LAB 14/10/20

SWAMINATHAN NAVINASHOK

2019115126

CIRCULAR QUEUE

SOURCE CODE

```c
#include<stdio.h>
#include<stdlib.h>
#define MIN_QUEUE_SIZE 5

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");
```

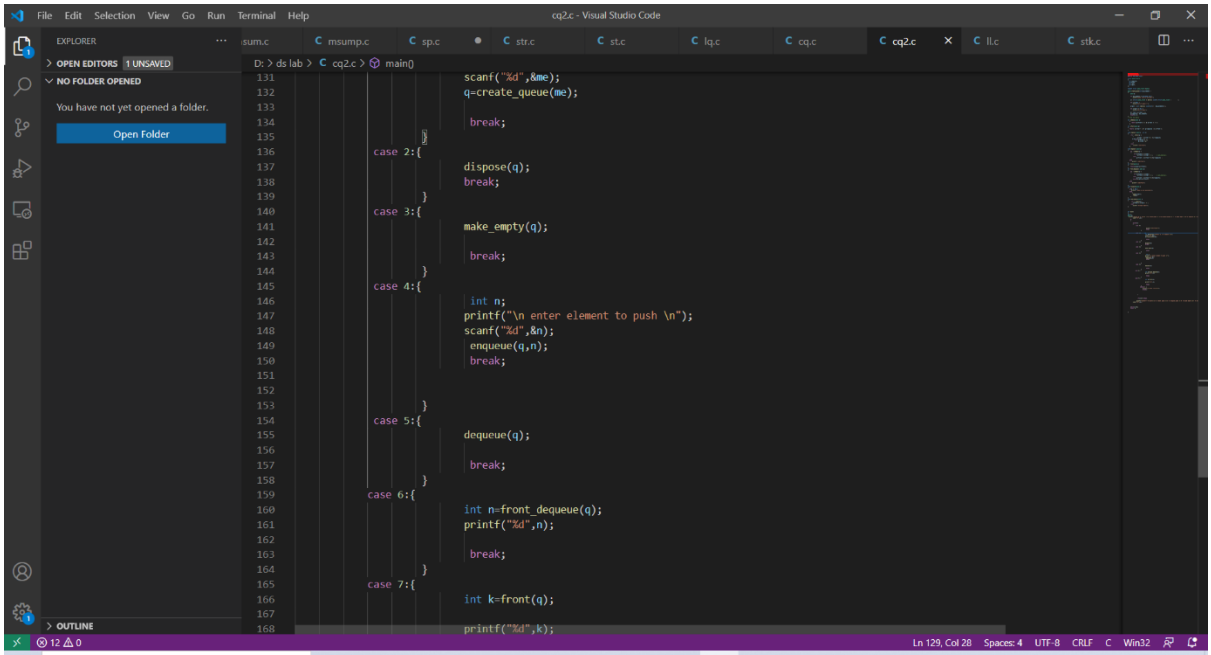
```
    q-> front = q->rear = -1;
    q->capacity = max_elements;
    return( q );
}
// Free the memory

int isEmpty(queue q)
   {
      return (q->front == -1 && q->rear == -1 );
   }

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

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

void dequeue( queue q)
{
    if( ! isEmpty(q) )
     {
          if( q->front == q->rear )
               q->front = q->rear = -1 ;     // make_empty(q);
         else
              q->front = (q->front +1 )%(q->capacity) ;
     }
   else
       printf(" underflow");
}
int front(queue q)
{
    return(q->qarr[q->front]);
}
int front_dequeue( queue q)
{
    if( ! isEmpty(q) )
```

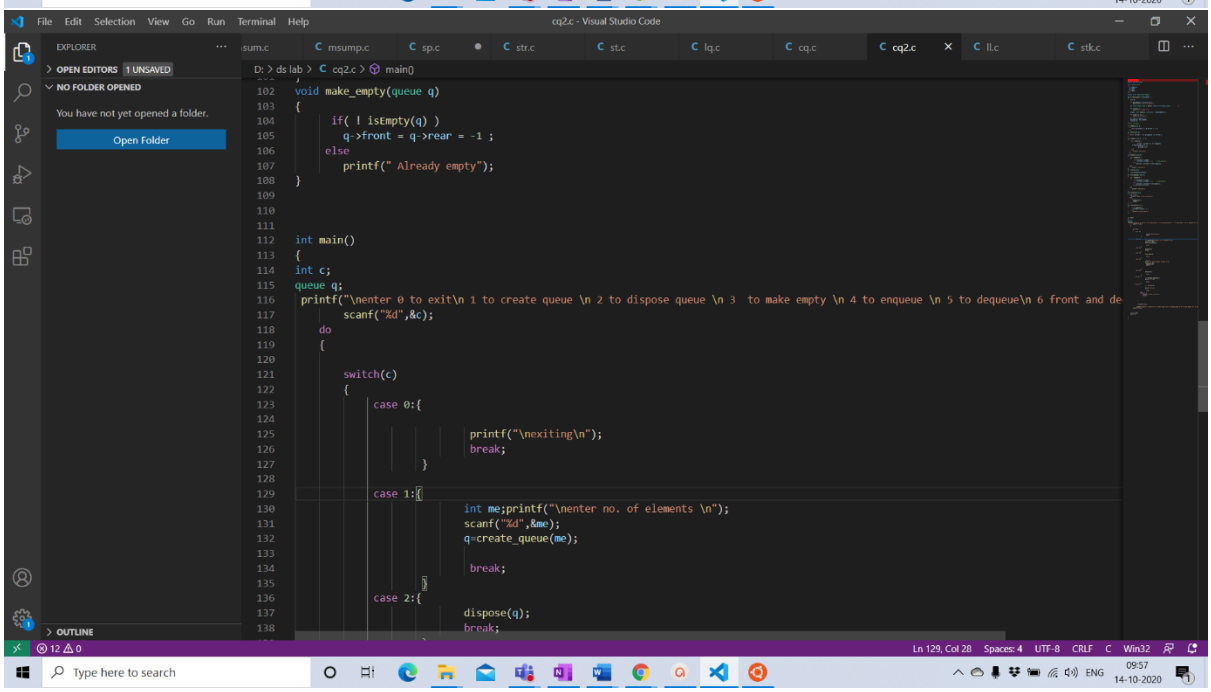
```
     {
          if( q->front == q->rear )
               q->front = q->rear = -1 ;     // make_empty(q);
          else
               q->front = (q->front +1 )%q->capacity ;
        return(q->qarr[q->front]);
     }
   else
       printf(" underflow");

}
void dispose(queue q)
{
   if(q == NULL)
      printf(" queue is not available");
   else
       {
        free(q->qarr);
        free(q);
    }
}
void make_empty(queue q)
{
      if( ! isEmpty(q) )
        q->front = q->rear = -1 ;
     else
        printf(" Already empty");
}



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 deque \n 7 front \n")
;
        scanf("%d",&c);
    do
    {

        switch(c)
        {
             case 0:{

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

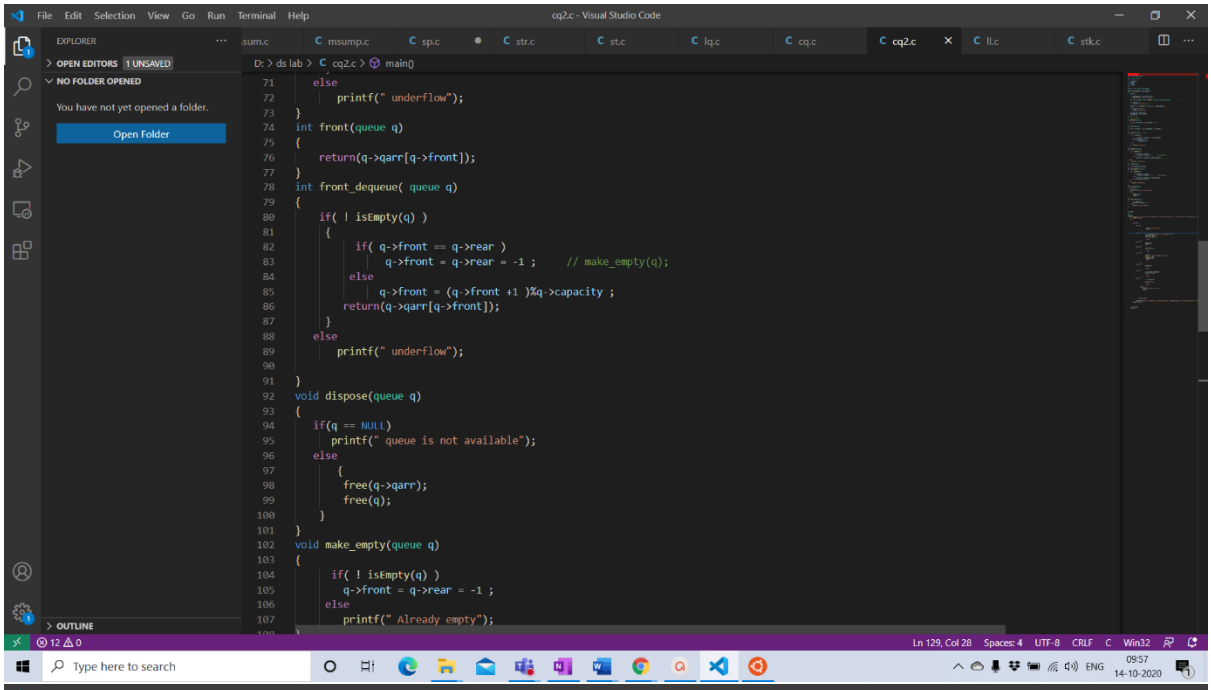
```
                }

        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:{
                        int n=front_dequeue(q);
                        printf("%d",n);

                         break;
                }
       case 7:{
                        int k=front(q);

                        printf("%d",k);

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

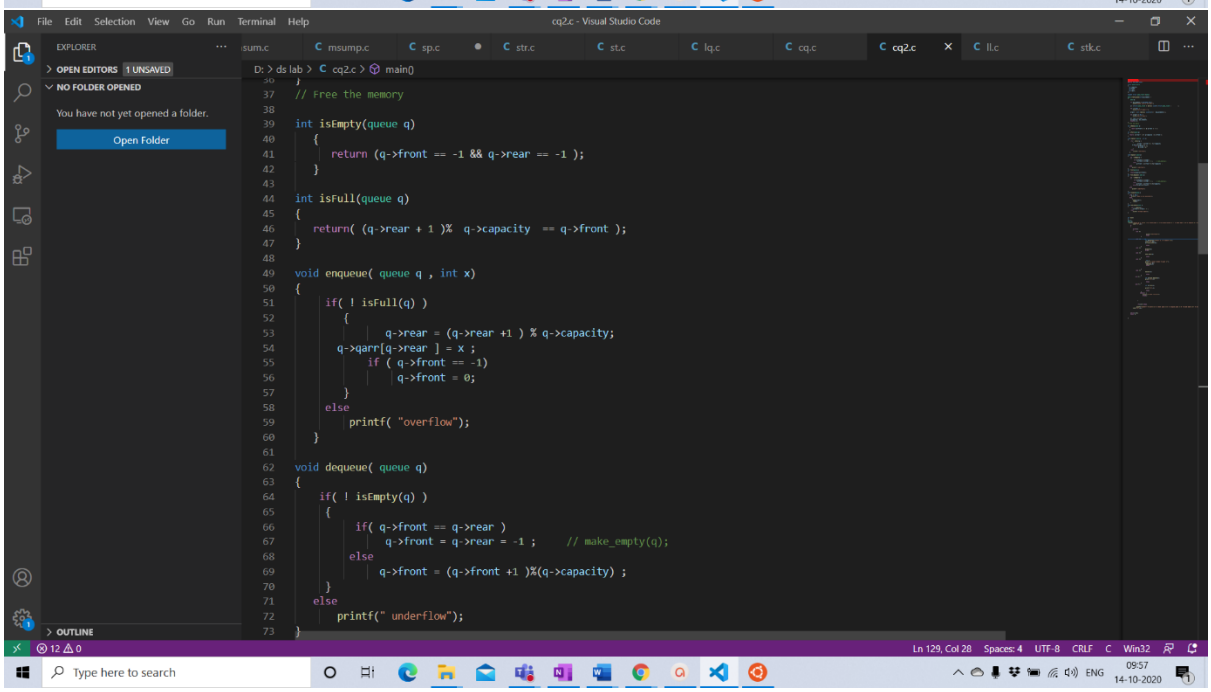
```
                }


                 if(c==0) break;

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



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


}
```

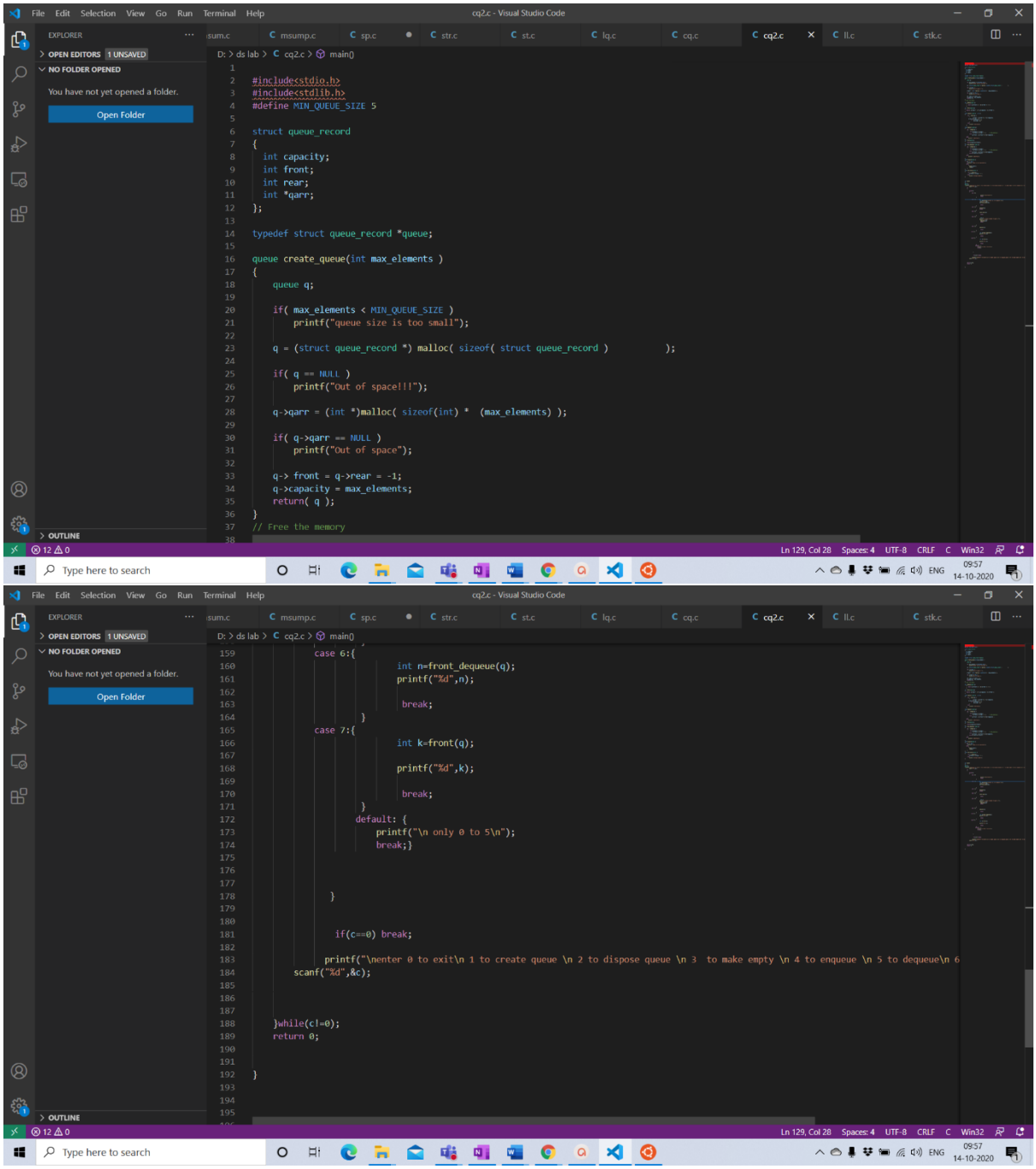
SCREENSHOT OF SOURCE CODE


File Edit Selection View Go Run Terminal Help
cq2.c - Visual Studio Code
EXPLORER
OPEN EDITORS 1 UNSAVED
NO FOLDER OPENED
You have not yet opened a folder.
Open Folder
OUTLINE
sum.c
msump.c
sp.c
str.c
st.c
lq.c
cq.c
cq2.c
ll.c
stk.c
D: > ds lab > cq2.c > main()
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:{
int n=front_dequeue(q);
printf("%d",n);
break;
case 7:{
int k=front(q);
printf("%d",k);
Ln 129, Col 28 Spaces: 4 UTF-8 CRLF C Win32
Type here to search
ENG
09:57
14-10-2020


File Edit Selection View Go Run Terminal Help
cq2.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 > cq2.c > main()
void make_empty(queue q)
{
if(! isEmpty(q))
q->front = q->rear = -1 ;
else
printf(" Already empty");
}
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;
Ln 129, Col 28 Spaces: 4 UTF-8 CRLF C Win32
Type here to search
ENG
09:57
14-10-2020


File Edit Selection View Go Run Terminal Help
cq2.c - Visual Studio Code
EXPLORER
OPEN EDITORS 1 UNSAVED
NO FOLDER OPENED
You have not yet opened a folder.
Open Folder
OUTLINE
sum.c
msump.c
sp.c
str.c
st.c
lq.c
cq.c
cq2.c
ll.c
stk.c
D: > ds lab > cq2.c > main()
else
printf(" underflow");
}
int front(queue q)
{
return(q->qarr[q->front]);
}
int front_dequeue(queue q)
{
if(! isEmpty(q))
{
if(q->front == q->rear)
q->front = q->rear = -1 ; // make_empty(q);
else
q->front = (q->front +1)%q->capacity ;
return(q->qarr[q->front]);
}
else
printf(" underflow");
}
void dispose(queue q)
{
if(q == NULL)
printf(" queue is not available");
else
{
free(q->qarr);
free(q);
}
}
void make_empty(queue q)
{
if(! isEmpty(q))
q->front = q->rear = -1 ;
else
printf(" Already empty");
}
Ln 129, Col 28 Spaces: 4 UTF-8 CRLF C Win32
Type here to search
ENG
09:57
14-10-2020


File Edit Selection View Go Run Terminal Help
cq2.c - Visual Studio Code
EXPLORER
OPEN EDITORS 1 UNSAVED
NO FOLDER OPENED
You have not yet opened a folder.
Open Folder
OUTLINE
sum.c
msump.c
sp.c
str.c
st.c
lq.c
cq.c
cq2.c
ll.c
stk.c
D: > ds lab > cq2.c > main()
}
// Free the memory
int isEmpty(queue q)
{
return (q->front == -1 && q->rear == -1);
}
int isFull(queue q)
{
return((q->rear + 1)% q->capacity == q->front);
}
void enqueue(queue q , int x)
{
if(! isFull(q))
{
q->rear = (q->rear +1) % q->capacity;
q->qarr[q->rear] = x ;
if (q->front == -1)
q->front = 0;
}
else
printf("overflow");
}
void dequeue(queue q)
{
if(! isEmpty(q))
{
if(q->front == q->rear)
q->front = q->rear = -1 ; // make_empty(q);
else
q->front = (q->front +1)%(q->capacity) ;
}
else
printf(" underflow");
}
Ln 129, Col 28 Spaces: 4 UTF-8 CRLF C Win32
Type here to search
ENG
09:57
14-10-2020

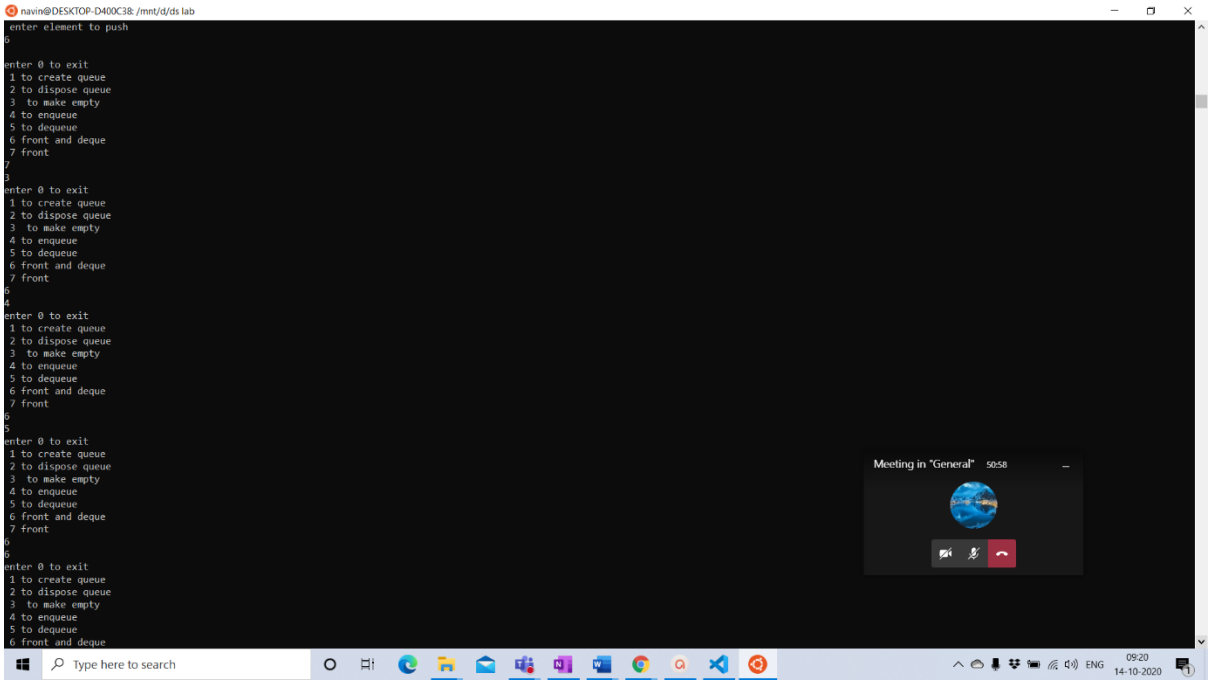


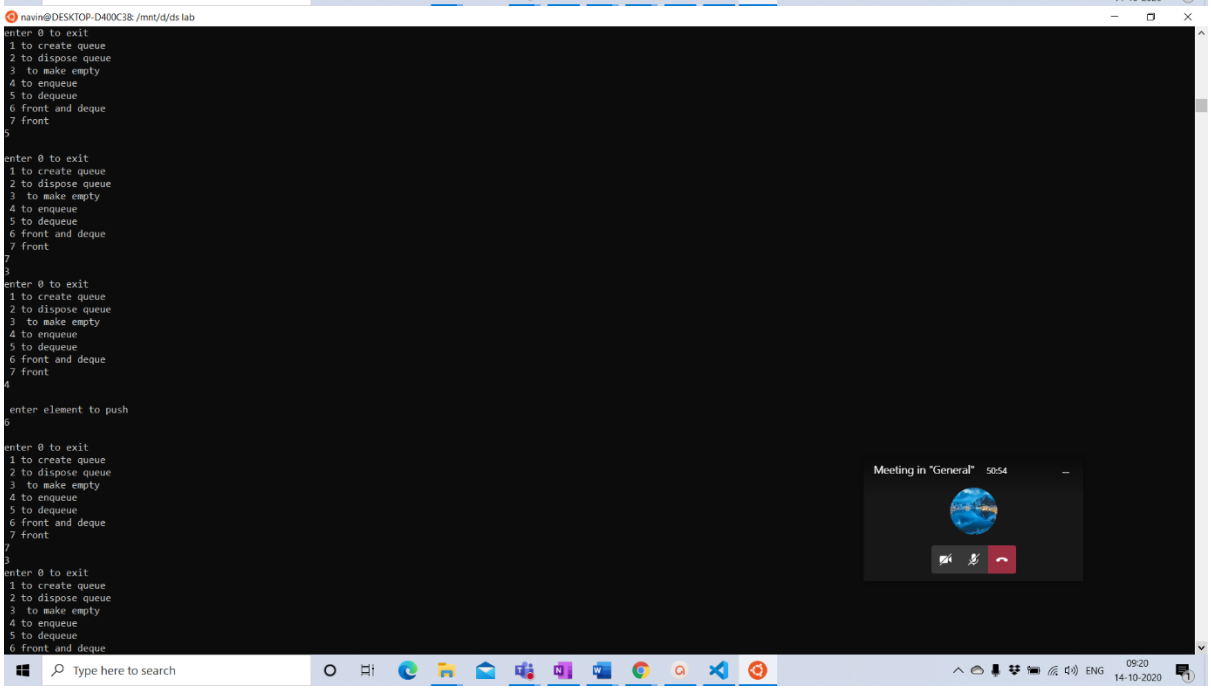
```c
#include<stdio.h>
#include<stdlib.h>
#define MIN_QUEUE_SIZE 5

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
```

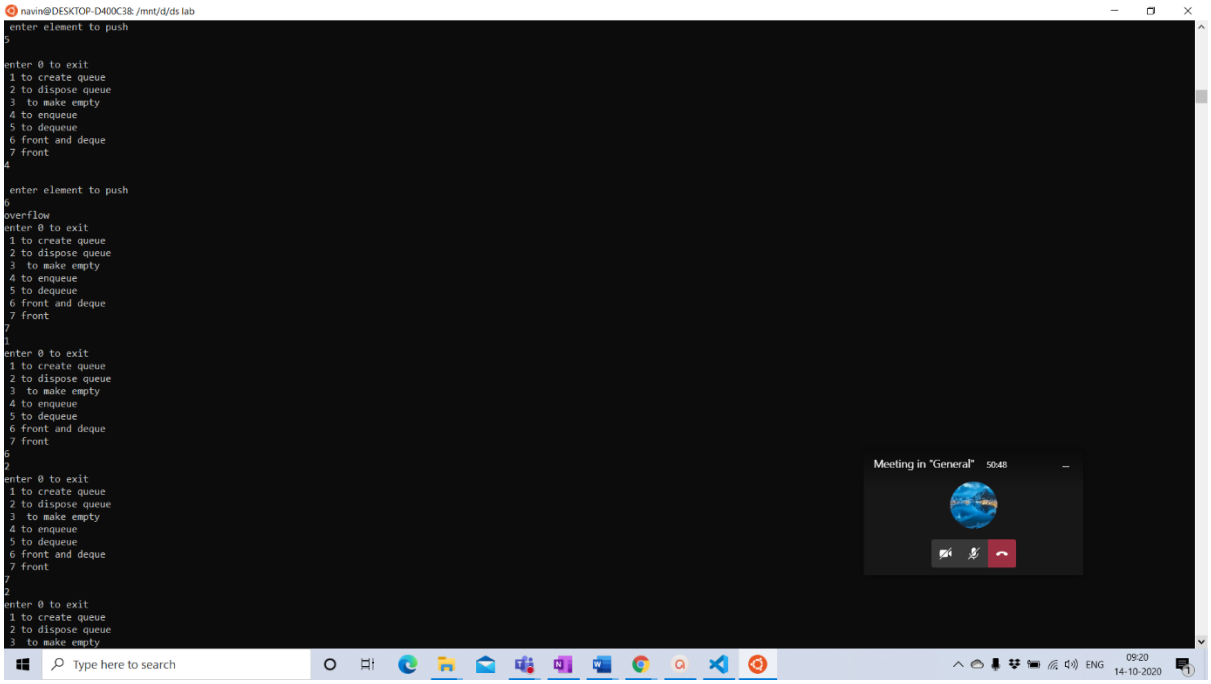


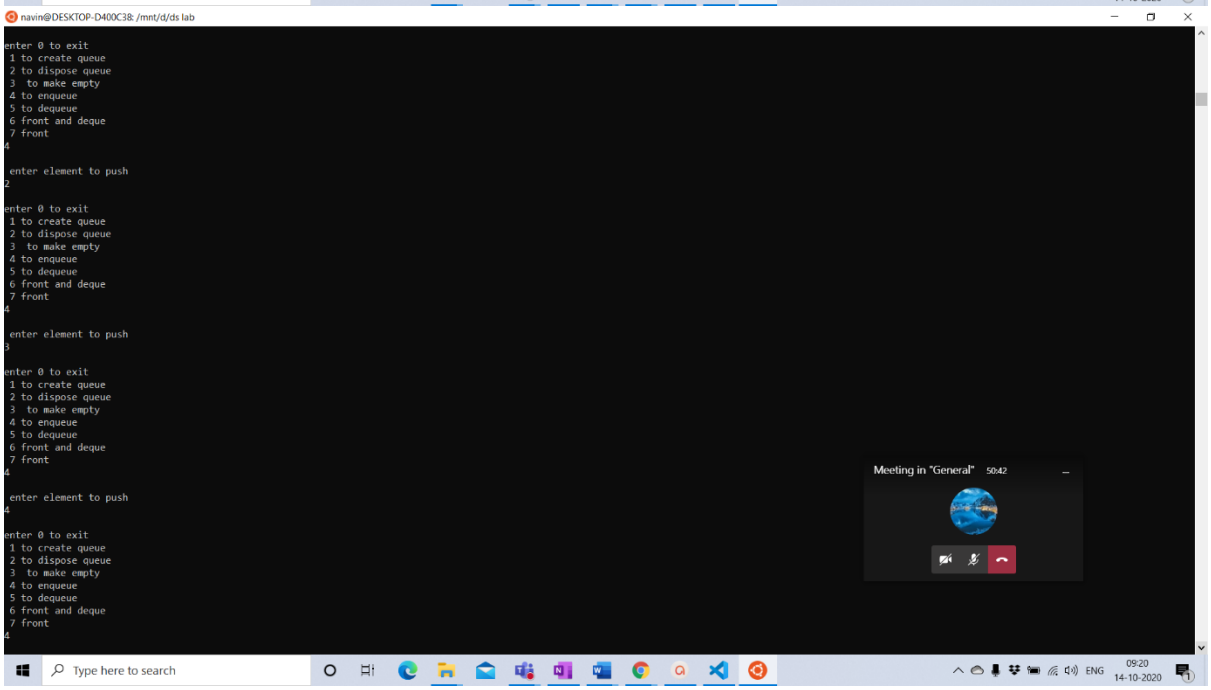
```c
            case 6:{
                        int n=front_dequeue(q);
                        printf("%d",n);

                        break;
                    }
            case 7:{
                        int k=front(q);

                        printf("%d",k);

                        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
        scanf("%d",&c);


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


}
```

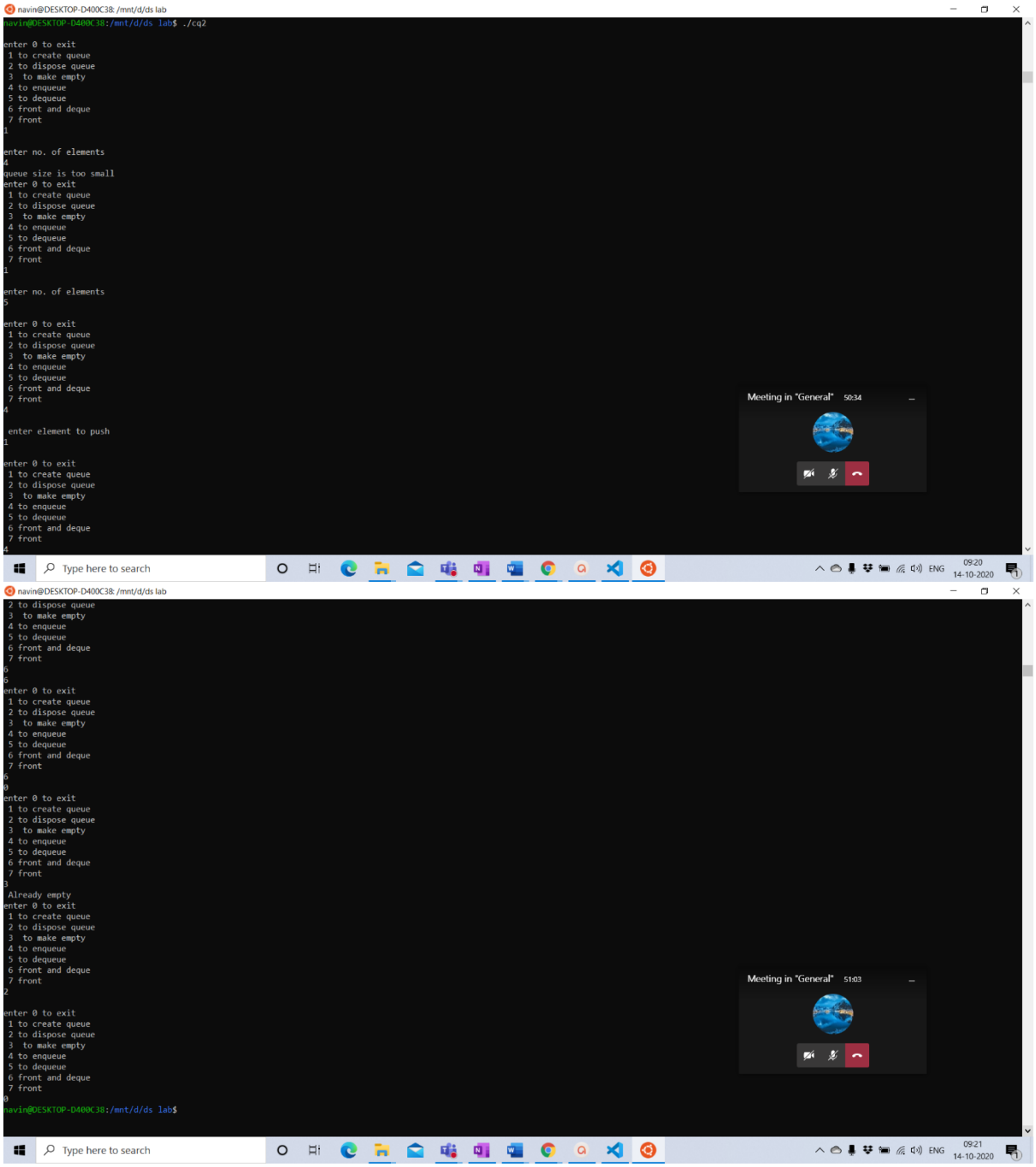
SCREENSHOT OF OUTPUT


navin@DESKTOP-D400C38: /mnt/d/ds lab
enter element to push
6
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
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
6
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
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
6
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
Meeting in "General" 50:58
Type here to search
ENG
09:20
14-10-2020


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
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
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
6
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
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
Meeting in "General" 50:54
Type here to search
ENG
09:20
14-10-2020


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
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
6
7
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
2
enter 0 to exit
1 to create queue
2 to dispose queue
3 to make empty
Meeting in "General" 50:48
Type here to search
ENG
09:20
14-10-2020


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
4
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
Meeting in "General" 50:42
Type here to search
ENG
09:20
14-10-2020



LINKED LIST

SOURCE CODE

```c
#include<stdio.h>
#include<stdlib.h>
struct noderec
{
     int info;
      struct noderec *next ;
};
typedef struct noderec * node ;
node start, last, new, temp, prev ;
void create()
{
    start = last = NULL ;
}
int isEmpty()
{
   return start ==NULL ;
}
void insertEnd(int x)
{
    new = (struct noderec*)malloc(sizeof(struct noderec) );
         new -> info = x;
         new -> next = NULL;
         if( isEmpty())
        start= last= new ;
         else
        {
            last->next = new ;
            last = new ;
        }
   }

void insertBeg(int x)
{
    new = (struct noderec*)malloc(sizeof(struct noderec) );
         new -> info = x;
    if(isEmpty())
    {
        new->next = NULL ;
        start = last = new ;
        }
    else
    {
        new->next = start ;
        start = new ;
    }
}

void insertMid(int x, int pos)
```

```
{
    int i = 1;
    new = (struct noderec*)malloc(sizeof(struct noderec) );
         new -> info = x;
    if( isEmpty())
    {
        new -> next = NULL;
        start = last = new;
    }
    else
     {
        temp = start ;
        while( temp != NULL && i<pos-1)
        {
            temp = temp->next;
            i++ ;
        }
        if( pos == 1)
        {
            new-> next = start ;
            start = new ;
        }
        else if ( temp == NULL)
        {
            new -> next = NULL;
            last -> next = new;
            last = new ;
        }
        else
        {
            new -> next = temp -> next ;
            temp->next = new ;
        }
    }
  }

void display()
{
    if( ! isEmpty())
    {
        temp = start ;
        while( temp != NULL)
        {
            printf("%d \n", temp->info );
            temp = temp ->next ;
        }
      }
    else
```

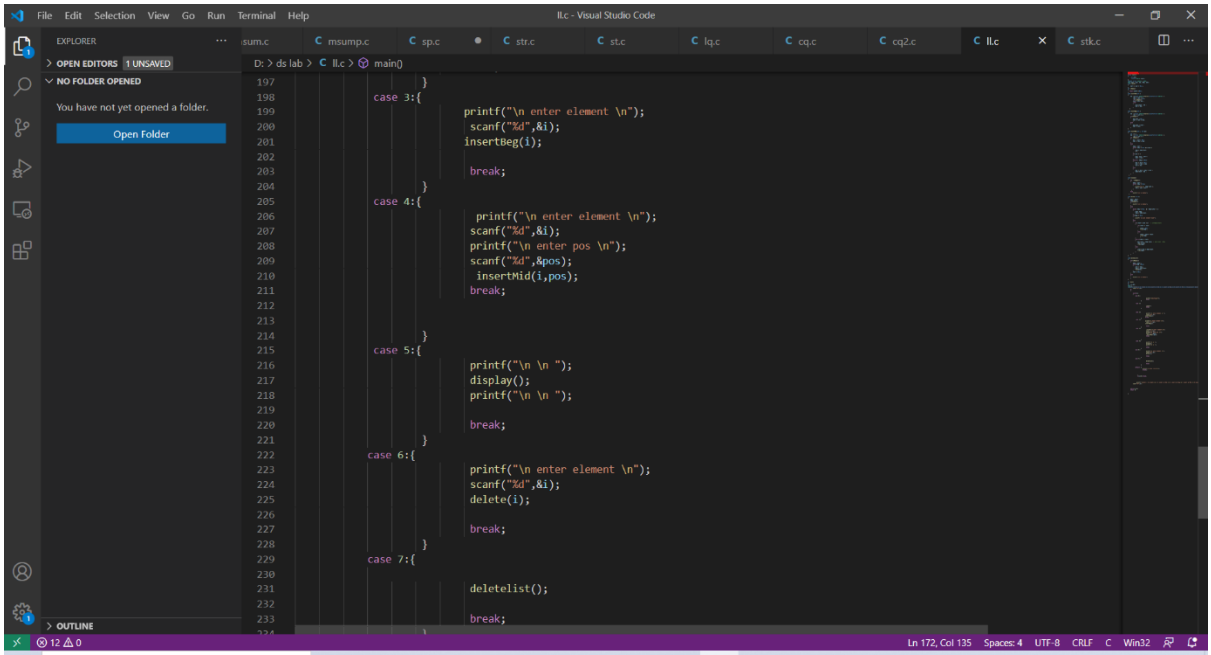
```
        printf("list is empty");
  }

void delete( int x)
{
    prev = start;
    temp= start;
    if(isEmpty())
    {
        printf("list is empty");
    }
    else
    {
        while( temp != NULL  &&  temp->info != x)
        {
            prev= temp;
            temp = temp->next ;
        }
        if(temp == NULL)
        {
            printf(" no such element found");
        }
        else
        {
            if( start-> info ==x)   // if(temp==start)
            {
                if( start == last)
                {
                    start = NULL ;
                    free(temp) ;
                }
                else
                {
                    start = start ->next;
                    free(temp);
                    }
            }
            else if(temp == last)
            {
               prev->next = temp->next; // prev->next = NULL
                last = prev;
                free(temp);
            }
            else
            {
                prev-> next = temp->next;
                free(temp);
            }
```

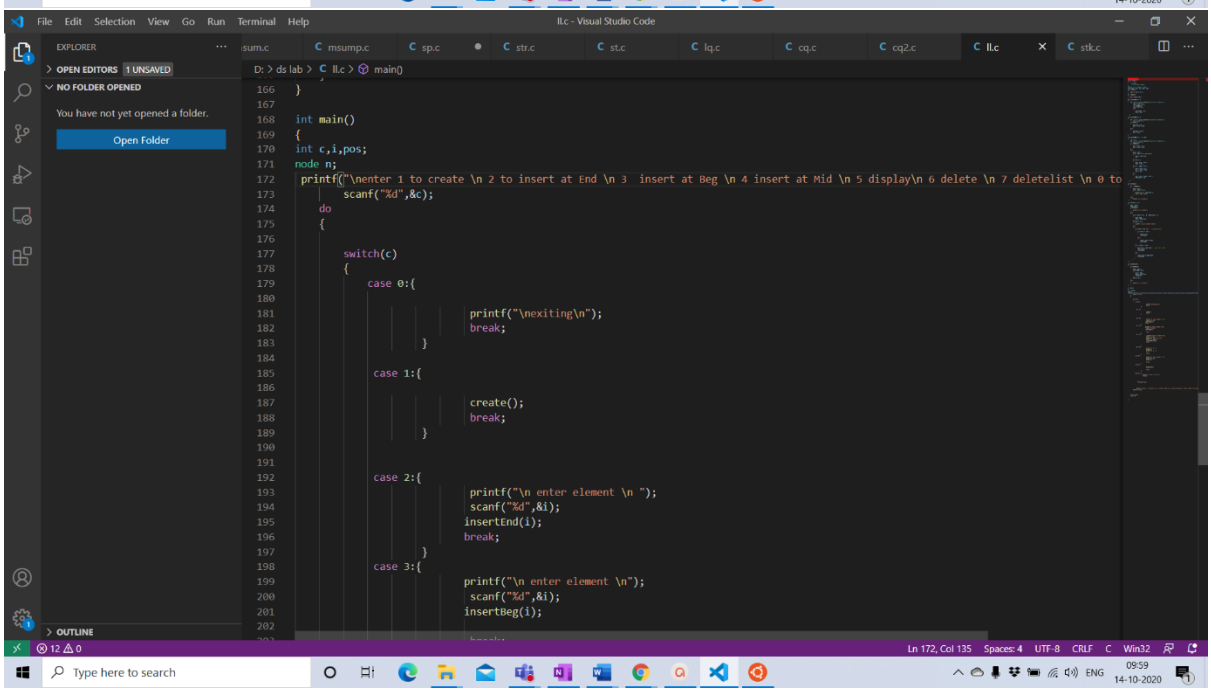
```
        }
    }
}

void deletelist()
{
    if(!isEmpty())
    {
        temp = start ;
        while(temp !=NULL)
        {
            prev = temp ;
            temp = temp->next;
            free(prev);
        }
        start = NULL;
    }
    else
    {
        printf("list is empty");
    }
}

int main()
{
int c,i,pos;
node n;
 printf("\nenter 1 to create \n 2 to insert at End \n 3  insert at Beg \n 4 in
sert at Mid \n 5 display\n 6 delete \n 7 deletelist \n 0 to exit \n");
        scanf("%d",&c);
    do
    {

        switch(c)
        {
            case 0:{

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

             case 1:{

                                create();
                                break;
                    }
```

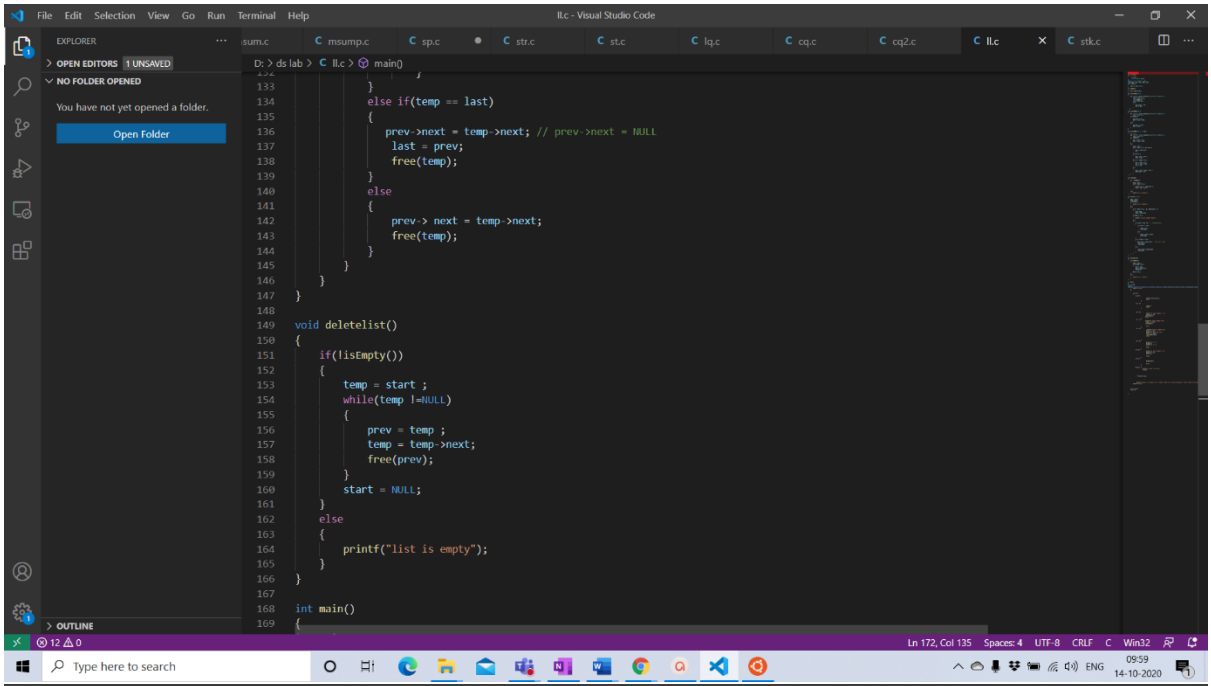
```
            case 2:{
                            printf("\n enter element \n ");
                            scanf("%d",&i);
                           insertEnd(i);
                           break;
                    }
            case 3:{
                           printf("\n enter element \n");
                            scanf("%d",&i);
                           insertBeg(i);

                            break;
                    }
            case 4:{
                             printf("\n enter element \n");
                            scanf("%d",&i);
                            printf("\n enter pos \n");
                            scanf("%d",&pos);
                             insertMid(i,pos);
                            break;


                    }
            case 5:{
                            printf("\n \n ");
                            display();
                            printf("\n \n ");

                            break;
                    }
           case 6:{
                            printf("\n enter element \n");
                            scanf("%d",&i);
                            delete(i);

                            break;
                    }
           case 7:{

                            deletelist();

                            break;
                    }

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

```
                }
                if(c==8) break;




              printf("\nenter 1 to create \n 2 to insert at End \n 3  insert a
t Beg \n 4 insert at Mid \n 5 display\n 6 delete \n 7 deletelist \n 0 to exit
\n");
        scanf("%d",&c);



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


}
```

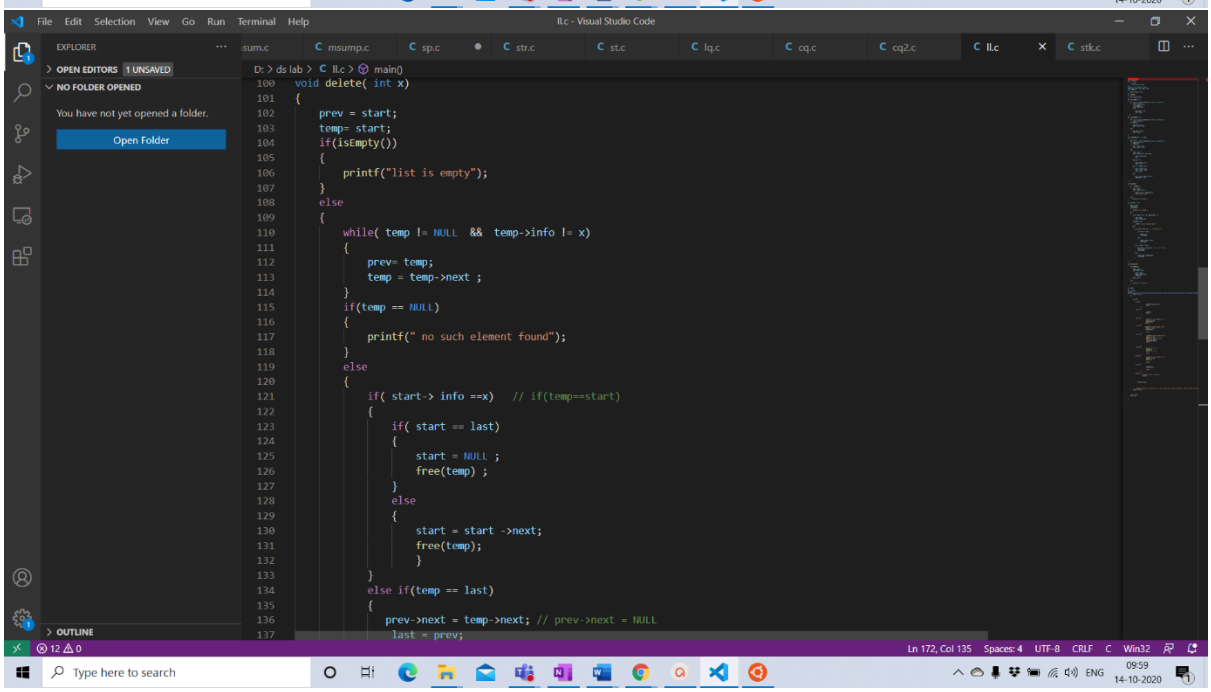
SCREENSHOT OF SOURCE CODE


File Edit Selection View Go Run Terminal Help
ll.c - Visual Studio Code
EXPLORER
OPEN EDITORS 1 UNSAVED
NO FOLDER OPENED
You have not yet opened a folder.
Open Folder
OUTLINE
sum.c msump.c sp.c str.c st.c lq.c cq.c cq2.c ll.c stk.c
D: > ds lab > ll.c > main()
case 3:{
printf("\n enter element \n");
scanf("%d",&i);
insertBeg(i);
break;
case 4:{
printf("\n enter element \n");
scanf("%d",&i);
printf("\n enter pos \n");
scanf("%d",&pos);
insertMid(i,pos);
break;
case 5:{
printf("\n \n ");
display();
printf("\n \n ");
break;
case 6:{
printf("\n enter element \n");
scanf("%d",&i);
delete(i);
break;
case 7:{
deletelist();
break;
Ln 172, Col 135 Spaces: 4 UTF-8 CRLF C Win32
Type here to search
ENG
09:59
14-10-2020

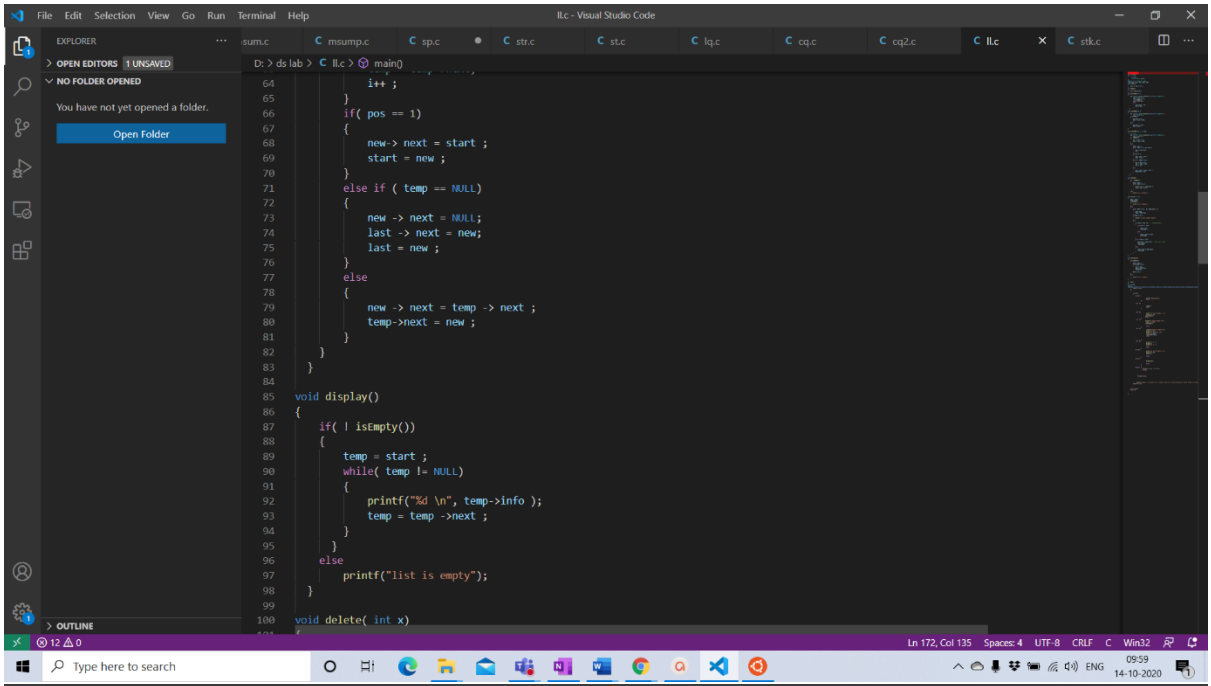

File Edit Selection View Go Run Terminal Help
ll.c - Visual Studio Code
EXPLORER
OPEN EDITORS 1 UNSAVED
NO FOLDER OPENED
You have not yet opened a folder.
Open Folder
OUTLINE
sum.c msump.c sp.c str.c st.c lq.c cq.c cq2.c ll.c stk.c
D: > ds lab > ll.c > main()
}
int main()
{
int c,i,pos;
node n;
printf("\nenter 1 to create \n 2 to insert at End \n 3 insert at Beg \n 4 insert at Mid \n 5 display\n 6 delete \n 7 deletelist \n 0 to
scanf("%d",&c);
do
{
switch(c)
{
case 0:{
printf("\nexiting\n");
break;
case 1:{
create();
break;
case 2:{
printf("\n enter element \n ");
scanf("%d",&i);
insertEnd(i);
break;
case 3:{
printf("\n enter element \n");
scanf("%d",&i);
insertBeg(i);
Ln 172, Col 135 Spaces: 4 UTF-8 CRLF C Win32
Type here to search
ENG
09:59
14-10-2020

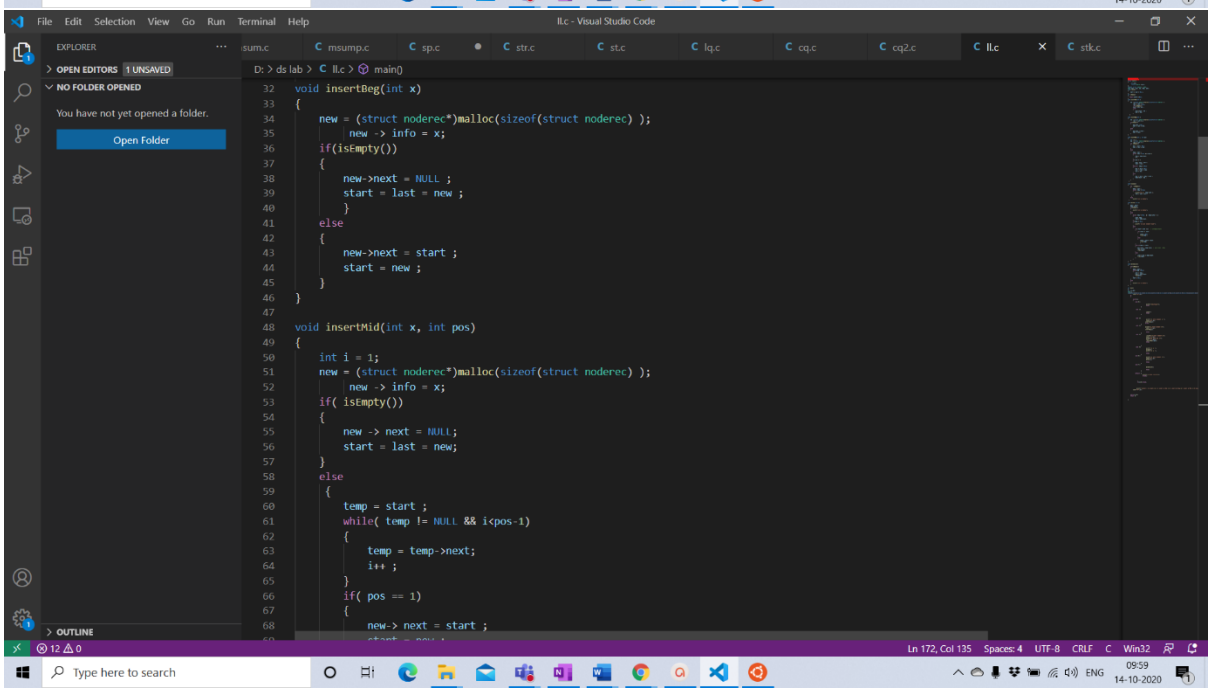


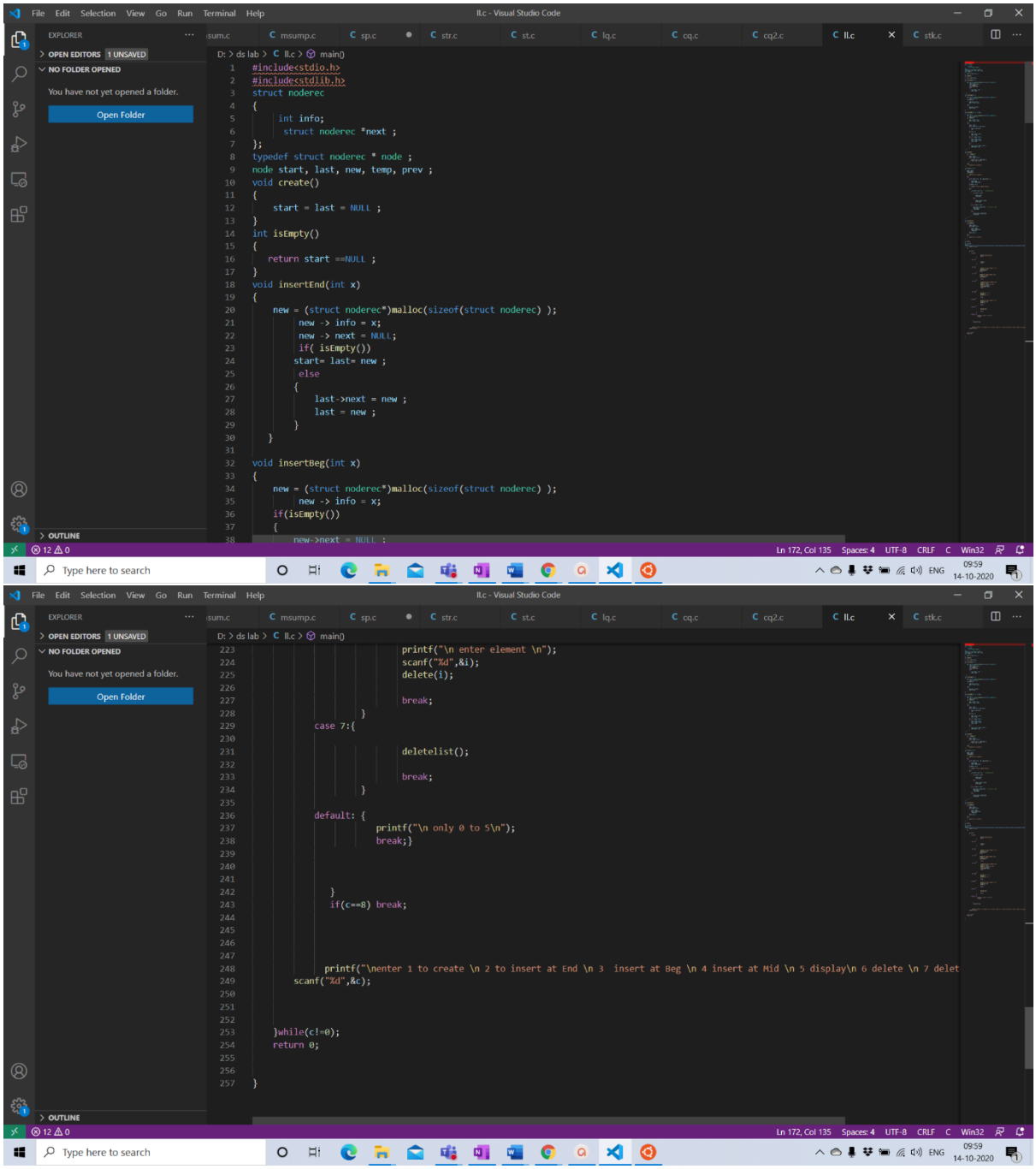
```
            }
            else if(temp == last)
            {
                prev->next = temp->next; // prev->next = NULL
                 last = prev;
                 free(temp);
            }
            else
            {
                prev-> next = temp->next;
                free(temp);
            }
        }
    }
}

void deletelist()
{
    if(!isEmpty())
    {
        temp = start ;
        while(temp !=NULL)
        {
            prev = temp ;
            temp = temp->next;
            free(prev);
        }
        start = NULL;
    }
    else
    {
        printf("list is empty");
    }
}

int main()
{
```



```
void delete( int x)
{
    prev = start;
    temp= start;
    if(isEmpty())
    {
        printf("list is empty");
    }
    else
    {
        while( temp != NULL  &&  temp->info != x)
        {
            prev= temp;
            temp = temp->next ;
        }
        if(temp == NULL)
        {
            printf(" no such element found");
        }
        else
        {
            if( start-> info ==x)   // if(temp==start)
            {
                if( start == last)
                {
                    start = NULL ;
                    free(temp) ;
                }
                else
                {
                    start = start ->next;
                    free(temp);
                    }
            }
            else if(temp == last)
            {
                prev->next = temp->next; // prev->next = NULL
                 last = prev;
```


File Edit Selection View Go Run Terminal Help
ll.c - Visual Studio Code
EXPLORER
OPEN EDITORS 1 UNSAVED
NO FOLDER OPENED
You have not yet opened a folder.
Open Folder
OUTLINE
sum.c
msump.c
sp.c
str.c
st.c
lq.c
cq.c
cq2.c
ll.c
stk.c
D: > ds lab > ll.c > main()
i++ ;
}
if(pos == 1)
{
new-> next = start ;
start = new ;
}
else if (temp == NULL)
{
new -> next = NULL;
last -> next = new;
last = new ;
}
else
{
new -> next = temp -> next ;
temp->next = new ;
}
}
}
void display()
{
if(! isEmpty())
{
temp = start ;
while(temp != NULL)
{
printf("%d \n", temp->info);
temp = temp ->next ;
}
}
else
printf("list is empty");
}
void delete(int x)
{
Ln 172, Col 135 Spaces: 4 UTF-8 CRLF C Win32
Type here to search
ENG
09:59
14-10-2020


File Edit Selection View Go Run Terminal Help
ll.c - Visual Studio Code
EXPLORER
OPEN EDITORS 1 UNSAVED
NO FOLDER OPENED
You have not yet opened a folder.
Open Folder
OUTLINE
sum.c
msump.c
sp.c
str.c
st.c
lq.c
cq.c
cq2.c
ll.c
stk.c
D: > ds lab > ll.c > main()
void insertBeg(int x)
{
new = (struct noderec*)malloc(sizeof(struct noderec));
new -> info = x;
if(isEmpty())
{
new->next = NULL ;
start = last = new ;
}
else
{
new->next = start ;
start = new ;
}
}
void insertMid(int x, int pos)
{
int i = 1;
new = (struct noderec*)malloc(sizeof(struct noderec));
new -> info = x;
if(isEmpty())
{
new -> next = NULL;
start = last = new;
}
else
{
temp = start ;
while(temp != NULL && i<pos-1)
{
temp = temp->next;
i++ ;
}
if(pos == 1)
{
new-> next = start ;
Ln 172, Col 135 Spaces: 4 UTF-8 CRLF C Win32
Type here to search
ENG
09:59
14-10-2020

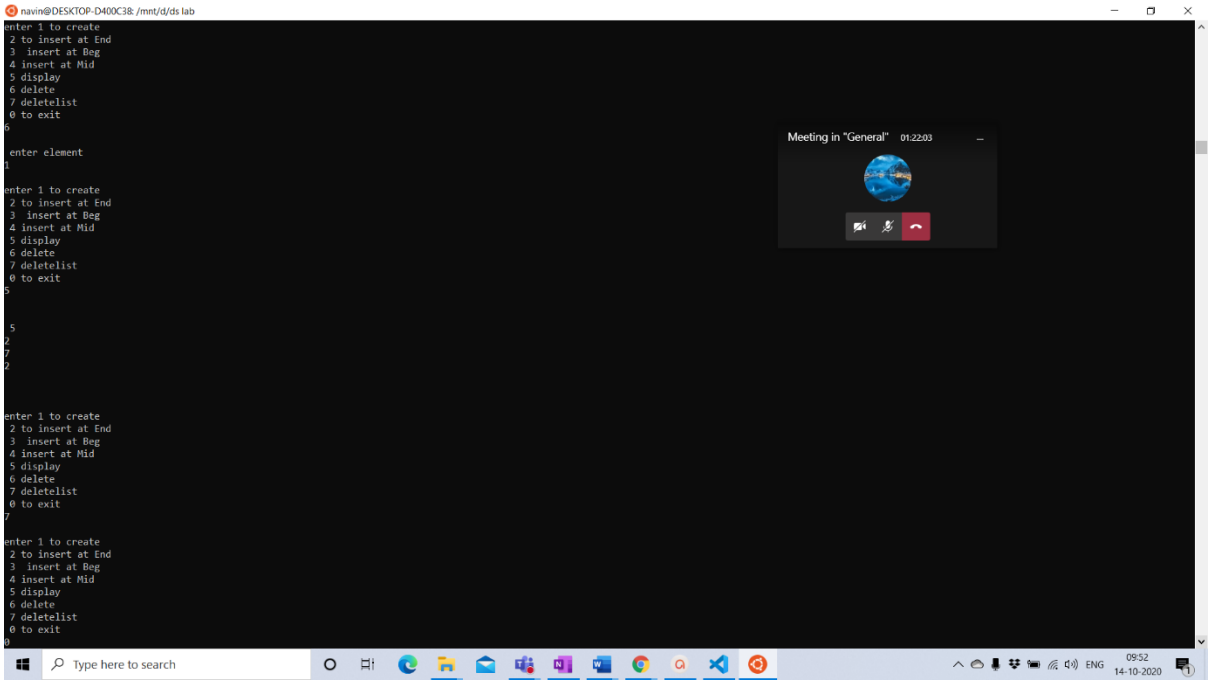


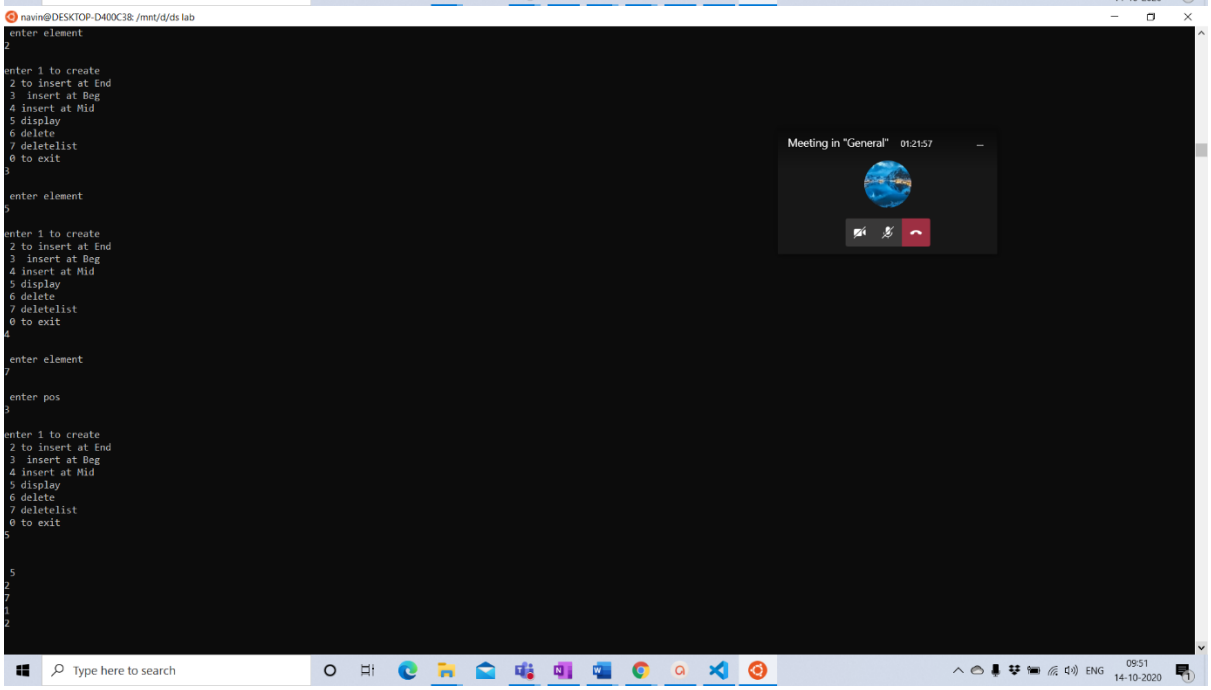
```c
#include<stdio.h>
#include<stdlib.h>
struct noderec
{
    int info;
    struct noderec *next ;
};
typedef struct noderec * node ;
node start, last, new, temp, prev ;
void create()
{
    start = last = NULL ;
}
int isEmpty()
{
   return start ==NULL ;
}
void insertEnd(int x)
{
    new = (struct noderec*)malloc(sizeof(struct noderec) );
        new -> info = x;
        new -> next = NULL;
        if( isEmpty())
       start= last= new ;
        else
        {
            last->next = new ;
            last = new ;
        }
   }

void insertBeg(int x)
{
    new = (struct noderec*)malloc(sizeof(struct noderec) );
        new -> info = x;
    if(isEmpty())
    {
        new->next = NULL ;
```

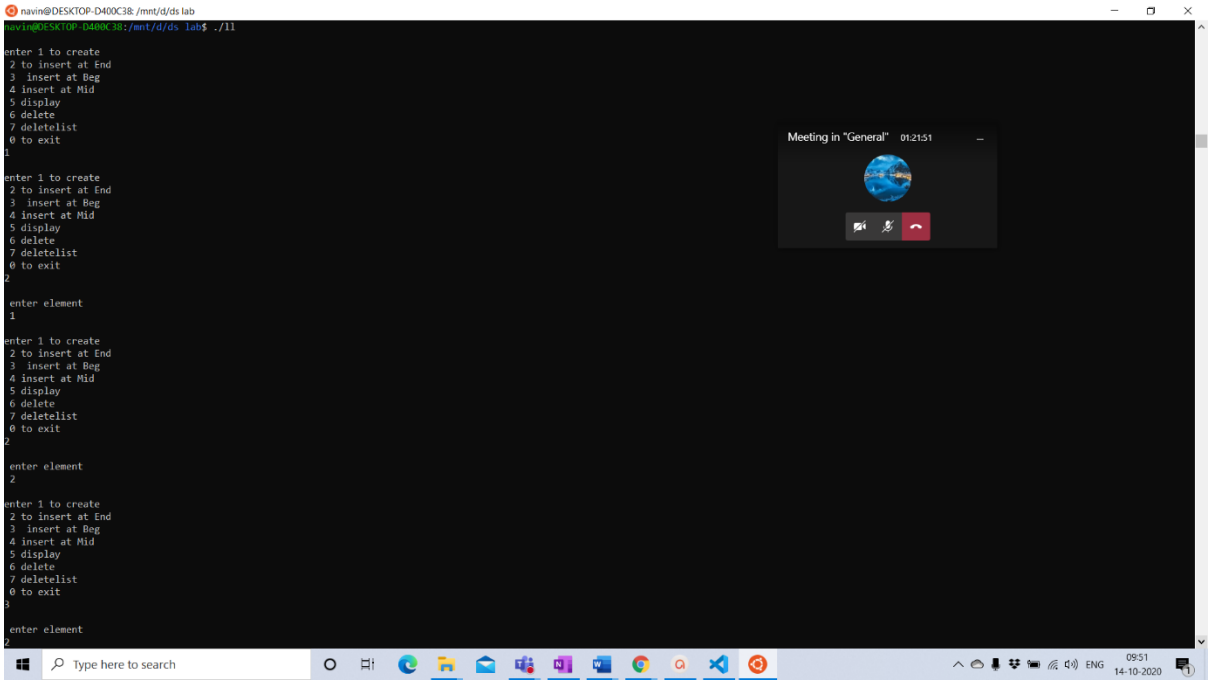


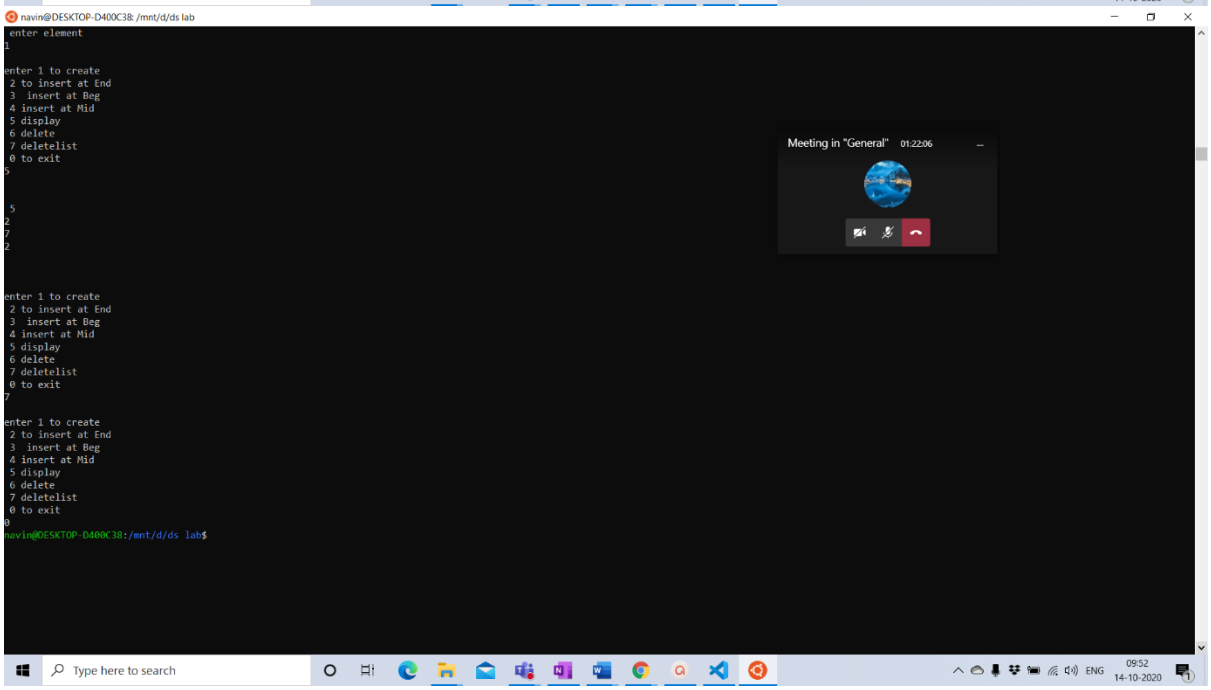
```c
                    printf("\n enter element \n");
                    scanf("%d",&i);
                    delete(i);

                    break;
                }
            case 7:{

                    deletelist();

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


           }
           if(c==8) break;



        printf("\nenter 1 to create \n 2 to insert at End \n 3  insert at Beg \n 4 insert at Mid \n 5 display\n 6 delete \n 7 delet
    scanf("%d",&c);


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


}
```

SCREENSHOT OF OUTPUT


navin@DESKTOP-D400C38: /mnt/d/ds lab
enter 1 to create
2 to insert at End
3 insert at Beg
4 insert at Mid
5 display
6 delete
7 deletelist
0 to exit
6
enter element
1
enter 1 to create
2 to insert at End
3 insert at Beg
4 insert at Mid
5 display
6 delete
7 deletelist
0 to exit
5
5
2
7
2
enter 1 to create
2 to insert at End
3 insert at Beg
4 insert at Mid
5 display
6 delete
7 deletelist
0 to exit
7
enter 1 to create
2 to insert at End
3 insert at Beg
4 insert at Mid
5 display
6 delete
7 deletelist
0 to exit
0
Meeting in "General" 01:22:03
Type here to search
ENG
09:52
14-10-2020


navin@DESKTOP-D400C38: /mnt/d/ds lab
enter element
2
enter 1 to create
2 to insert at End
3 insert at Beg
4 insert at Mid
5 display
6 delete
7 deletelist
0 to exit
3
enter element
5
enter 1 to create
2 to insert at End
3 insert at Beg
4 insert at Mid
5 display
6 delete
7 deletelist
0 to exit
4
enter element
7
enter pos
3
enter 1 to create
2 to insert at End
3 insert at Beg
4 insert at Mid
5 display
6 delete
7 deletelist
0 to exit
5
5
2
7
1
2
Meeting in "General" 01:21:57
Type here to search
ENG
09:51
14-10-2020


navin@DESKTOP-D400C38: /mnt/d/ds lab
navin@DESKTOP-D400C38:/mnt/d/ds lab$ ./ll
enter 1 to create
2 to insert at End
3 insert at Beg
4 insert at Mid
5 display
6 delete
7 deletelist
0 to exit
1
enter 1 to create
2 to insert at End
3 insert at Beg
4 insert at Mid
5 display
6 delete
7 deletelist
0 to exit
2
enter element
1
enter 1 to create
2 to insert at End
3 insert at Beg
4 insert at Mid
5 display
6 delete
7 deletelist
0 to exit
2
enter element
2
enter 1 to create
2 to insert at End
3 insert at Beg
4 insert at Mid
5 display
6 delete
7 deletelist
0 to exit
3
enter element
2
Meeting in "General" 01:21:51
Type here to search
ENG
09:51
14-10-2020


navin@DESKTOP-D400C38: /mnt/d/ds lab
enter element
1
enter 1 to create
2 to insert at End
3 insert at Beg
4 insert at Mid
5 display
6 delete
7 deletelist
0 to exit
5
5
2
7
2
enter 1 to create
2 to insert at End
3 insert at Beg
4 insert at Mid
5 display
6 delete
7 deletelist
0 to exit
7
enter 1 to create
2 to insert at End
3 insert at Beg
4 insert at Mid
5 display
6 delete
7 deletelist
0 to exit
0
navin@DESKTOP-D400C38:/mnt/d/ds lab$
Meeting in "General" 01:22:06
Type here to search
ENG
09:52
14-10-2020