Stack and Queue

Swaminathan Navinsashok
2019115126

Stack

SOURCE CODE

```c
#include<stdio.h>
#include<stdlib.h>
#define EMPTY_TOS (-1)
#define MIN_STACK_SIZE 2

struct stack_record
{
  int capacity;
  int tos;
  int *starr;   // element_type starr[100] ;
};

typedef struct stack_record *stack;

stack create_stack(int max_elements )
{
    stack s;

    if( max_elements < MIN_STACK_SIZE )
        printf("\n Stack size is too small \n ");

    s = (struct stack_record *) malloc( sizeof( struct stack_record ) );

    if( s == NULL )
        printf(" \n Out of space!!! \n ");

    s->starr = (int *)malloc( sizeof( int ) * max_elements );

    if( s->starr == NULL )
```

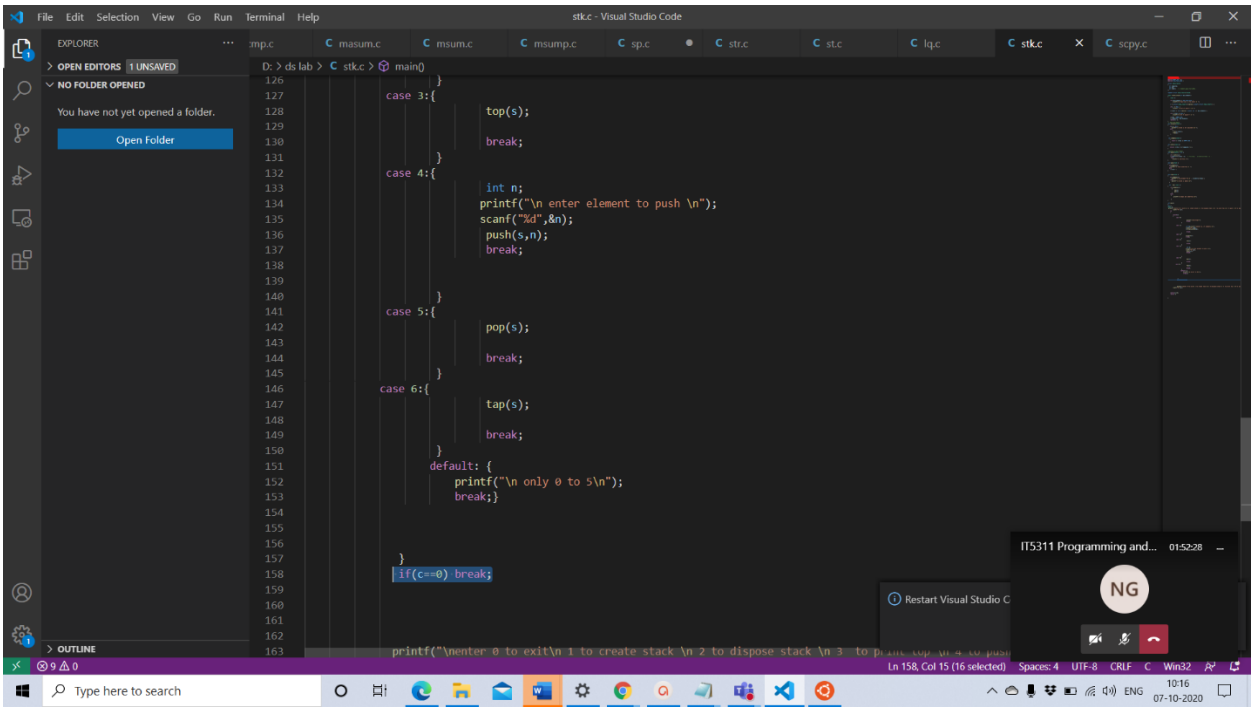
```
        printf("\n Out of space!!! \n ");

    s->tos = EMPTY_TOS;
    s->capacity = max_elements;
    return( s );
}
// Free the memory
void dispose(stack s)
{
   if(s == NULL)
      printf("\n stack is not available \n ");
   else
       {
        free(s->starr);
        free(s);
    }
}

 int isEmpty(stack s)
   {
      return ( s->tos == EMPTY_TOS );
   }

int isFull(stack s)
{
   return( s->tos == s->capacity -1 );
}

//pushing a new element
void push(stack s, int x)
{
    if( !isFull(s))
    s->starr[++s->tos] = x;   // ++s->tos;   s->starr[s->tos] = x ;
     else
       printf("\n overflow \n");
}

void pop(stack s)
{
  if(isEmpty(s))
    printf("\n stack underflow \n ");
  else
    s->tos-- ;
}
```

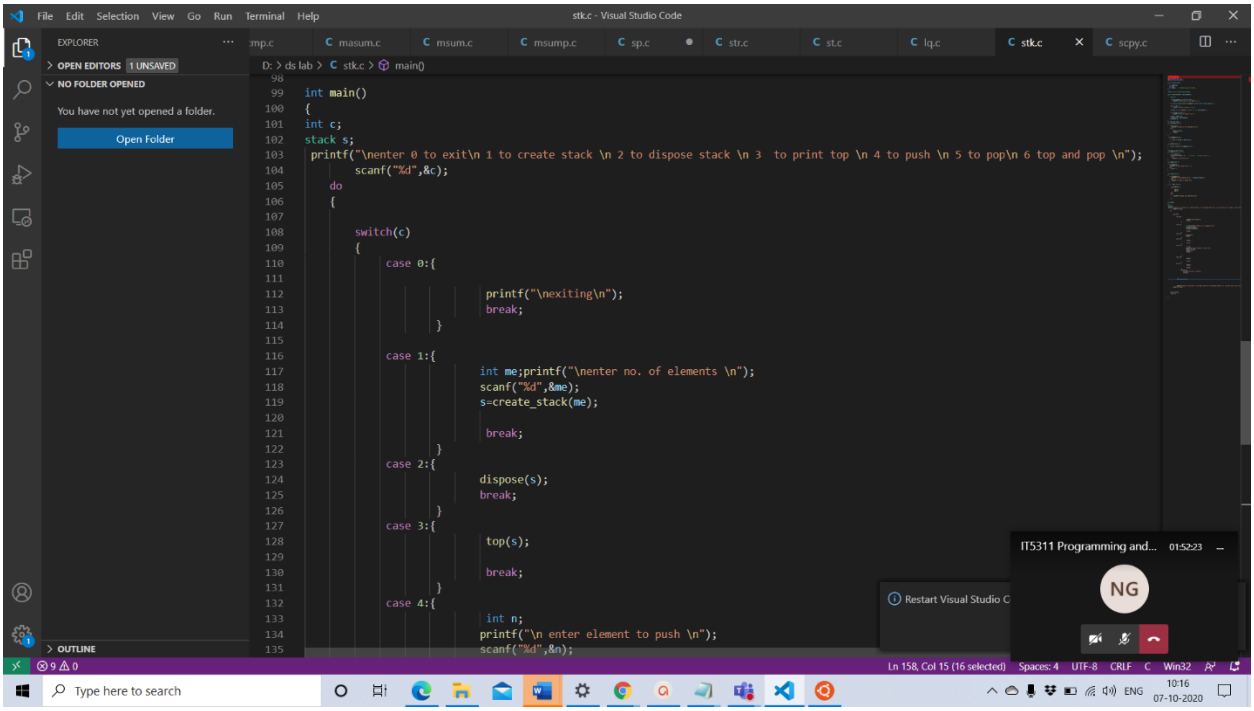
```
void top(stack s)
{
   if(!isEmpty(s))
      printf("\n top element %d \n ", s->starr[s->tos] );
    else
      printf("\n stack is empty \n");
  }

  int  tap( stack s)
{
    if(!isEmpty(s))
      {
          top(s);
          pop(s);
      }
    else
     {
         printf("\n empty and underflow \n");

     }
}

int main()
{
int c;
stack s;
 printf("\nenter 0 to exit\n 1 to create stack \n 2 to dispose stack \n 3  to pri
nt top \n 4 to push \n 5 to pop\n 6 top and pop \n");
        scanf("%d",&c);
    do
    {

        switch(c)
        {
             case 0:{

                              printf("\nexiting\n");
                              break;
                    }

             case 1:{
                             int me;printf("\nenter no. of elements \n");
                             scanf("%d",&me);
                             s=create_stack(me);
```

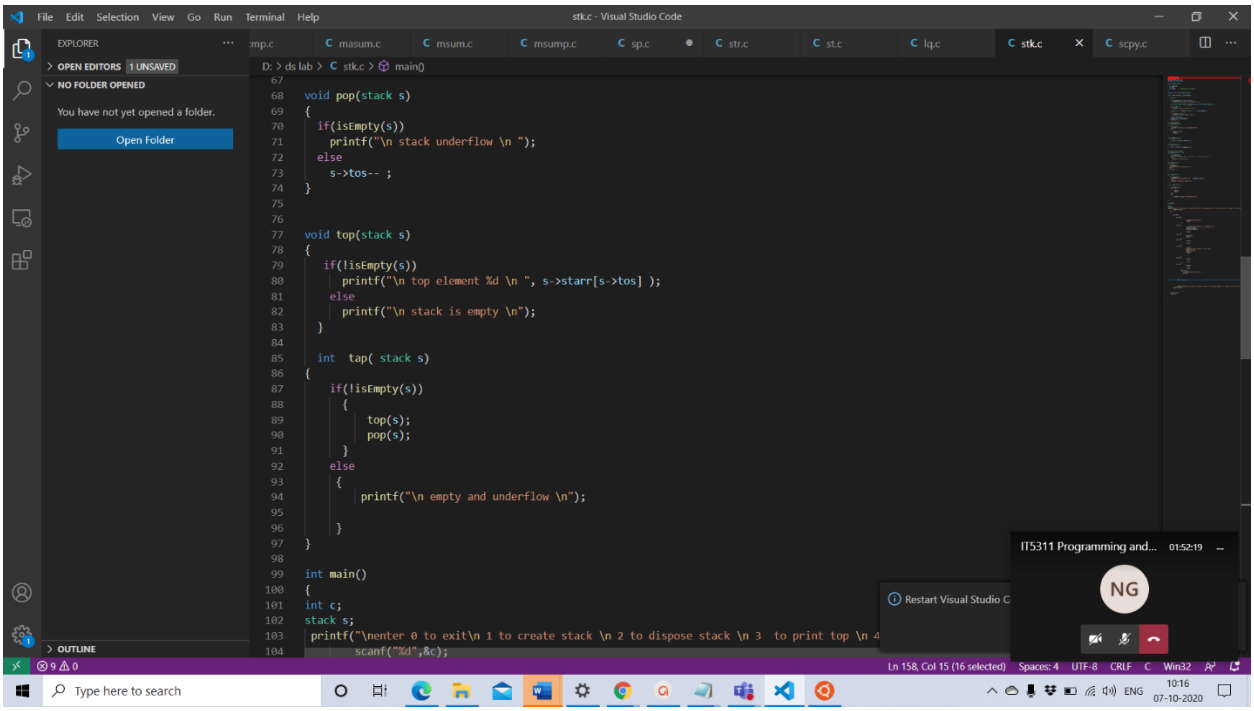
```
                        break;
                }
            case 2:{
                        dispose(s);
                        break;
                }
            case 3:{
                        top(s);

                        break;
                }
            case 4:{
                        int n;
                        printf("\n enter element to push \n");
                        scanf("%d",&n);
                        push(s,n);
                        break;


                }
            case 5:{
                        pop(s);

                        break;
                }
           case 6:{
                        tap(s);

                        break;
                }
                  default: {
                      printf("\n only 0 to 5\n");
                      break;}



              }
              if(c==0) break;




             printf("\nenter 0 to exit\n 1 to create stack \n 2 to dispose stack
\n 3  to print top \n 4 to push \n 5 to pop\n 6 top and pop \n");
       scanf("%d",&c);
```

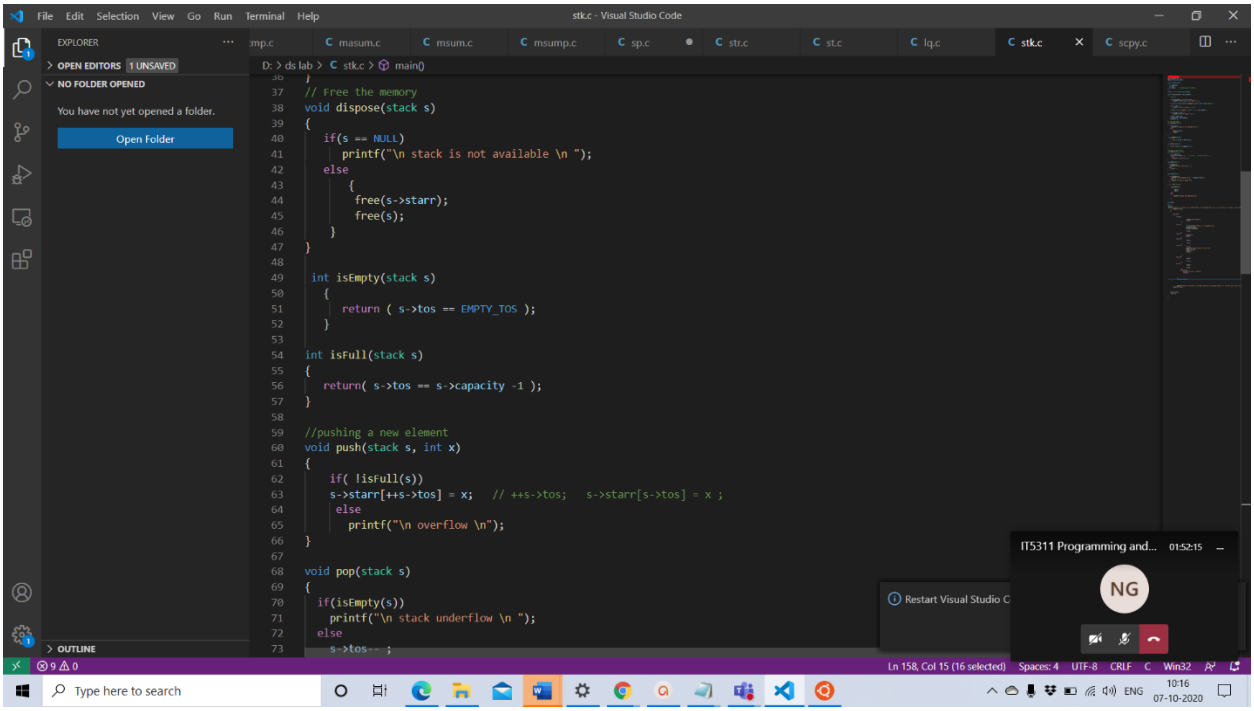
```
    }while(c!=0);
    return 0;


}
```

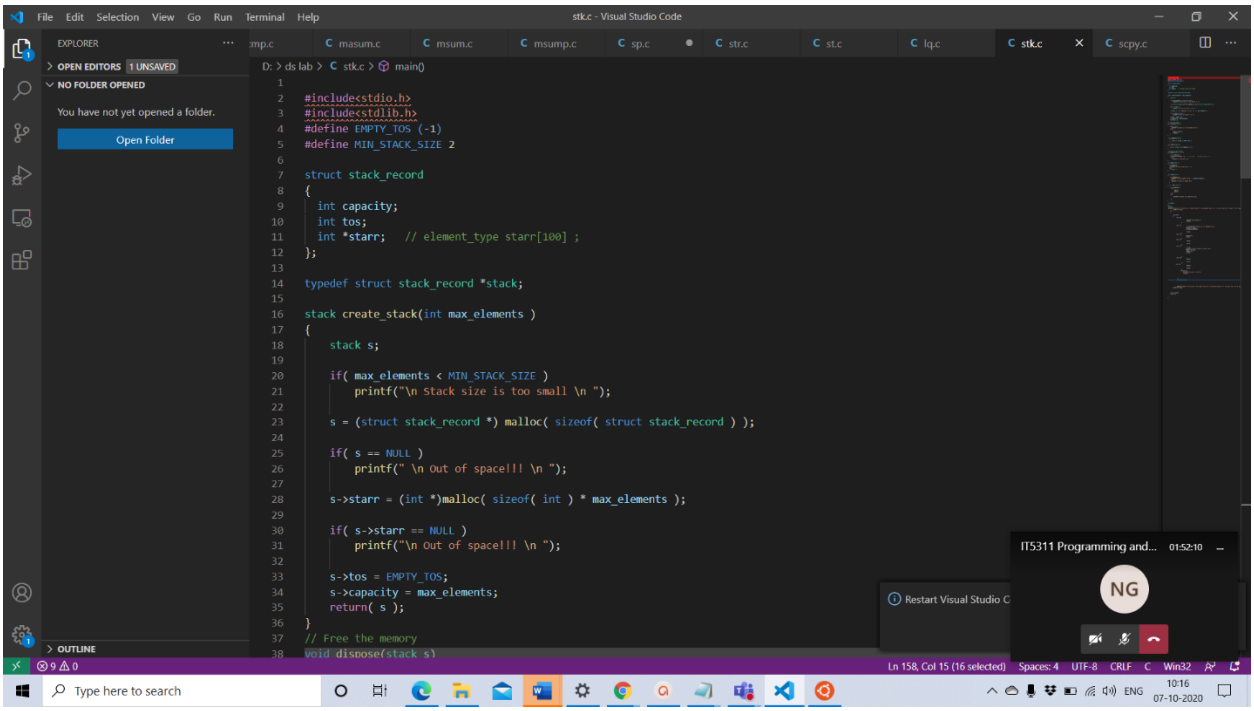
SCREENSHOT OF CODE

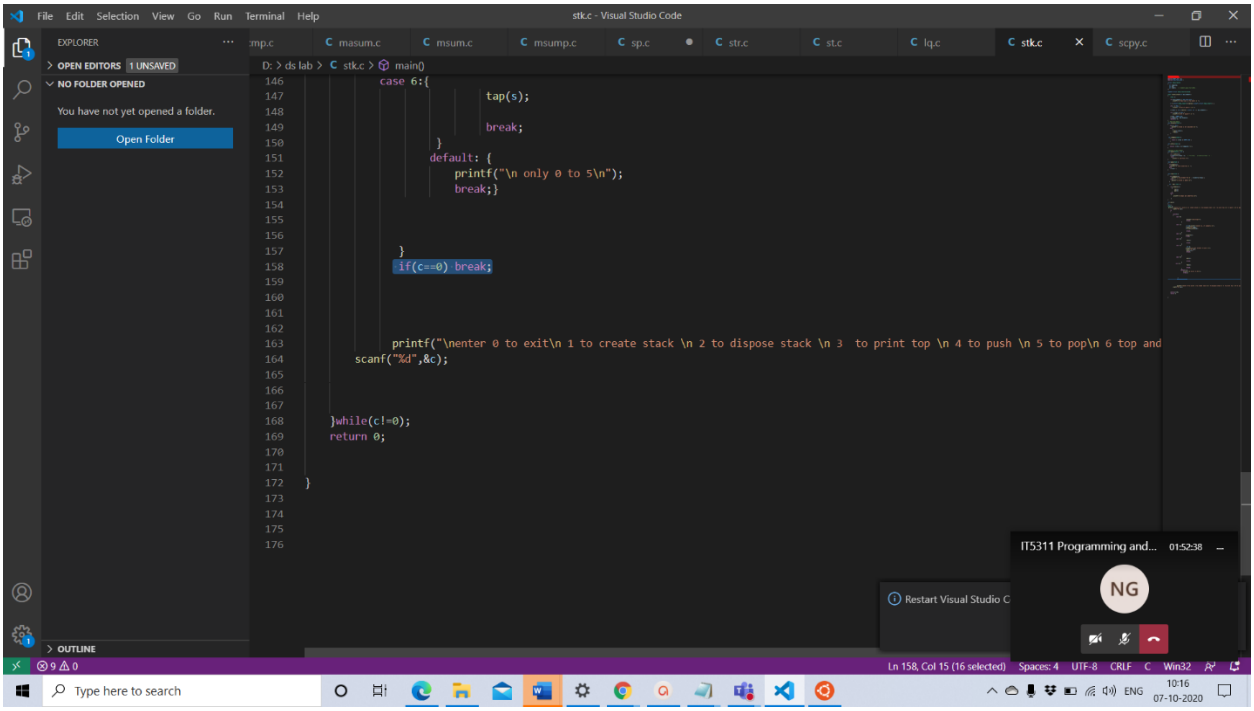

File Edit Selection View Go Run Terminal Help
stk.c - Visual Studio Code
EXPLORER
OPEN EDITORS 1 UNSAVED
NO FOLDER OPENED
You have not yet opened a folder.
Open Folder
OUTLINE
D: > ds lab > stk.c > main()
case 3:{
top(s);
break;
case 4:{
int n;
printf("\n enter element to push \n");
scanf("%d",&n);
push(s,n);
break;
case 5:{
pop(s);
break;
case 6:{
tap(s);
break;
default: {
printf("\n only 0 to 5\n");
break;}
if(c==0) break;
printf("\nenter 0 to exit\n 1 to create stack \n 2 to dispose stack \n 3 to p
Restart Visual Studio C
IT5311 Programming and... 01:52:28
NG
Ln 158, Col 15 (16 selected) Spaces: 4 UTF-8 CRLF C Win32
Type here to search
ENG
10:16
07-10-2020


File Edit Selection View Go Run Terminal Help
stk.c - Visual Studio Code
EXPLORER
OPEN EDITORS 1 UNSAVED
NO FOLDER OPENED
You have not yet opened a folder.
Open Folder
OUTLINE
D: > ds lab > stk.c > main()
int main()
{
int c;
stack s;
printf("\nenter 0 to exit\n 1 to create stack \n 2 to dispose stack \n 3 to print top \n 4 to push \n 5 to pop\n 6 top and pop \n");
scanf("%d",&c);
do
{
switch(c)
{
case 0:{
printf("\nexiting\n");
break;
case 1:{
int me;printf("\nenter no. of elements \n");
scanf("%d",&me);
s=create_stack(me);
break;
case 2:{
dispose(s);
break;
case 3:{
top(s);
break;
case 4:{
int n;
printf("\n enter element to push \n");
scanf("%d",&n);
Restart Visual Studio C
IT5311 Programming and... 01:52:23
NG
Ln 158, Col 15 (16 selected) Spaces: 4 UTF-8 CRLF C Win32
Type here to search
ENG
10:16
07-10-2020

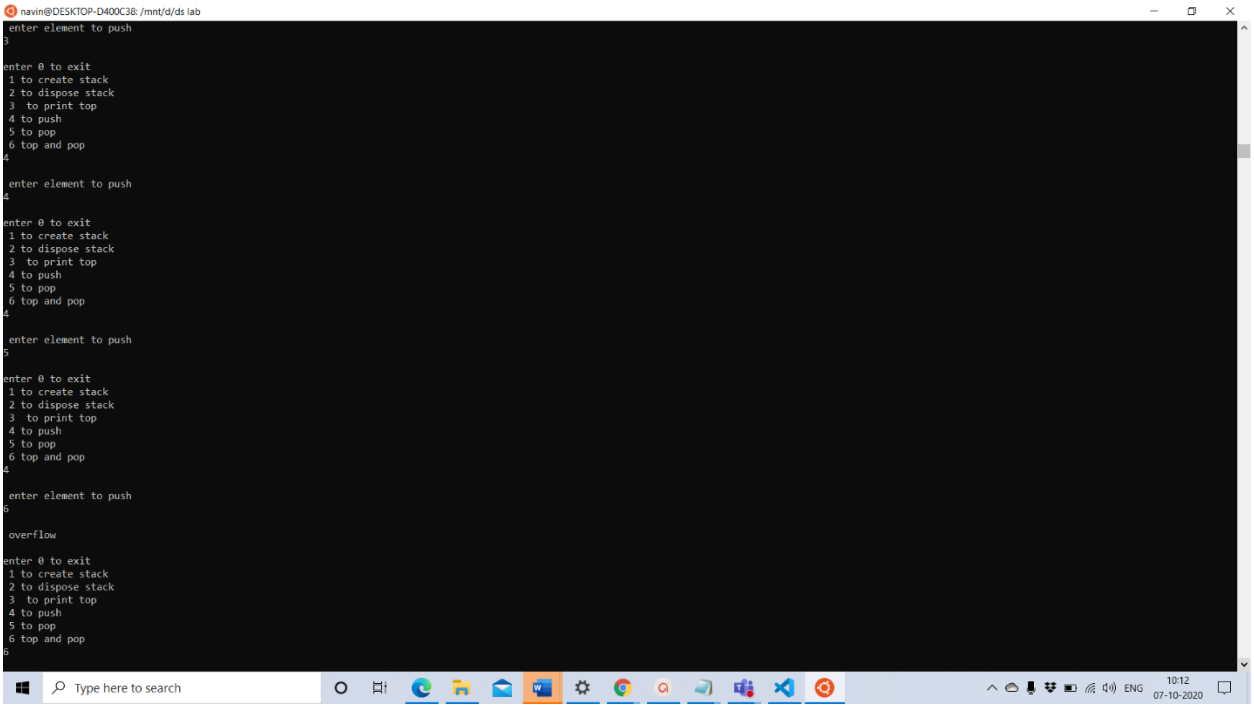

File Edit Selection View Go Run Terminal Help
stk.c - Visual Studio Code
EXPLORER
OPEN EDITORS 1 UNSAVED
NO FOLDER OPENED
You have not yet opened a folder.
Open Folder
OUTLINE
D: > ds lab > stk.c > main()
void pop(stack s)
{
if(isEmpty(s))
printf("\n stack underflow \n ");
else
s->tos-- ;
}
void top(stack s)
{
if(!isEmpty(s))
printf("\n top element %d \n ", s->starr[s->tos]);
else
printf("\n stack is empty \n");
}
int tap(stack s)
{
if(!isEmpty(s))
{
top(s);
pop(s);
}
else
{
printf("\n empty and underflow \n");
}
}
int main()
{
int c;
stack s;
printf("\nenter 0 to exit\n 1 to create stack \n 2 to dispose stack \n 3 to print top \n 4
scanf("%d",&c);
Restart Visual Studio C
IT5311 Programming and... 01:52:19
NG
Ln 158, Col 15 (16 selected) Spaces: 4 UTF-8 CRLF C Win32
Type here to search
ENG
10:16
07-10-2020

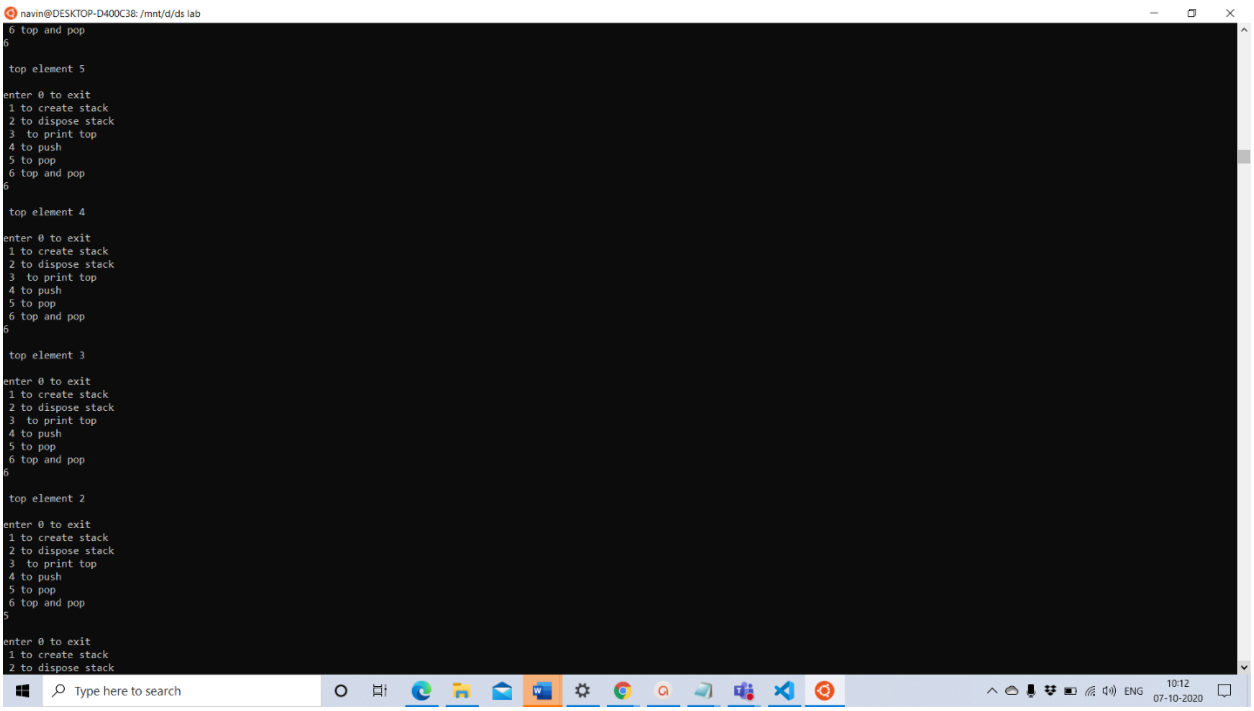

File Edit Selection View Go Run Terminal Help
stk.c - Visual Studio Code
EXPLORER
OPEN EDITORS 1 UNSAVED
NO FOLDER OPENED
You have not yet opened a folder.
Open Folder
OUTLINE
D: > ds lab > stk.c > main()
// Free the memory
void dispose(stack s)
{
if(s == NULL)
printf("\n stack is not available \n ");
else
{
free(s->starr);
free(s);
}
}
int isEmpty(stack s)
{
return (s->tos == EMPTY_TOS);
}
int isFull(stack s)
{
return(s->tos == s->capacity -1);
}
//pushing a new element
void push(stack s, int x)
{
if(!isFull(s))
s->starr[++s->tos] = x; // ++s->tos; s->starr[s->tos] = x ;
else
printf("\n overflow \n");
}
void pop(stack s)
{
if(isEmpty(s))
printf("\n stack underflow \n ");
else
s->tos-- ;
Restart Visual Studio C
IT5311 Programming and... 01:52:15
NG
Ln 158, Col 15 (16 selected) Spaces: 4 UTF-8 CRLF C Win32
Type here to search
ENG
10:16
07-10-2020

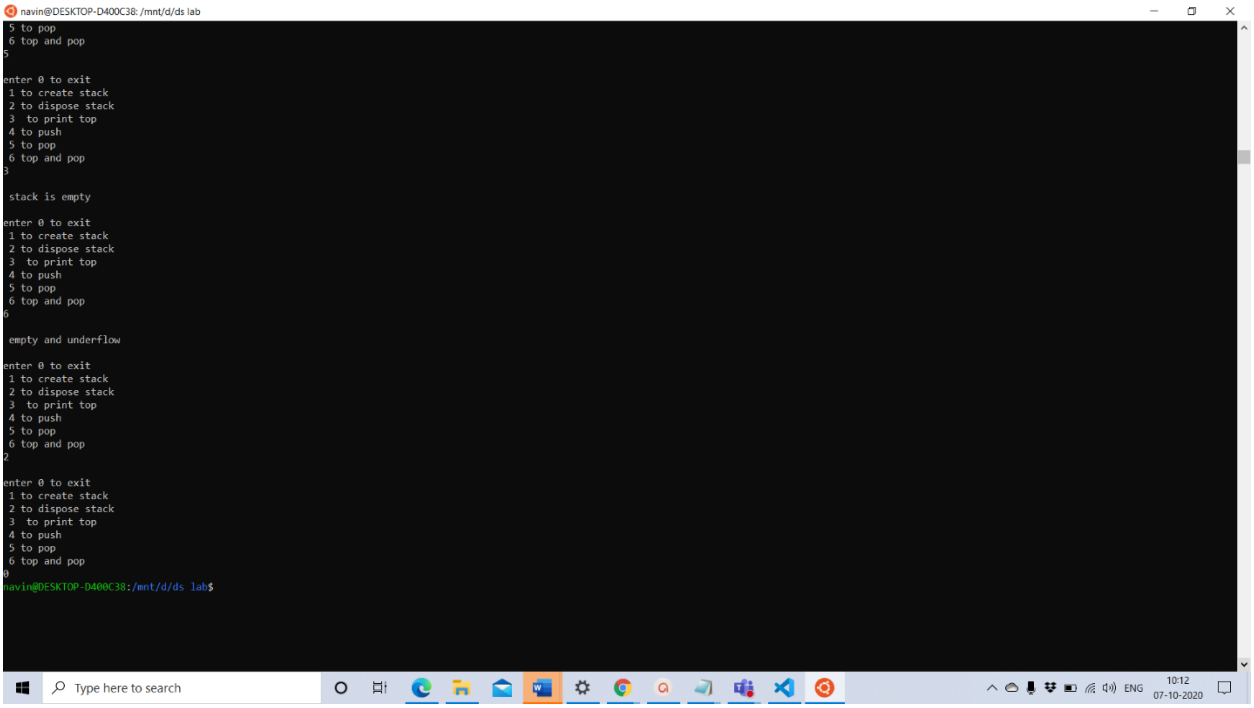


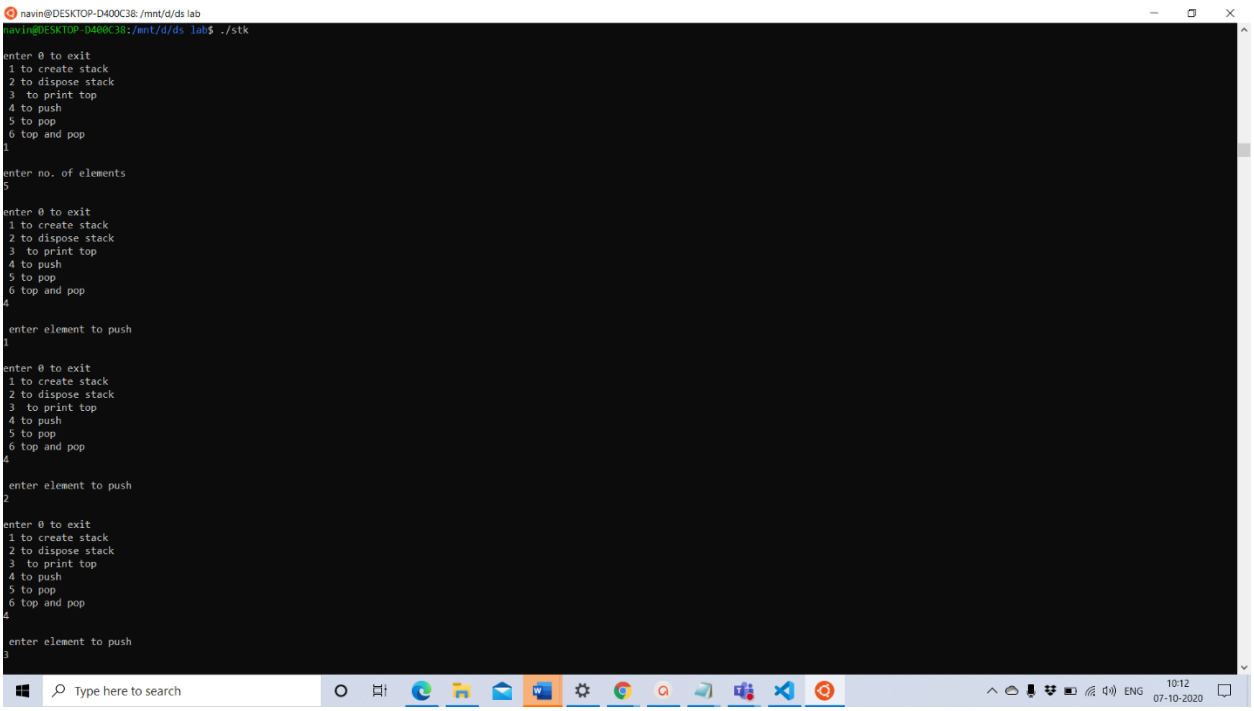


Output screenshot


navin@DESKTOP-D400C38: /mnt/d/ds lab
enter element to push
3
enter 0 to exit
1 to create stack
2 to dispose stack
3 to print top
4 to push
5 to pop
6 top and pop
4
enter element to push
4
enter 0 to exit
1 to create stack
2 to dispose stack
3 to print top
4 to push
5 to pop
6 top and pop
4
enter element to push
5
enter 0 to exit
1 to create stack
2 to dispose stack
3 to print top
4 to push
5 to pop
6 top and pop
4
enter element to push
6
overflow
enter 0 to exit
1 to create stack
2 to dispose stack
3 to print top
4 to push
5 to pop
6 top and pop
6
Type here to search
ENG
10:12
07-10-2020


navin@DESKTOP-D400C38: /mnt/d/ds lab
6 top and pop
6
top element 5
enter 0 to exit
1 to create stack
2 to dispose stack
3 to print top
4 to push
5 to pop
6 top and pop
6
top element 4
enter 0 to exit
1 to create stack
2 to dispose stack
3 to print top
4 to push
5 to pop
6 top and pop
6
top element 3
enter 0 to exit
1 to create stack
2 to dispose stack
3 to print top
4 to push
5 to pop
6 top and pop
6
top element 2
enter 0 to exit
1 to create stack
2 to dispose stack
3 to print top
4 to push
5 to pop
6 top and pop
5
enter 0 to exit
1 to create stack
2 to dispose stack
Type here to search
ENG
10:12
07-10-2020


navin@DESKTOP-D400C38: /mnt/d/ds lab
5 to pop
6 top and pop
5
enter 0 to exit
1 to create stack
2 to dispose stack
3 to print top
4 to push
5 to pop
6 top and pop
3
stack is empty
enter 0 to exit
1 to create stack
2 to dispose stack
3 to print top
4 to push
5 to pop
6 top and pop
6
empty and underflow
enter 0 to exit
1 to create stack
2 to dispose stack
3 to print top
4 to push
5 to pop
6 top and pop
2
enter 0 to exit
1 to create stack
2 to dispose stack
3 to print top
4 to push
5 to pop
6 top and pop
0
navin@DESKTOP-D400C38:/mnt/d/ds lab$
Type here to search
ENG
10:12
07-10-2020


navin@DESKTOP-D400C38: /mnt/d/ds lab
navin@DESKTOP-D400C38:/mnt/d/ds lab$ ./stk
enter 0 to exit
1 to create stack
2 to dispose stack
3 to print top
4 to push
5 to pop
6 top and pop
1
enter no. of elements
5
enter 0 to exit
1 to create stack
2 to dispose stack
3 to print top
4 to push
5 to pop
6 top and pop
4
enter element to push
1
enter 0 to exit
1 to create stack
2 to dispose stack
3 to print top
4 to push
5 to pop
6 top and pop
4
enter element to push
2
enter 0 to exit
1 to create stack
2 to dispose stack
3 to print top
4 to push
5 to pop
6 top and pop
4
enter element to push
3
Type here to search
ENG
10:12
07-10-2020

Queue

Source code

```c
#include<stdio.h>
#include<stdlib.h>
#define MIN_QUEUE_SIZE 2

struct queue_record
{
  int capacity;
  int front;
  int rear;
  int *qarr;
};

typedef struct queue_record *queue;

queue create_queue(int max_elements )
{
    queue q;

    if( max_elements < MIN_QUEUE_SIZE )
        printf("queue size is too small");

    q = (struct queue_record *) malloc( sizeof( struct queue_record )           )
;

    if( q == NULL )
        printf("Out of space!!!");

    q->qarr = (int *)malloc( sizeof( int ) * max_elements );

    if( q->qarr == NULL )
        printf("Out of space!!!");

    q-> front = q->rear = -1;
    q->capacity = max_elements;
    return( q );
}
```

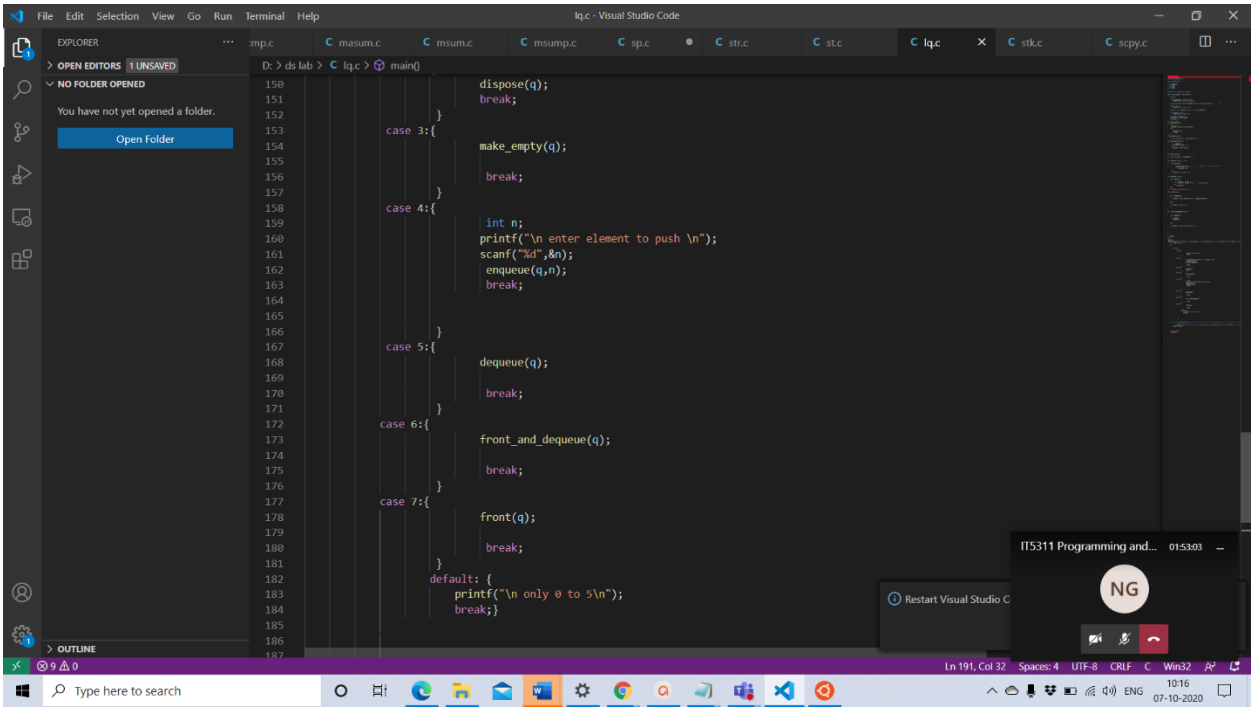
```
// Free the memory
void dispose(queue q)
{
   if(q == NULL)
      printf(" queue is not available");
   else
       {
        free(q->qarr);
        free(q);
    }
}
int isEmpty(queue q)
   {
      return (q->front == -1 && q->rear == -1 );
   }
void make_empty(queue q)
{
      if( !isEmpty(q) )
        q->front = q->rear = -1 ;
     else
        printf(" Already empty");
}



int isFull(queue q)
{
   return( q->rear ==  q->capacity -1 );
}

void enqueue( queue q , int x)
{
     if( !isFull(q) )
        {
            q->qarr[ ++q->rear ] = x ;       // ++q->rear ;  q->qarr[q-
>rear] = x ;
            if ( q->front == -1)
                 q->front = 0;
        }
     else
         printf( "\n overflow \n ");
   }

void dequeue( queue q)
{
```

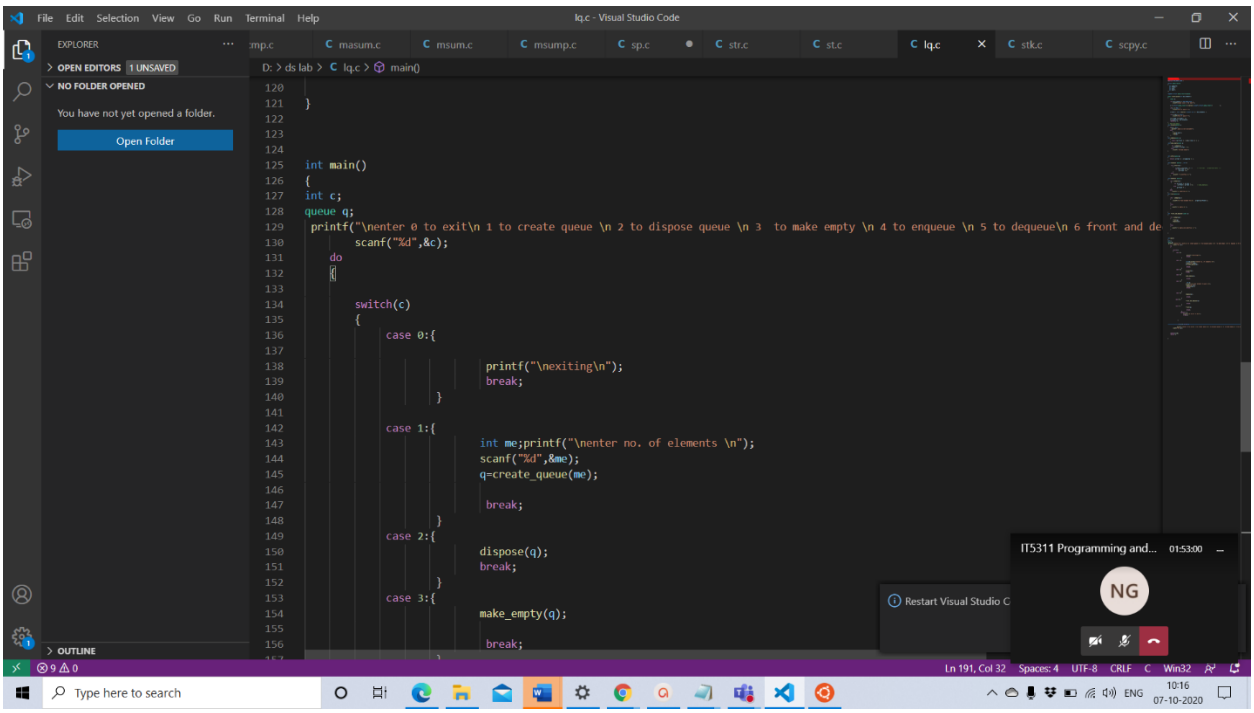
```
    if( !isEmpty(q) )
     {
          if( q->front == q->rear )
               q->front = q->rear = -1 ;     // make_empty(q);
         else
              q->front ++ ;
     }
   else
       printf("\n underflow \n ");
}
int front(queue q)
{

    if( ! isEmpty(q) )
    {
        printf("\n front element %d \n ", q->qarr[q->front] );

    }
    else
    {
       printf("\n empty \n ");
    }


}
int  front_and_dequeue( queue q)
{

    if( !isEmpty(q) )
    {
        front(q);
        dequeue(q);

    }
    else
    {
       printf("\n empty and underflow \n ");
    }


}



int main()
```

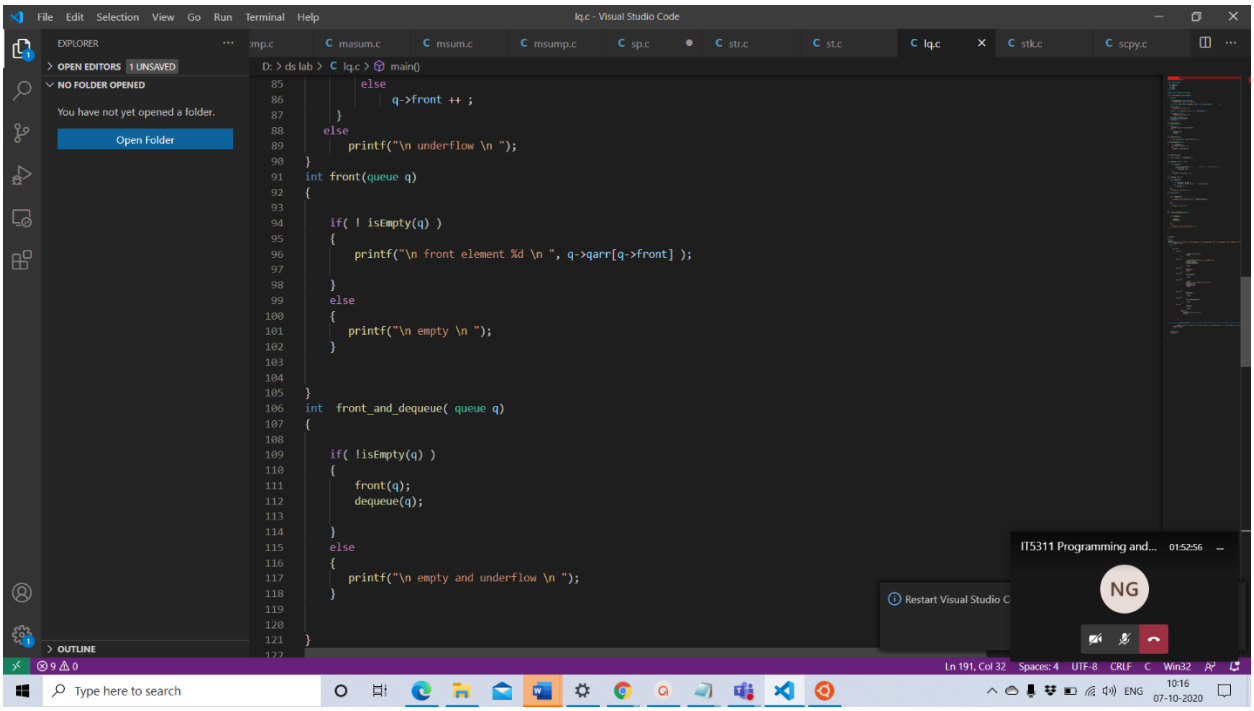
```
{
int c;
queue q;
 printf("\nenter 0 to exit\n 1 to create queue \n 2 to dispose queue \n 3  to mak
e empty \n 4 to enqueue \n 5 to dequeue\n 6 front and deque \n 7 front \n");
        scanf("%d",&c);
    do
    {

        switch(c)
        {
             case 0:{

                              printf("\nexiting\n");
                              break;
                     }

             case 1:{
                             int me;printf("\nenter no. of elements \n");
                             scanf("%d",&me);
                             q=create_queue(me);

                              break;
                     }
             case 2:{
                             dispose(q);
                             break;
                     }
             case 3:{
                             make_empty(q);

                              break;
                     }
             case 4:{
                              int n;
                             printf("\n enter element to push \n");
                             scanf("%d",&n);
                              enqueue(q,n);
                              break;


                     }
             case 5:{
                             dequeue(q);
```

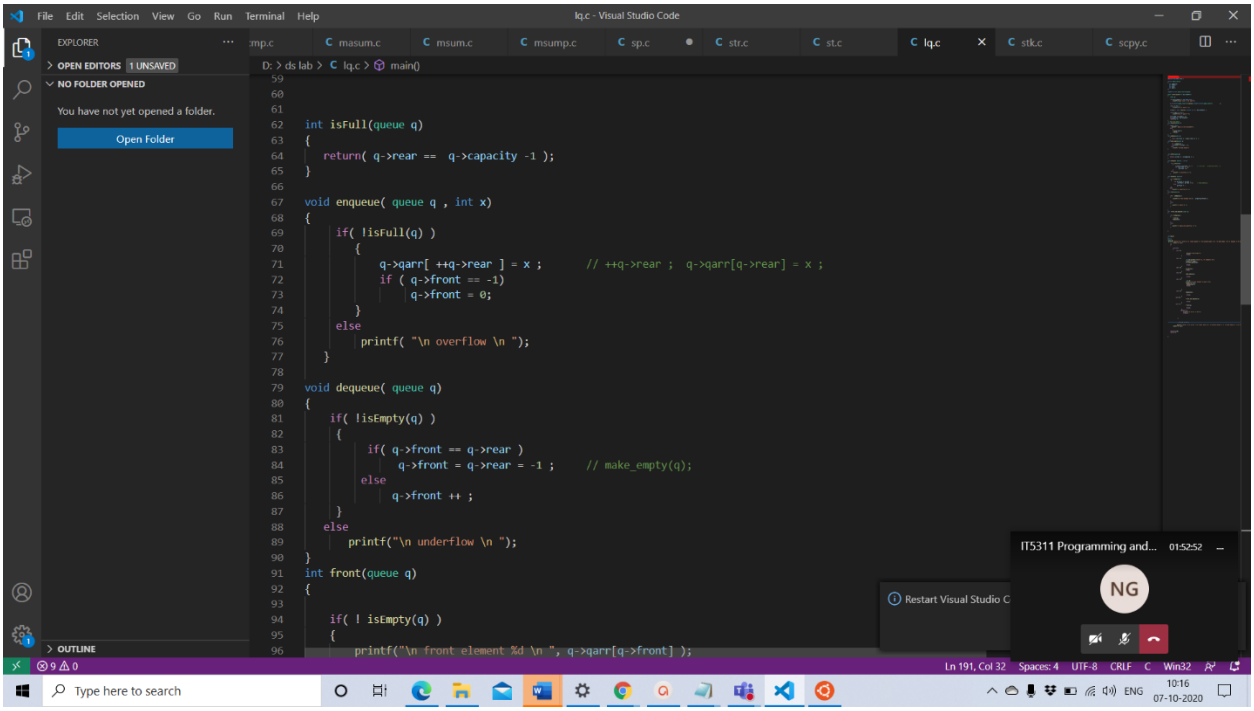
```
                            break;
                    }
            case 6:{
                            front_and_dequeue(q);

                            break;
                    }
            case 7:{
                            front(q);

                            break;
                    }
                   default: {
                       printf("\n only 0 to 5\n");
                       break;}



               }


                if(c==0) break;

              printf("\nenter 0 to exit\n 1 to create queue \n 2 to dispose queue
 \n 3  to make empty \n 4 to enqueue \n 5 to dequeue\n 6 front and deque \n 7 fro
nt \n");
        scanf("%d",&c);



    }while(c!=0);
    return 0;


}
```

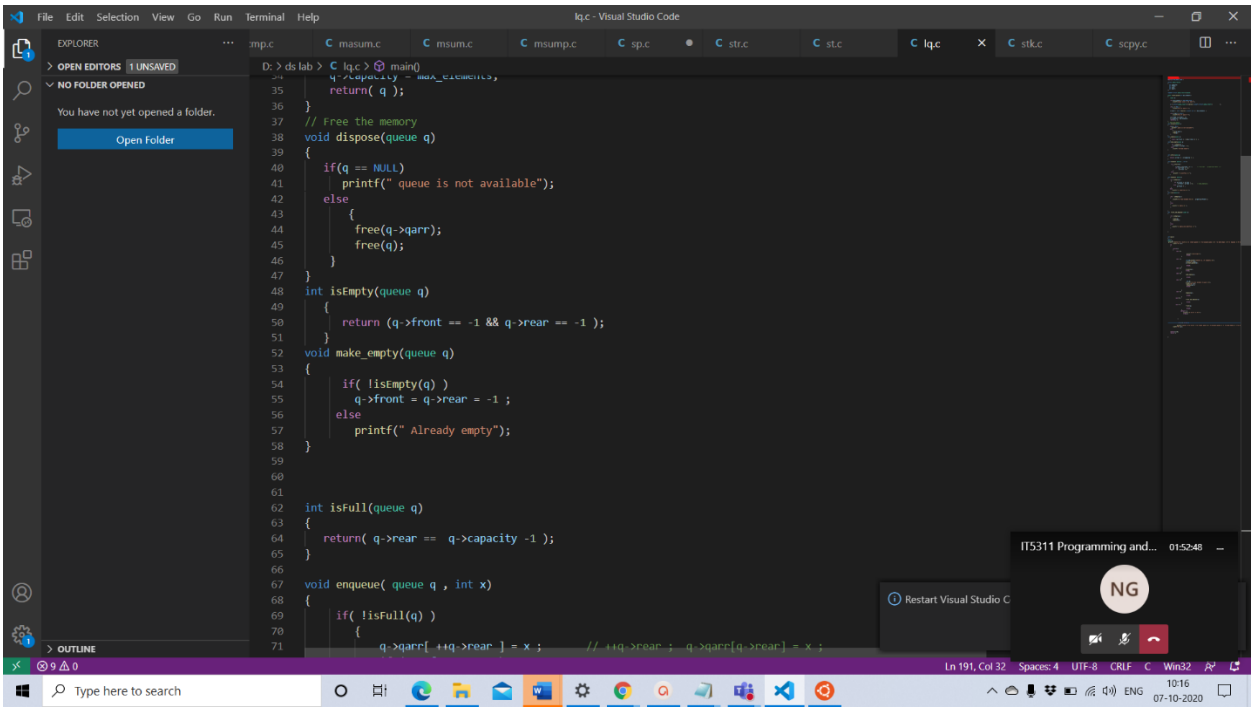
Code screenshots

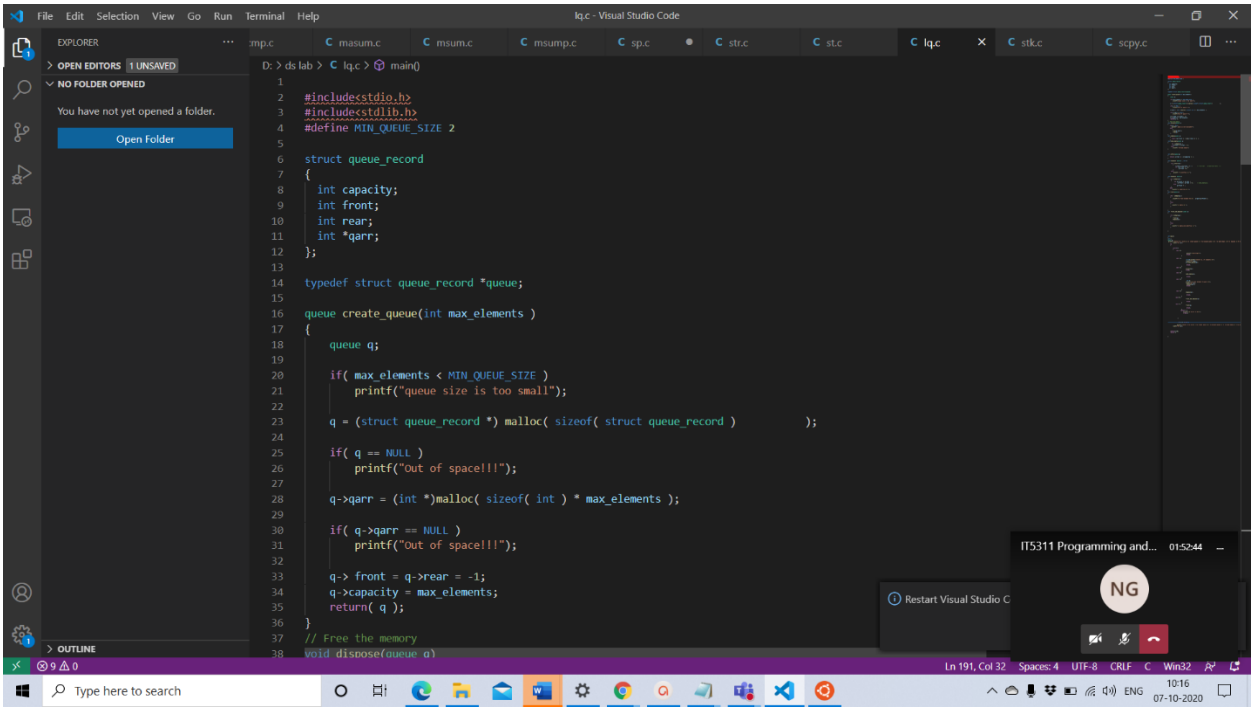

File Edit Selection View Go Run Terminal Help
lq.c - Visual Studio Code
EXPLORER
OPEN EDITORS 1 UNSAVED
NO FOLDER OPENED
You have not yet opened a folder.
Open Folder
OUTLINE
D: > ds lab > lq.c > main()
dispose(q);
break;
case 3:{
make_empty(q);
break;
case 4:{
int n;
printf("\n enter element to push \n");
scanf("%d",&n);
enqueue(q,n);
break;
case 5:{
dequeue(q);
break;
case 6:{
front_and_dequeue(q);
break;
case 7:{
front(q);
break;
default: {
printf("\n only 0 to 5\n");
break;}
Restart Visual Studio C
IT5311 Programming and... 01:53:03
NG
Ln 191, Col 32 Spaces: 4 UTF-8 CRLF C Win32
Type here to search
ENG
10:16
07-10-2020

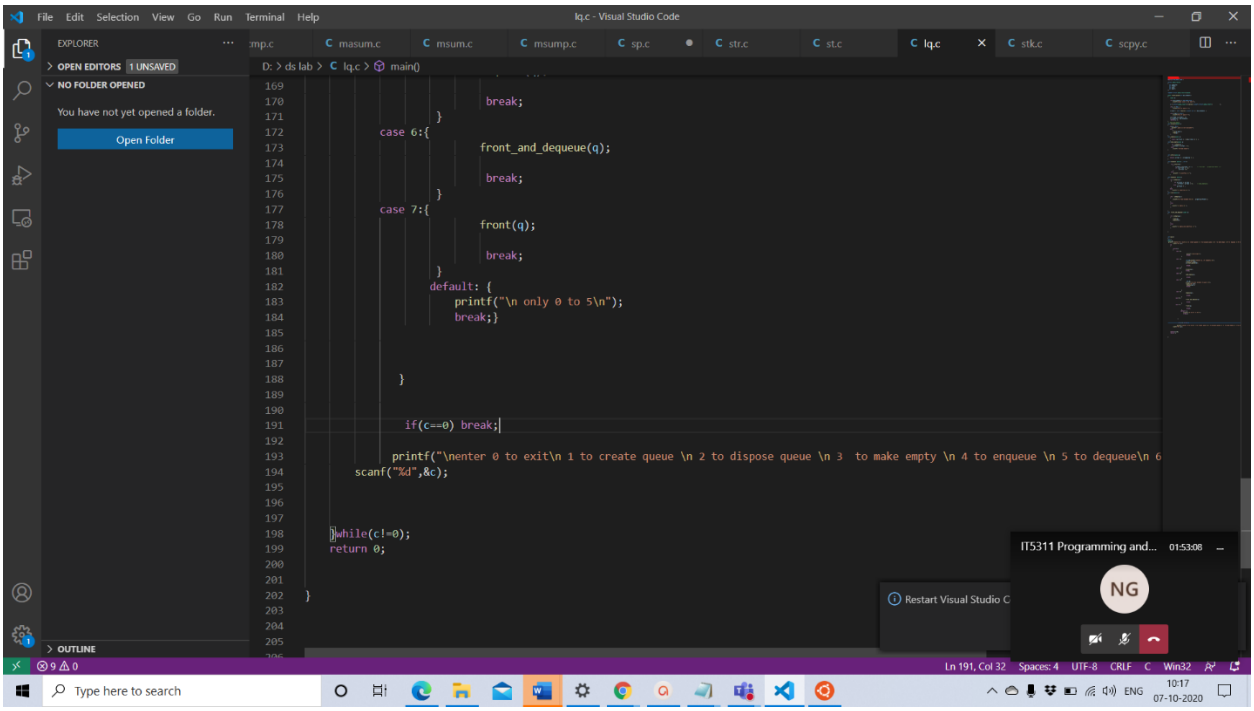

File Edit Selection View Go Run Terminal Help
lq.c - Visual Studio Code
EXPLORER
OPEN EDITORS 1 UNSAVED
NO FOLDER OPENED
You have not yet opened a folder.
Open Folder
OUTLINE
D: > ds lab > lq.c > main()
}
int main()
{
int c;
queue q;
printf("\nenter 0 to exit\n 1 to create queue \n 2 to dispose queue \n 3 to make empty \n 4 to enqueue \n 5 to dequeue\n 6 front and de
scanf("%d",&c);
do
{
switch(c)
{
case 0:{
printf("\nexiting\n");
break;
case 1:{
int me;printf("\nenter no. of elements \n");
scanf("%d",&me);
q=create_queue(me);
break;
case 2:{
dispose(q);
break;
case 3:{
make_empty(q);
break;
Restart Visual Studio C
IT5311 Programming and... 01:53:00
NG
Ln 191, Col 32 Spaces: 4 UTF-8 CRLF C Win32
Type here to search
ENG
10:16
07-10-2020


File Edit Selection View Go Run Terminal Help
lq.c - Visual Studio Code
EXPLORER
OPEN EDITORS 1 UNSAVED
NO FOLDER OPENED
You have not yet opened a folder.
Open Folder
OUTLINE
D: > ds lab > lq.c > main()
else
q->front ++ ;
}
else
printf("\n underflow \n ");
}
int front(queue q)
{
if(! isEmpty(q))
{
printf("\n front element %d \n ", q->qarr[q->front]);
}
else
{
printf("\n empty \n ");
}
}
int front_and_dequeue(queue q)
{
if(!isEmpty(q))
{
front(q);
dequeue(q);
}
else
{
printf("\n empty and underflow \n ");
}
}
IT5311 Programming and... 01:52:56
NG
Restart Visual Studio C
Ln 191, Col 32 Spaces: 4 UTF-8 CRLF C Win32
Type here to search
ENG
10:16
07-10-2020

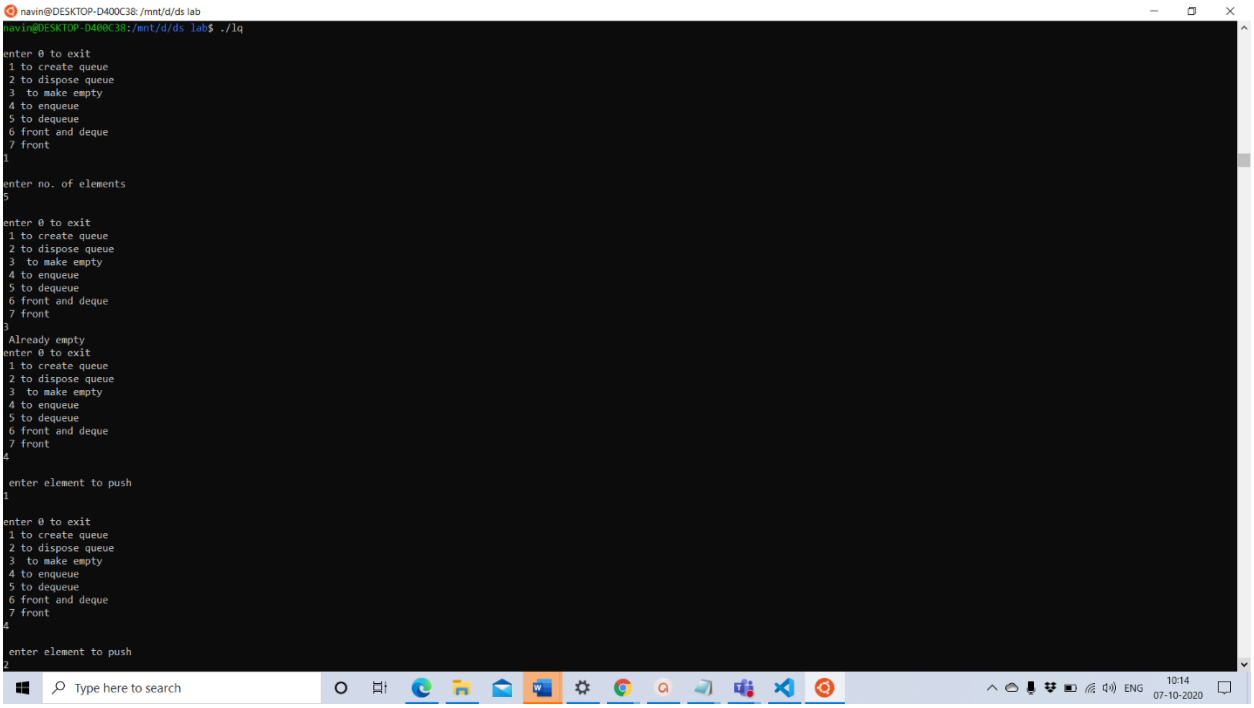

File Edit Selection View Go Run Terminal Help
lq.c - Visual Studio Code
EXPLORER
OPEN EDITORS 1 UNSAVED
NO FOLDER OPENED
You have not yet opened a folder.
Open Folder
OUTLINE
D: > ds lab > lq.c > main()
int isFull(queue q)
{
return(q->rear == q->capacity -1);
}
void enqueue(queue q , int x)
{
if(!isFull(q))
{
q->qarr[++q->rear] = x ; // ++q->rear ; q->qarr[q->rear] = x ;
if (q->front == -1)
q->front = 0;
}
else
printf("\n overflow \n ");
}
void dequeue(queue q)
{
if(!isEmpty(q))
{
if(q->front == q->rear)
q->front = q->rear = -1 ; // make_empty(q);
else
q->front ++ ;
}
else
printf("\n underflow \n ");
}
int front(queue q)
{
if(! isEmpty(q))
{
printf("\n front element %d \n ", q->qarr[q->front]);
IT5311 Programming and... 01:52:52
NG
Restart Visual Studio C
Ln 191, Col 32 Spaces: 4 UTF-8 CRLF C Win32
Type here to search
ENG
10:16
07-10-2020

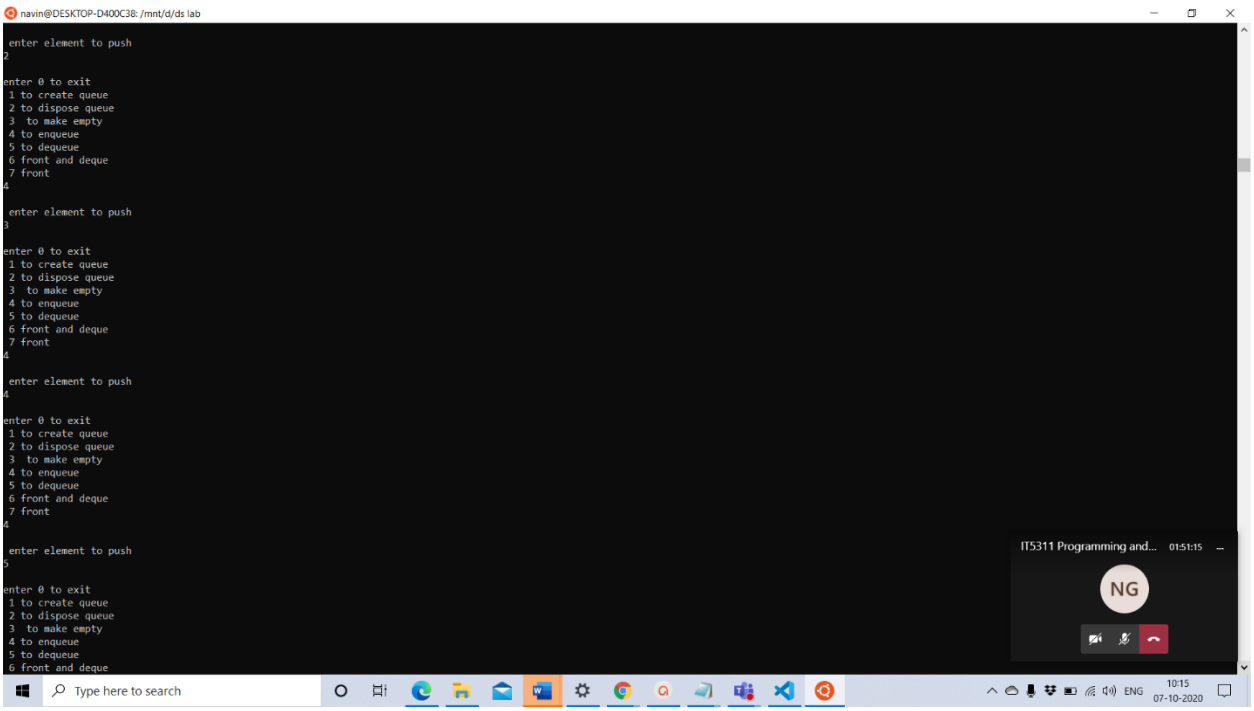

File Edit Selection View Go Run Terminal Help
lq.c - Visual Studio Code
EXPLORER
OPEN EDITORS 1 UNSAVED
NO FOLDER OPENED
You have not yet opened a folder.
Open Folder
OUTLINE
D: > ds lab > lq.c > main()
return(q);
}
// Free the memory
void dispose(queue q)
{
if(q == NULL)
printf(" queue is not available");
else
{
free(q->qarr);
free(q);
}
}
int isEmpty(queue q)
{
return (q->front == -1 && q->rear == -1);
}
void make_empty(queue q)
{
if(!isEmpty(q))
q->front = q->rear = -1 ;
else
printf(" Already empty");
}
int isFull(queue q)
{
return(q->rear == q->capacity -1);
}
void enqueue(queue q , int x)
{
if(!isFull(q))
{
q->qarr[++q->rear] = x ;
Restart Visual Studio C
IT5311 Programming and...
01:52:48
NG
Ln 191, Col 32 Spaces: 4 UTF-8 CRLF C Win32
Type here to search
ENG
10:16
07-10-2020

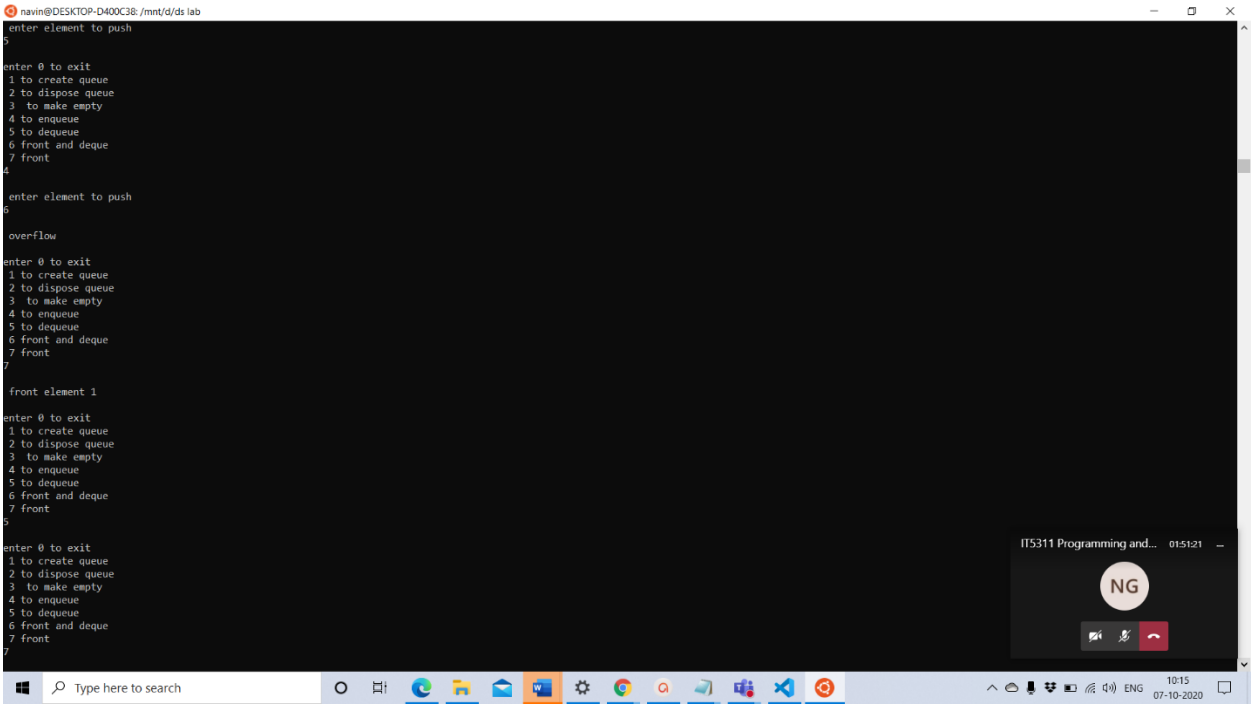

File Edit Selection View Go Run Terminal Help
lq.c - Visual Studio Code
EXPLORER
OPEN EDITORS 1 UNSAVED
NO FOLDER OPENED
You have not yet opened a folder.
Open Folder
OUTLINE
D: > ds lab > lq.c > main()
#include<stdio.h>
#include<stdlib.h>
#define MIN_QUEUE_SIZE 2
struct queue_record
{
int capacity;
int front;
int rear;
int *qarr;
};
typedef struct queue_record *queue;
queue create_queue(int max_elements)
{
queue q;
if(max_elements < MIN_QUEUE_SIZE)
printf("queue size is too small");
q = (struct queue_record *) malloc(sizeof(struct queue_record));
if(q == NULL)
printf("Out of space!!!");
q->qarr = (int *)malloc(sizeof(int) * max_elements);
if(q->qarr == NULL)
printf("Out of space!!!");
q-> front = q->rear = -1;
q->capacity = max_elements;
return(q);
}
// Free the memory
void dispose(queue q)
Restart Visual Studio C
IT5311 Programming and...
01:52:44
NG
Ln 191, Col 32 Spaces: 4 UTF-8 CRLF C Win32
Type here to search
ENG
10:16
07-10-2020

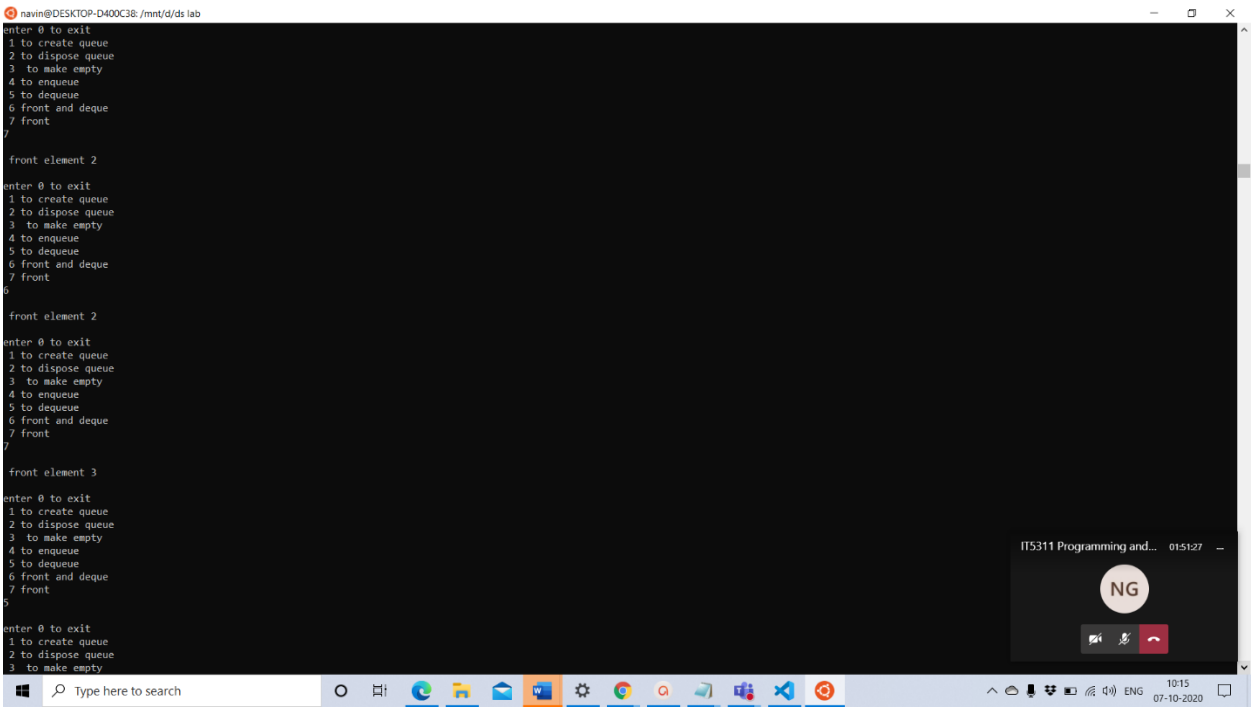


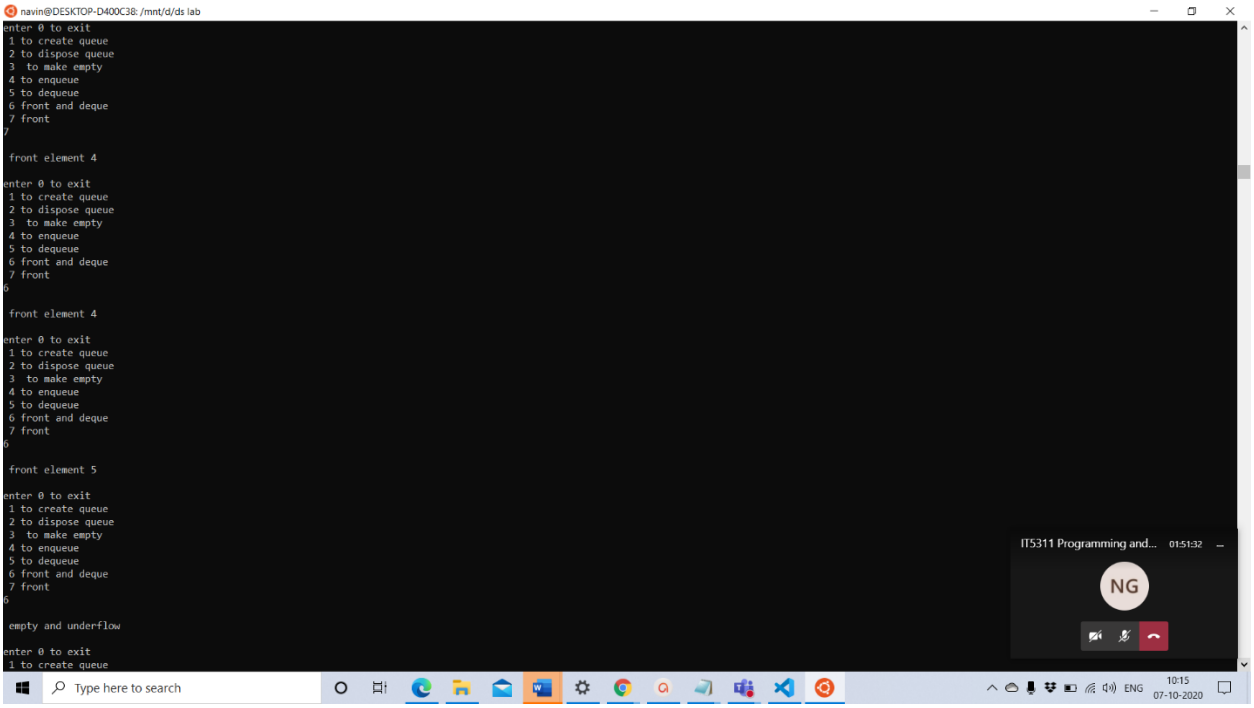
Output screenshots

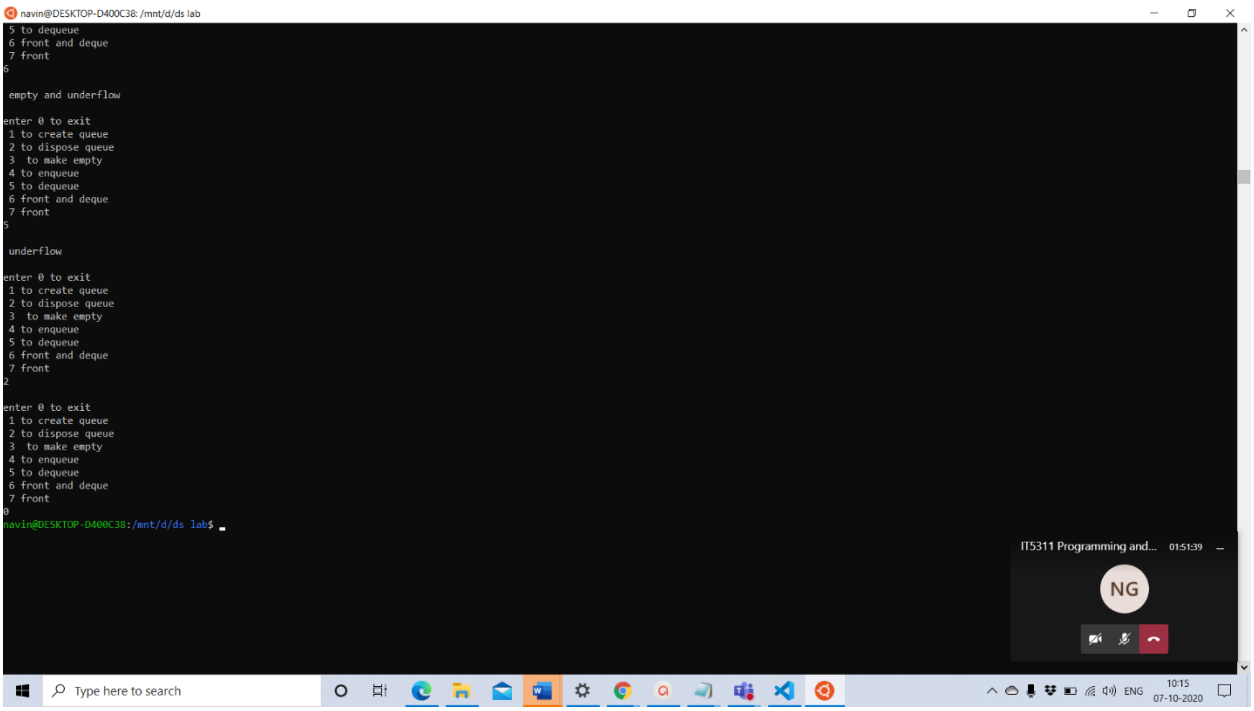

navin@DESKTOP-D400C38: /mnt/d/ds lab
navin@DESKTOP-D400C38:/mnt/d/ds lab$ ./lq
enter 0 to exit
1 to create queue
2 to dispose queue
3 to make empty
4 to enqueue
5 to dequeue
6 front and deque
7 front
1
enter no. of elements
5
enter 0 to exit
1 to create queue
2 to dispose queue
3 to make empty
4 to enqueue
5 to dequeue
6 front and deque
7 front
3
Already empty
enter 0 to exit
1 to create queue
2 to dispose queue
3 to make empty
4 to enqueue
5 to dequeue
6 front and deque
7 front
4
enter element to push
1
enter 0 to exit
1 to create queue
2 to dispose queue
3 to make empty
4 to enqueue
5 to dequeue
6 front and deque
7 front
4
enter element to push
2
Type here to search
ENG
10:14
07-10-2020


navin@DESKTOP-D400C38: /mnt/d/ds lab
enter element to push
2
enter 0 to exit
1 to create queue
2 to dispose queue
3 to make empty
4 to enqueue
5 to dequeue
6 front and deque
7 front
4
enter element to push
3
enter 0 to exit
1 to create queue
2 to dispose queue
3 to make empty
4 to enqueue
5 to dequeue
6 front and deque
7 front
4
enter element to push
4
enter 0 to exit
1 to create queue
2 to dispose queue
3 to make empty
4 to enqueue
5 to dequeue
6 front and deque
7 front
4
enter element to push
5
enter 0 to exit
1 to create queue
2 to dispose queue
3 to make empty
4 to enqueue
5 to dequeue
6 front and deque
IT5311 Programming and...
01:51:15
NG
Type here to search
ENG
10:15
07-10-2020


navin@DESKTOP-D400C38: /mnt/d/ds lab
enter element to push
5
enter 0 to exit
1 to create queue
2 to dispose queue
3 to make empty
4 to enqueue
5 to dequeue
6 front and deque
7 front
4
enter element to push
6
overflow
enter 0 to exit
1 to create queue
2 to dispose queue
3 to make empty
4 to enqueue
5 to dequeue
6 front and deque
7 front
7
front element 1
enter 0 to exit
1 to create queue
2 to dispose queue
3 to make empty
4 to enqueue
5 to dequeue
6 front and deque
7 front
5
enter 0 to exit
1 to create queue
2 to dispose queue
3 to make empty
4 to enqueue
5 to dequeue
6 front and deque
7 front
7
IT5311 Programming and... 01:51:21
NG
Type here to search
ENG
10:15
07-10-2020


navin@DESKTOP-D400C38: /mnt/d/ds lab
enter 0 to exit
1 to create queue
2 to dispose queue
3 to make empty
4 to enqueue
5 to dequeue
6 front and deque
7 front
7
front element 2
enter 0 to exit
1 to create queue
2 to dispose queue
3 to make empty
4 to enqueue
5 to dequeue
6 front and deque
7 front
6
front element 2
enter 0 to exit
1 to create queue
2 to dispose queue
3 to make empty
4 to enqueue
5 to dequeue
6 front and deque
7 front
7
front element 3
enter 0 to exit
1 to create queue
2 to dispose queue
3 to make empty
4 to enqueue
5 to dequeue
6 front and deque
7 front
5
enter 0 to exit
1 to create queue
2 to dispose queue
3 to make empty
IT5311 Programming and... 01:51:27
NG
Type here to search
ENG
10:15
07-10-2020


navin@DESKTOP-D400C38: /mnt/d/ds lab
enter 0 to exit
1 to create queue
2 to dispose queue
3 to make empty
4 to enqueue
5 to dequeue
6 front and deque
7 front
7
front element 4
enter 0 to exit
1 to create queue
2 to dispose queue
3 to make empty
4 to enqueue
5 to dequeue
6 front and deque
7 front
6
front element 4
enter 0 to exit
1 to create queue
2 to dispose queue
3 to make empty
4 to enqueue
5 to dequeue
6 front and deque
7 front
6
front element 5
enter 0 to exit
1 to create queue
2 to dispose queue
3 to make empty
4 to enqueue
5 to dequeue
6 front and deque
7 front
6
empty and underflow
enter 0 to exit
1 to create queue
IT5311 Programming and... 01:51:32
NG
Type here to search
ENG
10:15
07-10-2020


navin@DESKTOP-D400C38: /mnt/d/ds lab
5 to dequeue
6 front and deque
7 front
6
empty and underflow
enter 0 to exit
1 to create queue
2 to dispose queue
3 to make empty
4 to enqueue
5 to dequeue
6 front and deque
7 front
5
underflow
enter 0 to exit
1 to create queue
2 to dispose queue
3 to make empty
4 to enqueue
5 to dequeue
6 front and deque
7 front
2
enter 0 to exit
1 to create queue
2 to dispose queue
3 to make empty
4 to enqueue
5 to dequeue
6 front and deque
7 front
0
navin@DESKTOP-D400C38:/mnt/d/ds lab$
IT5311 Programming and... 01:51:39
NG
Type here to search
ENG
10:15
07-10-2020