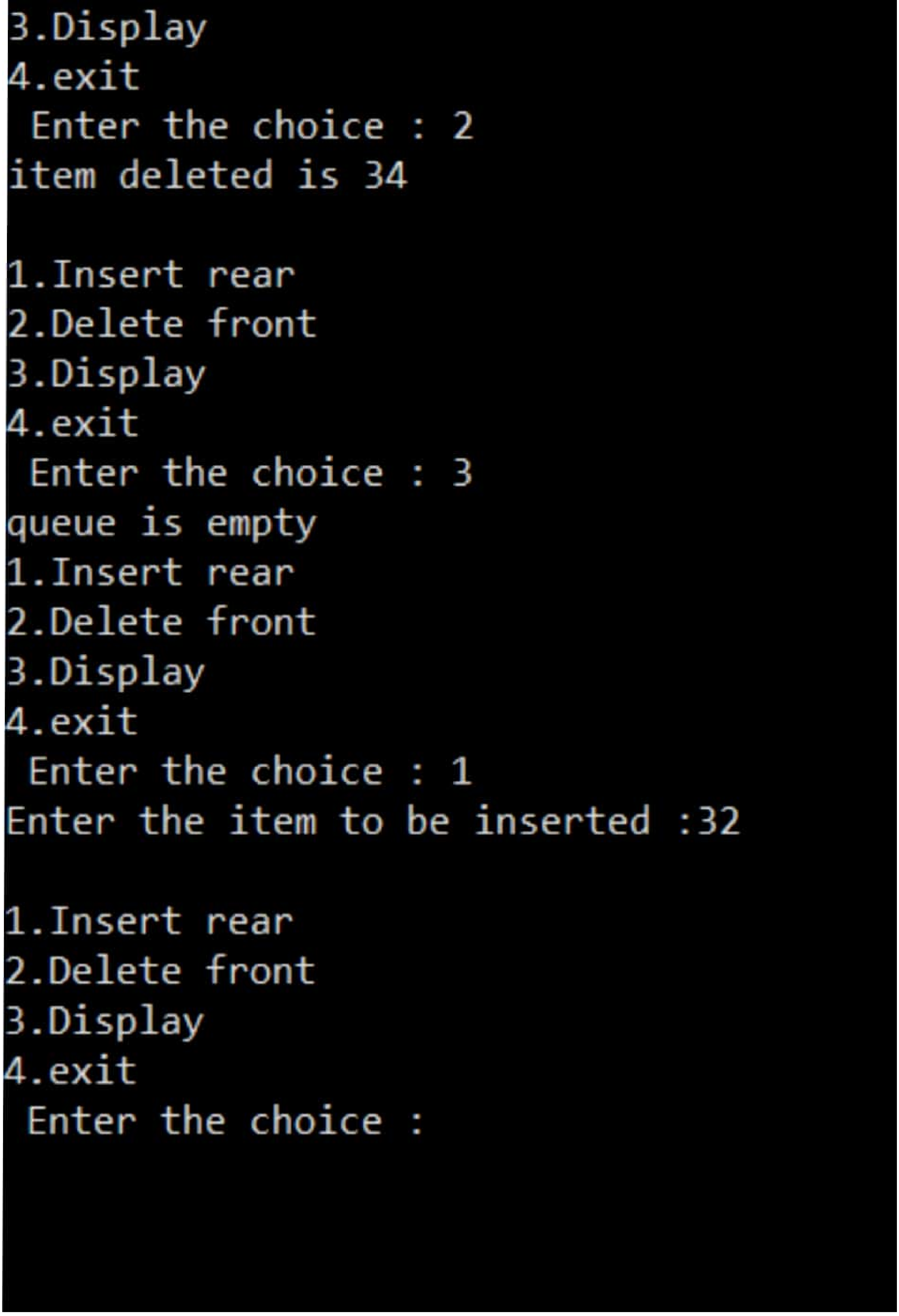
```
void deletefront()
{ if(isempty())
  { printf("queue empty\n");
    return;
  }
printf("item deleted is %d\n",q[f
if(f>r)
{   f=0;
    r=-1;
  }
}
void display()
{ int i;
if(isempty())
{   printf("queue empty\n");
    return;}
for (i=f; i<=r; i++)
printf("%d\n", q[i]); }
```



```
1.insert_rear   2.delete_front  3.displa
enter choice
2
Queue is UnderFlow
1.insert_rear   2.delete_front  3.displa
enter choice
1
enter the item: 23
1.insert_rear   2.delete_front  3.displa
enter choice
1
enter the item: 33
1.insert_rear   2.delete_front  3.displa
enter choice
3
contents of queue
23
33
1.insert_rear   2.delete_front  3.displa
enter choice
2
item Deleted: 23
1.insert_rear   2.delete_front  3.displa
enter choice
4
Press any key to continue . . .
```



```
 Enter the choice : 2
queue is empty

1.Insert rear
2.Delete front
3.Display
4.exit
 Enter the choice : 1
Enter the item to be inserted :34

1.Insert rear
2.Delete front
3.Display
4.exit
 Enter the choice : 3
contents of queue
34

1.Insert rear
2.Delete front
3.Display
4.exit
 Enter the choice : 2
item deleted is 34

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 :32

1.Insert rear
```



```
3.Display
4.exit
 Enter the choice : 2
item deleted is 34

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 :32

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