

Practice Lab 10 - Circularly and Doubly Linked list - 21-10-2020

Swaminathan Navinashok

2019115126

Double linked list

Source code:

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

struct noderec
{
    int info;
    struct noderec *lptr;
    struct noderec *rptr;
};
typedef struct noderec * node ;
typedef node header;
node new, temp;
header h ;
int i = 1 ;
void create()
{
    h = (struct noderec*) malloc(sizeof(struct noderec));
    h->lptr = NULL;
    h->rptr = NULL;
}
int isEmpty()
{
    return (h->rptr ==NULL);
}
void insertBeg( header h, int x)
{
    new = (struct noderec*) malloc(sizeof(struct noderec));
    new->info = x;
    if(isEmpty())
    {
        new->rptr = NULL;
```

```

        new->lptr = h;
        h->rptr = new;
    }
    else
    {
        new->rptr = h->rptr;
        new->lptr = h;
        (h->rptr)->lptr = new;
        h->rptr = new;
    }
}

void insertEnd(header h, int x)
{
    new = (struct noderec*) malloc(sizeof(struct noderec));
    new->info = x;
    if(isEmpty())
    {
        new->rptr = NULL;
        new->lptr = h;
        h->rptr = new;
    }
    else
    {
        temp = h->rptr ;
        while(temp->rptr !=NULL)
            temp = temp->rptr ;
        new->rptr = NULL;
        new->lptr = temp;
        temp->rptr = new;
    }
}

void insertMid(header h, int x, int pos)
{
    new = (struct noderec*) malloc(sizeof(struct noderec));
    new->info = x;
    if(isEmpty())
    {
        new->rptr = NULL;
        new->lptr = h;
        h->rptr = new;
    }
    else
    {
        temp = h;
        while( temp->rptr !=NULL && i< pos)
        {
            temp = temp->rptr ;
            i++ ;
        }
    }
}

```

```

    }
    if(temp->rptr ==NULL)
    {
        //insertEnd(h, x);
        new->rptr = NULL;
        new->lptr = temp;
        temp->rptr = new;
    }
    else
    {
        new->lptr = temp;
        new->rptr = temp->rptr;
        (temp->rptr)->lptr = new;
        temp->rptr = new ;
    }
}
}
void delete( header h, int x)
{
    if(!isEmpty(h))
    {
        temp = h->rptr ;
        while(temp != NULL &&temp->info!=x)
            temp = temp->rptr;
        if(temp ==NULL)
            printf("element not found");
        else
        {
            (temp->rptr)->lptr = temp->lptr ;
            (temp->lptr)->rptr = temp->rptr ;
            free(temp);
        }
    }
    else
        printf("list is empty");
}
int search(header h, int x)
{
    temp = h->rptr;
    int pos=0;
    if(!isEmpty(h))
    {
        while(temp!=h && temp->info!=x)
        {
            temp = temp->rptr ;
            pos++;
        }
        if(temp ==h)
            return 0 ;
    }
}

```

```

        else
            return pos;
    }
    else
        return -1 ;
}
void display()
{

    printf("\n\n");
    if( !isEmpty(h) )
    {
        temp = h->rptr ;
        while( temp != NULL )
        {
            printf(" \t %d ", temp->info );
            temp = temp ->rptr ;
        }
        printf("\n\n");
    }
    else
        printf("list is empty");
}
void delist() {
    if (isEmpty(h)) {
        free(h);
    } else {
        node temp = h->rptr;
        node prev = h;
        while (temp != NULL) {
            prev = temp;
            temp = temp->rptr;
            free(prev);
        }
        free(h);
    }
}

int main()
{
    int c;

    create();
    printf("\nenter 0 to exit\n 1 to insert at beginning \n 2  to insert at end
\n 3 to display \n 4 to delete element\n 5 to delete list \n 6 search \n 7 ins
ert mid\n");
    scanf("%d",&c);
    do

```

```

{
    switch(c)
    {
        case 0:{

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

        }

        case 1:{

            int me;printf("\nenter element to insert at begin
ning\n");

            scanf("%d",&me);
            insertBeg(h,me);

            break;

        }
        case 2:{

            int me;printf("\nenter element to insert at end \n
");

            scanf("%d",&me);
            insertEnd(h,me);

            break;

        }
        case 3:{

            display();

            break;

        }
        case 4:{

            int n;
            printf("\n enter element to delete \n");
            scanf("%d",&n);
            delete(h,n);
            break;

        }

        case 5:{

            delist();

            break;

        }
    }
}

```

```

        case 6:{
            int q,p=0;
            printf("\n enter element to search \n");
            scanf("%d",&q);
            p=search(h,q);p++;

            printf("\n position of element is(0 if not found)
: %d \n ",p);

            break;

        }
        case 7:{
            int f,pos;
            printf("\n enter element to enter \n");
            scanf("%d",&f);
            printf("\n enter position to enter element \n");
            scanf("%d",&pos);
            insertMid(h,f,pos);
            break;

        }

        default: {
            printf("\n only 0 to 7\n");
            break;
        }

    }

    if(c==0) break;
    printf("\nenter 0 to exit\n 1 to insert at beginning \n 2 to insert at
end \n 3 to display \n 4 to delete element\n 5 to deletelist \n 6 search \n
7 insert at mid\n");
    scanf("%d",&c);

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

```

```
}
```

Source code screenshot:

```
File Edit Selection View Go Run Terminal Help
dllc - c programs - Visual Studio Code

D:\ds lab > c dllc > main()
217 case 5:{
218     delist();
219     break;
220 }
221
222 case 6:{
223     int q,p=0;
224     printf("\n enter element to search \n");
225     scanf("%d",&q);
226     p=search(h,q);p++;
227
228     printf("\n position of element is(0 if not found) : %d \n ",p);
229     break;
230 }
231
232
233
234
235 case 7:{
236     int f,pos;
237     printf("\n enter element to enter \n");
238     scanf("%d",&f);
239     printf("\n enter position to enter element \n");
240     scanf("%d",&pos);
241     insertMid(h,f,pos);
242     break;
243 }
244
245
246
247
248
249 default: {
250     printf("\n only 0 to 7\n");
251     break;
252 }
253
254
```

```
File Edit Selection View Go Run Terminal Help
dllc - c programs - Visual Studio Code

D:\ds lab > c dllc > main()
194 case 2:{
195     int me;printf("\n enter element to insert at end \n");
196     scanf("%d",&me);
197     insertend(h,me);
198
199     break;
200 }
201
202 case 3:{
203     display();
204     break;
205 }
206
207 case 4:{
208     int n;
209     printf("\n enter element to delete \n");
210     scanf("%d",&n);
211     delete(h,n);
212     break;
213 }
214
215
216
217 case 5:{
218     delist();
219     break;
220 }
221
222 case 6:{
223     int q,p=0;
224     printf("\n enter element to search \n");
225     scanf("%d",&q);
226     p=search(h,q);p++;
227
228     printf("\n position of element is(0 if not found) : %d \n ",p);
229     break;
230 }
```



```
File Edit Selection View Go Run Terminal Help
dllc - c programs - Visual Studio Code

C asses1.c C asses1-2.c C dllc x C llc.c C cq2.c
D:\ds lab > C dllc > main()
167
168 int main()
169 {
170     int c;
171
172     create();
173     printf("\nEnter 0 to exit\n 1 to insert at beginning\n 2 to insert at end\n 3 to display\n 4 to delete element\n 5 to delete list\n 6 search\n 7 insert mid\n");
174     scanf("%d",&c);
175
176     do
177     {
178         switch(c)
179         {
180             case 0:
181                 printf("\nexiting\n");
182                 break;
183             case 1:
184                 int me;printf("\nEnter element to insert at beginning\n");
185                 scanf("%d",&me);
186                 insertBeg(h,me);
187                 break;
188             case 2:
189                 int me;printf("\nEnter element to insert at end\n");
190                 scanf("%d",&me);
191                 insertend(h,me);
192                 break;
193             case 3:
194                 display();
195         }
196     }
197 }
```

```
File Edit Selection View Go Run Terminal Help
dllc - c programs - Visual Studio Code

C asses1.c C asses1-2.c C dllc x C llc.c C cq2.c
D:\ds lab > C dllc > main()
136 void display()
137 {
138     printf("\n\n");
139     if( !isEmpty(h) )
140     {
141         temp = h->rptr ;
142         while( temp != NULL )
143         {
144             printf("\t %d ", temp->info );
145             temp = temp->rptr ;
146         }
147         printf("\n\n");
148     }
149     else
150         printf("list is empty");
151 }
152
153 void delist() {
154     if (isEmpty(h)) {
155         free(h);
156     } else {
157         node temp = h->rptr;
158         node prev = h;
159         while (temp != NULL) {
160             prev = temp;
161             temp = temp->rptr;
162             free(temp);
163         }
164         free(h);
165     }
166 }
167
168 int main()
169 {
170     int c;
171
172     create();
173 }
```

```
File Edit Selection View Go Run Terminal Help
dllc - c programs - Visual Studio Code

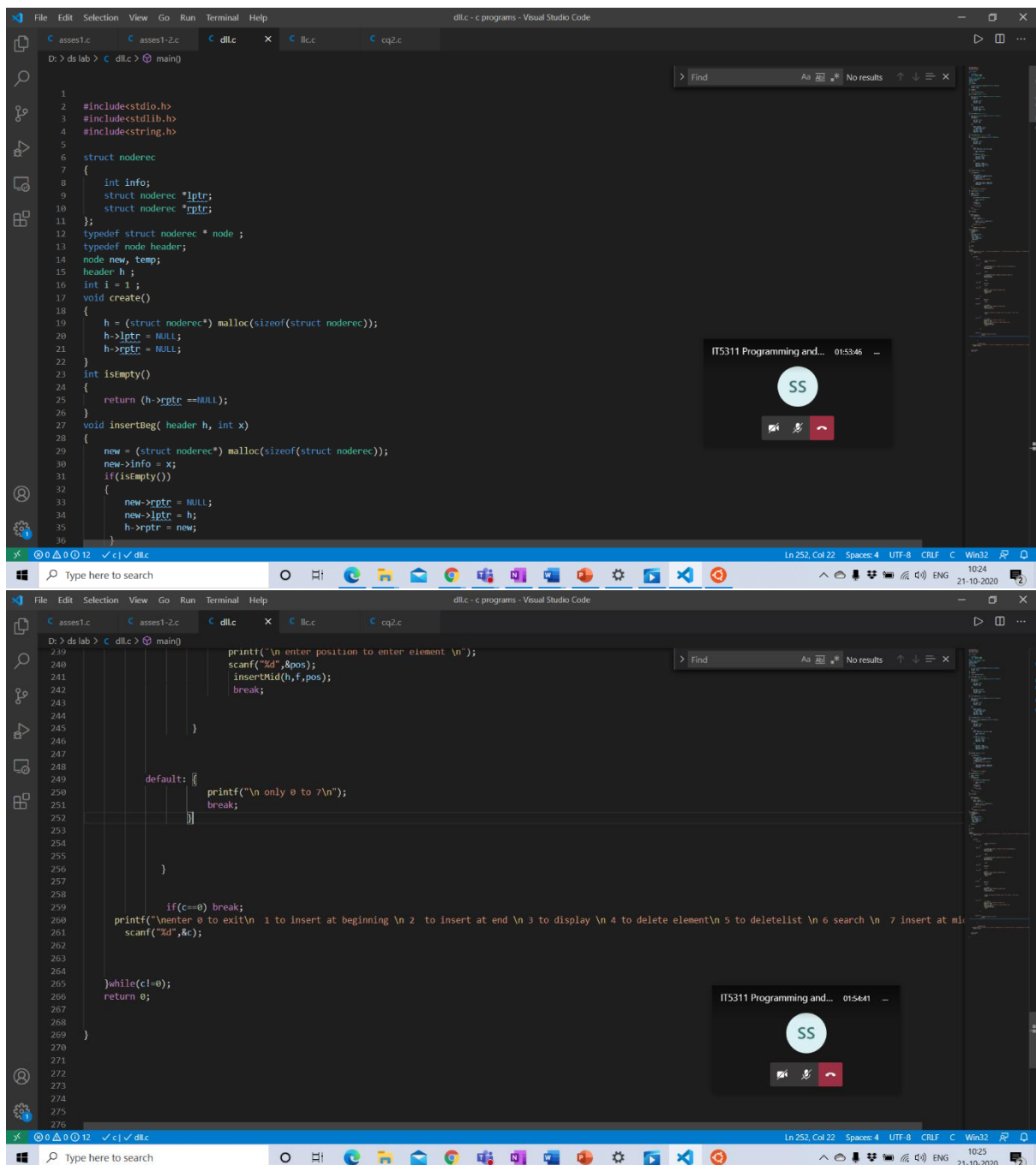
C asses1.c C asses1-2.c dllc x llc.c cq2.c
D:\ds lab> cd dllc > main()
117 int search(header h, int x)
118 {
119     temp = h->rptr;
120     int pos=0;
121     if(listempty(h))
122     {
123         while(temp!=h && temp->info!=x)
124         {
125             temp = temp->rptr ;
126             pos++;
127         }
128         if(temp ==h)
129             return 0 ;
130         else
131             return pos;
132     }
133     else
134         return -1 ;
135 }
136 void display()
137 {
138     printf("\n\n");
139     if( listempty(h) )
140     {
141         temp = h->rptr ;
142         while( temp != NULL )
143         {
144             printf(" \t %d ", temp->info );
145             temp = temp->rptr ;
146         }
147         printf("\n\n");
148     }
149     else
150         printf("list is empty");
151 }
152 void delist() {
153 }
```

```
File Edit Selection View Go Run Terminal Help
dllc - c programs - Visual Studio Code

C asses1.c C asses1-2.c dllc x llc.c cq2.c
D:\ds lab> cd dllc > main()
98 void delete( header h, int x)
99 {
100     if(listempty(h))
101     {
102         temp = h->rptr ;
103         while(temp != NULL && temp->info!=x)
104             temp = temp->rptr;
105         if(temp ==NULL)
106             printf("element not found");
107         else
108         {
109             (temp->rptr)->lptr = temp->lptr ;
110             (temp->lptr)->rptr = temp->rptr ;
111             free(temp);
112         }
113     }
114     else
115         printf("list is empty");
116 }
117 int search(header h, int x)
118 {
119     temp = h->rptr;
120     int pos=0;
121     if(listempty(h))
122     {
123         while(temp!=h && temp->info!=x)
124         {
125             temp = temp->rptr ;
126             pos++;
127         }
128         if(temp ==h)
129             return 0 ;
130         else
131             return pos;
132     }
133     else
134         return -1 ;
135 }
```

```
File Edit Selection View Go Run Terminal Help dflc - c programs - Visual Studio Code
C asses1.c C asses1-2.c C dflc x C llc.c C cq2.c
D:\ds lab > C dflc > main()
65 void insertMid(header h, int x, int pos)
66 {
67     new = (struct noderec*) malloc(sizeof(struct noderec));
68     new->info = x;
69     if(isEmpty())
70     {
71         new->rptr = NULL;
72         new->lptr = h;
73         h->rptr = new;
74     }
75     else
76     {
77         temp = h;
78         while( temp->rptr !=NULL && i< pos)
79         {
80             temp = temp->rptr ;
81             i++ ;
82         }
83         if(temp->rptr ==NULL)
84         {
85             //insertEnd(h, x);
86             new->rptr = NULL;
87             new->lptr = temp;
88             temp->rptr = new;
89         }
90         else
91         {
92             new->lptr = temp;
93             new->rptr = temp->rptr;
94             (temp->rptr)->lptr = new;
95             temp->rptr = new ;
96         }
97     }
98 void delete( header h, int x)
99 {
100     if(!isEmpty(h))
101     {
102         temp = h->rptr ;
103     }
104 }
```

```
File Edit Selection View Go Run Terminal Help dflc - c programs - Visual Studio Code
C asses1.c C asses1-2.c C dflc x C llc.c C cq2.c
D:\ds lab > C dflc > main()
27 void insertBeg( header h, int x)
28 {
29     new = (struct noderec*) malloc(sizeof(struct noderec));
30     new->info = x;
31     if(isEmpty())
32     {
33         new->rptr = NULL;
34         new->lptr = h;
35         h->rptr = new;
36     }
37     else
38     {
39         new->rptr = h->rptr;
40         new->lptr = h;
41         (h->rptr)->lptr = new;
42         h->rptr = new;
43     }
44 }
45 void insertEnd(header h, int x)
46 {
47     new = (struct noderec*) malloc(sizeof(struct noderec));
48     new->info = x;
49     if(isEmpty())
50     {
51         new->rptr = NULL;
52         new->lptr = h;
53         h->rptr = new;
54     }
55     else
56     {
57         temp = h->rptr ;
58         while(temp->rptr !=NULL)
59             temp = temp->rptr ;
60         new->rptr = NULL;
61         new->lptr = temp;
62         temp->rptr = new;
63     }
64 }
```



Output screenshot

```
navin@DESKTOP-D40C38: /mnt/d/ds lab
enter position to enter element
3
enter 0 to exit
1 to insert at beginning
2 to insert at end
3 to display
4 to delete element
5 to deletelist
6 search
7 insert at mid
3
2 1 5 3 4
enter 0 to exit
1 to insert at beginning
2 to insert at end
3 to display
4 to delete element
5 to deletelist
6 search
7 insert at mid
0
enter element to search
5
position of element is(0 if not found) : 3
enter 0 to exit
1 to insert at beginning
2 to insert at end
3 to display
4 to delete element
5 to deletelist
6 search
7 insert at mid
4
enter element to delete
3
enter 0 to exit
1 to insert at beginning
2 to insert at end
3 to display
4 to delete element
```

```
navin@DESKTOP-D40C38: /mnt/d/ds lab
enter element to insert at end
3
enter 0 to exit
1 to insert at beginning
2 to insert at end
3 to display
4 to delete element
5 to deletelist
6 search
7 insert at mid
2
enter element to insert at end
4
enter 0 to exit
1 to insert at beginning
2 to insert at end
3 to display
4 to delete element
5 to deletelist
6 search
7 insert at mid
3
2 1 3 4
enter 0 to exit
1 to insert at beginning
2 to insert at end
3 to display
4 to delete element
5 to deletelist
6 search
7 insert at mid
7
enter element to enter
0
enter position to enter element
3
enter 0 to exit
1 to insert at beginning
2 to insert at end
3 to display
```

```
navin@DESKTOP-D40C38:/mnt/d/ds lab$
navin@DESKTOP-D40C38:/mnt/d/ds lab$ gcc dll.c -o dll
navin@DESKTOP-D40C38:/mnt/d/ds lab$ ./dll

enter 0 to exit
1 to insert at beginning
2 to insert at end
3 to display
4 to delete element
5 to deletelist
6 search
7 insert mid
1
enter element to insert at beginning
1
enter 0 to exit
1 to insert at beginning
2 to insert at end
3 to display
4 to delete element
5 to deletelist
6 search
7 insert at mid
2
enter element to insert at end
3
enter 0 to exit
1 to insert at beginning
2 to insert at end
3 to display
4 to delete element
5 to deletelist
6 search
7 insert at mid
3
2 1 5 4
enter 0 to exit
1 to insert at beginning
2 to insert at end
3 to display
4 to delete element
5 to deletelist
6 search
7 insert at mid
5
enter 0 to exit
1 to insert at beginning
2 to insert at end
3 to display
4 to delete element
5 to deletelist
6 search
7 insert at mid
0
navin@DESKTOP-D40C38:/mnt/d/ds lab$
```

Circular linked list

Source code

```

#include<stdio.h>
#include<stdlib.h>
#include<string.h>
struct noderec
{
    int info;
    struct noderec *next ;
};
typedef struct noderec * node ;
node new, temp, prev ;
typedef node header;
header h;
void create()
{
    h = (struct noderec*) malloc(sizeof(struct noderec));
    h->next = h ;
}

int isEmpty(header h)
{
    return (h->next == h) ;
}
void insertBeg(header h, int x)
{
    new = (struct noderec*) malloc(sizeof(struct noderec));
    new->info = x;
    new->next = h->next ;
    h->next = new ;
}

void insertEnd(header h, int x)
{
    new = (struct noderec*) malloc(sizeof(struct noderec));
    new->info = x;
    temp = h->next ;
    while(temp->next != h)
    {
        temp = temp->next;
    }
    new->next = h;
    temp->next = new;
}

void display(header h)
{
    printf("\n\n");
    if( !isEmpty(h) )

```

```

{
    temp = h->next ;
    while( temp != h )
    {
        printf("\t %d ", temp->info );
        temp = temp ->next ;
    }
    printf("\n\n");
}
else
    printf("list is empty");
}

int search(header h, int x)
{
    temp = h->next;
int pos=0;
    if(!isEmpty(h))
    {
        while(temp!=h && temp->info!=x)
        {
            temp = temp->next;
            pos++;
        }
        if(temp ==h)
            return 0 ;
        else
            return pos;
    }
    else
        return -1 ;
}

void delete( header h, int x)
{
    if(!isEmpty(h))
    {
        temp = h->next;
        while(temp != h &&temp->next->info!=x)
            temp = temp->next;
        if(temp ==h)
            printf("element not found");
        else
        {
            node temp1=temp->next;
            temp->next = temp->next->next ;

            free(temp1);
        }
    }
}

```



```

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

void delist() {
    if (isEmpty(h)) {
        free(h);
    } else {
        node temp = h->next;
        node prev = h;
        while (temp != h) {
            prev = temp;
            temp = temp->next;
            free(prev);
        }
        free(h);
    }
}

int main()
{
    int c;

    create();
    printf("\nenter 0 to exit\n 1 to insert at beginning \n 2 to insert at end\n 3 to display \n 4 to delete element\n 5 to delete list \n 6 search\n");
    scanf("%d",&c);
    do
    {
        switch(c)
        {
            case 0:{
                printf("\nexiting\n");
                break;
            }

            case 1:{
                int me;printf("\nenter element to insert at beginning\n");
                scanf("%d",&me);
                insertBeg(h,me);

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

```

```

    }
    case 2:{
        int me;printf("\nenter element to insert at end \n");
        scanf("%d",&me);
        insertEnd(h,me);

        break;
    }
    case 3:{
        display(h);

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

    case 5:{
        delist();

        break;
    }
    case 6:{
        int n,p=0;
        printf("\n enter element to search \n");
        scanf("%d",&n);
        p=search(h,n);p++;

        printf("\n position of element is(0 if not found)
: %d \n ",p);

        break;
    }

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

```

```
    }

}

    if(c==0) break;

    printf("\nenter 0 to exit\n 1 to insert at beginning \n 2 to i
insert at end \n 3 to display \n 4 to delete element\n 5 to delete list\n 6 se
arch\n");
    scanf("%d",&c);

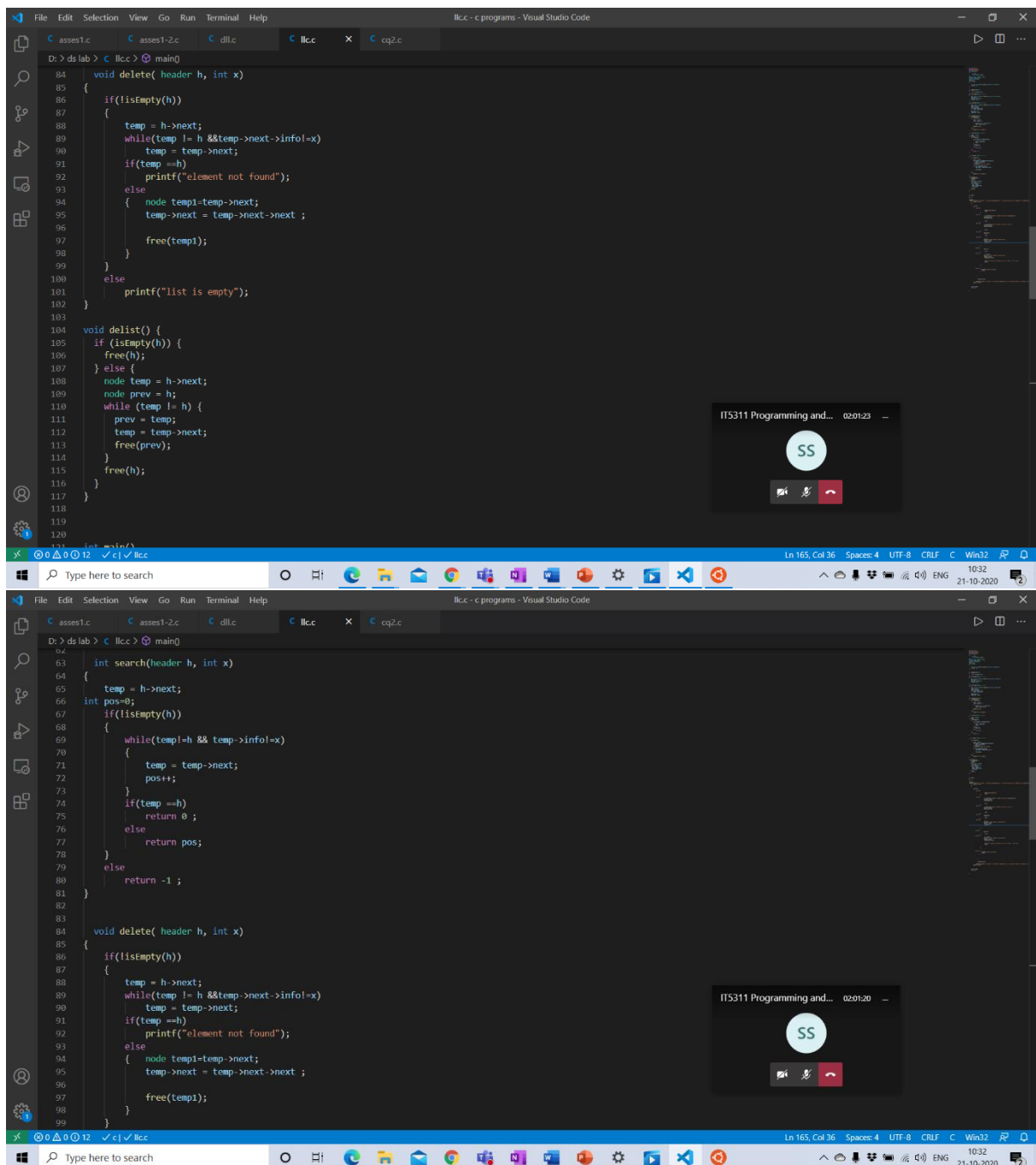
}while(c!=0);
return 0;

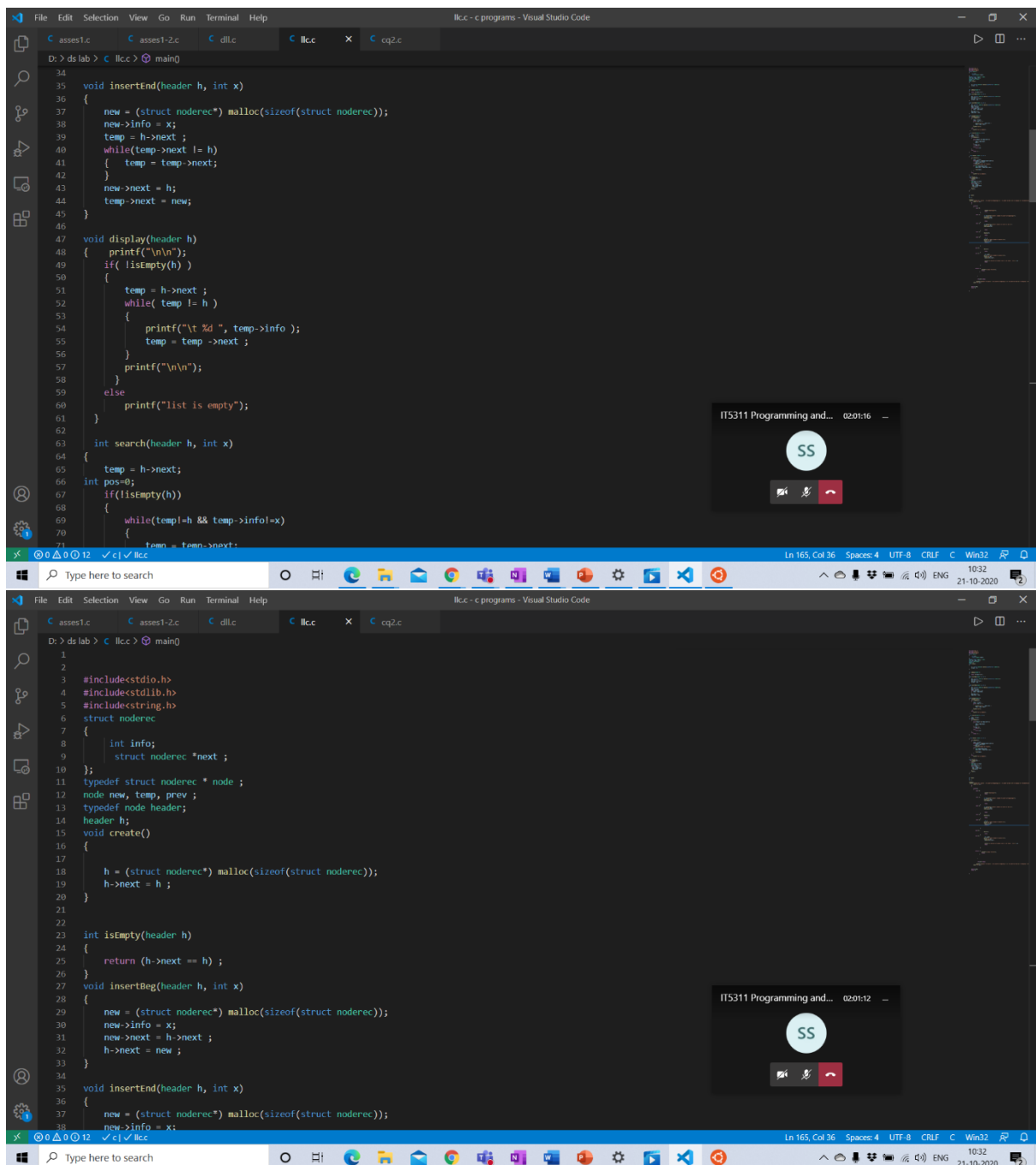
}
```

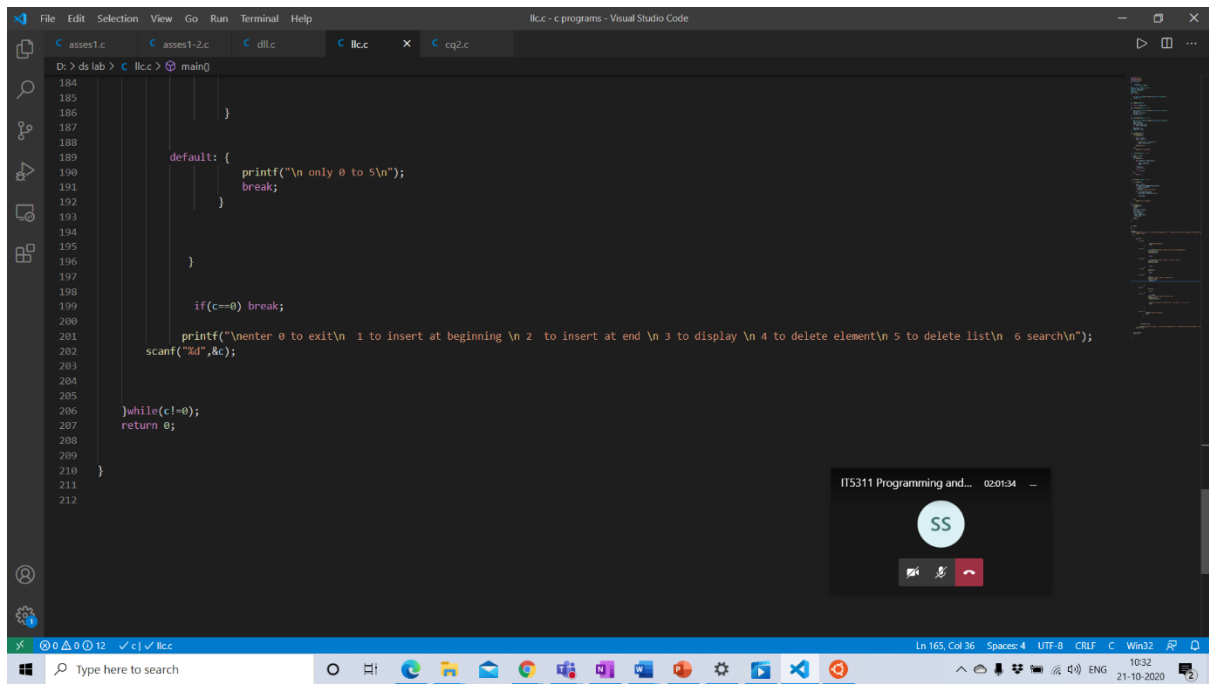
Screenshot of source code

```
155     case 3:{
156         display(h);
157         break;
158     }
159     case 4:{
160         int n;
161         printf("\n enter element to delete \n");
162         scanf("%d",&n);
163         delete(h,n);
164         break;
165     }
166
167
168     }
169
170     case 5:{
171         delist();
172         break;
173     }
174
175     case 6:{
176         int n,p=0;
177         printf("\n enter element to search \n");
178         scanf("%d",&n);
179         p=search(h,n);p++;
180
181         printf("\n position of element is(%d if not found) : %d \n ",p);
182         break;
183     }
184
185 }
186
187
188
189 default: {
190     printf("\n only 0 to 5\n");
191     break;
192 }
```

```
120 int main()
121 {
122     int c;
123
124     create();
125     printf("\n enter 0 to exit\n 1 to insert at beginning\n 2 to insert at end\n 3 to display\n 4 to delete element\n 5 to delete list\n 6 search\n");
126     scanf("%d",&c);
127     do
128     {
129         switch(c)
130         {
131             case 0:{
132                 printf("\nexiting\n");
133                 break;
134             }
135             case 1:{
136                 int me;printf("\n enter element to insert at beginning\n");
137                 scanf("%d",&me);
138                 insertBeg(h,me);
139                 break;
140             }
141             case 2:{
142                 int me;printf("\n enter element to insert at end \n");
143                 scanf("%d",&me);
144                 insertend(h,me);
145                 break;
146             }
147             case 3:{
148                 display(h);
149                 break;
150             }
151             case 4:{
152                 break;
153             }
154             case 5:{
155                 break;
156             }
157             case 6:{
158                 break;
159             }
```







Screenshot of output

```
navin@DESKTOP-D40C38: /mnt/d/ds lab
enter element to insert at end
6
enter 0 to exit
1 to insert at beginning
2 to insert at end
3 to display
4 to delete element
5 to delete list
6 search
3
3 2 5 6
enter 0 to exit
1 to insert at beginning
2 to insert at end
3 to display
4 to delete element
5 to delete list
6 search
6
enter element to search
2
position of element is(0 if not found) : 2
enter 0 to exit
1 to insert at beginning
2 to insert at end
3 to display
4 to delete element
5 to delete list
6 search
6
enter element to search
5
position of element is(0 if not found) : 3
enter 0 to exit
1 to insert at beginning
2 to insert at end
3 to display
4 to delete element
5 to delete list
6 search
6
```

```
navin@DESKTOP-D40C38: /mnt/d/ds lab$ ./c11
navin@DESKTOP-D40C38: /mnt/d/ds lab$ ./c11
enter 0 to exit
1 to insert at beginning
2 to insert at end
3 to display
4 to delete element
5 to delete list
6 search
1
enter element to insert at beginning
2
enter 0 to exit
1 to insert at beginning
2 to insert at end
3 to display
4 to delete element
5 to delete list
6 search
1
enter element to insert at beginning
3
enter 0 to exit
1 to insert at beginning
2 to insert at end
3 to display
4 to delete element
5 to delete list
6 search
2
enter element to insert at end
9
enter 0 to exit
1 to insert at beginning
2 to insert at end
3 to display
4 to delete element
5 to delete list
6 search
2
enter element to insert at end
6
```



```
navin@DESKTOP-D40C38:/mnt/d/ds lab$
enter element to search
5
position of element is(0 if not found) : 3
enter 0 to exit
1 to insert at beginning
2 to insert at end
3 to display
4 to delete element
5 to delete list
6 search
4
enter element to delete
5
enter 0 to exit
1 to insert at beginning
2 to insert at end
3 to display
4 to delete element
5 to delete list
6 search
3
3 2 6
enter 0 to exit
1 to insert at beginning
2 to insert at end
3 to display
4 to delete element
5 to delete list
6 search
5
enter 0 to exit
1 to insert at beginning
2 to insert at end
3 to display
4 to delete element
5 to delete list
6 search
0
navin@DESKTOP-D40C38:/mnt/d/ds lab$
```

