1BM19CS159

Sahan R Kumar.

Lab Program - 9

Q) Write a program to implement doubly linked list with following operations:

i) Insert front
ii) Insert rear
iii) Delete front
iv) Delete rear
v) Display
vi) Insert after/before key node
vii) Search
viii) Delete duplicate.

A)

```
#include<stdio.h>
#include<conio.h>
#include<process.h>
#include<stdlib.h>
#include<malloc.h>
struct node{
int info;
struct node *llink;
struct node *rlink;
};
typedef struct node *NODE;

NODE getnode(){
NODE x;
x=(NODE)malloc(sizeof(struct node));
if(x==NULL){
printf("mem full\n");
exit(0);
}
return x;
}
void freenode(NODE x){
free(x);
}
NODE dinsert_front(int item, NODE head){
NODE temp, cur;
```

```
temp = getnode();
temp->info = item;
cur = head->rlink;
head->rlink = temp;
temp->llink = head;
temp->rlink = cur;
cur->llink = temp;
return head;
}

NODE dinsert_rear(int item, NODE head){
NODE temp, cur;
temp = getnode();
temp->info = item;
cur = head->llink;
head->llink = temp;
temp->rlink = head;
temp->llink = cur;
cur->rlink = temp;
return head;
}

NODE ddelete_front(NODE head){
NODE cur, next;
if(head->rlink == head){
printf("dq empty\n");
return head;
}
cur = head->rlink;
next = cur->rlink;
head->rlink = next;
next->llink = head;
printf("the node deleted is %d", cur->info);
freenode(cur);
return head;
}
```

```
NODE ddelete_rear(NODE head){
NODE cur, prev;
if(head->rlink==head){
printf(" dq empty \n");
return head;
}
cur=head->llink;
prev=cur->llink;
head->llink=prev;
prev->rlink=head;
printf("the node deleted is %d", cur->info);
freenode(cur);
return head;
}
void display(NODE head){
NODE temp;
if(head->rlink==head){
printf(" dq empty \n");
return;
}
printf("contents of dq \n");
temp=head->rlink;
while(temp!=head){
printf("%d ", temp->info);
temp=temp->rlink;
}
printf("\n");
}
NODE delete_all_key(int item, NODE head){
NODE prev, cur, next;
int count;
if(head->rlink==head){
printf("List Empty \n");
return head;
}
```

```
count 0;
cur = head->rlink;
cur = cur->rlink;
while(cur != head){
if(item != cur->info)
cur = cur->rlink;
else {
count++;
prev = cur->llink;
next = cur->rlink;
prev->rlink = next;
next->llink = prev;
freenode(cur);
cur = next;
}}
if(count == 0)
printf("key not found \n");
else
printf("key found at %d positions and are deleted \n", count);
return head;
}
NODE insert_leftpos(int item, NODE head){
NODE temp, cur, prev;
if(head->rlink == head){
printf("Empty \n");
return head;
}
cur = head->rlink;
while(cur != head){
if(item == cur->info) break;
cur = cur->rlink;
}
if(cur == head){
printf("key not found \n");
```

```
return head;
}

prev = cur->llink;
printf("Enter towards left of %d:", item);
temp = getnode();
scanf("%d", &temp->info);
prev->rlink = temp;
temp->llink = prev;
cur->llink = temp;
temp->rlink = cur;
return head;
}

NODE insert_right(int item, NODE head){
NODE temp, cur, prev;
if(head->llink == head){
printf("Empty \n");
return head;
}
cur = head->llink;
while(cur != head){
if(item == cur->info), break;
cur = cur->llink;
}
if(cur == head){
printf("key not found \n");
return head;
}
prev = cur->rlink;
printf("Enter towards right of %d:", item);
temp = getnode();
scanf("%d", &temp->info);
prev->llink = temp;
temp->rlink = prev;
cur->rlink = temp;
```

```
temp->llink = cur;
return head;
}

NODE search(NODE head){
NODE temp = head->rlink;
int count = 0; key, flag = 0;
printf("Enter the key : ");
scanf("%d", &key);
while(temp != head){
count++;
if(temp->info == key){
flag = 1;
printf("key %d found in position %d ", key, count);
}
temp = temp->rlink;
}
if(flag == 0){
printf("key is not found in list");
}
return head;
}

int main(){
NODE head, last;
int item, choice, option;
head = getnode();
head->rlink = head;
head->llink = head;
system("cls");
for(;;){
printf("\n1: insert front \n2: insert rear \n3: delete front \n
4: delete rear \n5: display \n6: Delete repeating occurences \n7:
search \n8: Insert node after/befor key node \n9: Exit \n");
printf("enter the choice \n");
scanf("%d", &choice);
```

```
switch (choice) {
case 1: printf (" enter the item at front end \n");
        scanf ("%d", &item);
        last = dinsert_front (item, head);
        break;
case 2: printf (" enter the item at rear end \n");
        scanf ("%d", &item);
        last = dinsert_rear (item, head);
        break;
case 3: last = ddelete_front (head);
        break;
case 4: last = ddelete_rear (head);
        break;
case 5: display (head);
        break;
case 6: printf (" enter the item \n");
        scanf ("%d", &item);
        last = delete_all_key (item, head);
        break;
case 8: printf (" press 1: for insert behind 2: for insert after ");
        scanf ("%d", &option);
        if (option == 1) {
        printf (" enter # key node \n");
        scanf ("%d", &item);
        last = insert_leftpos (item, head); }
        else if (option == 2) {
        printf (" enter key node \n");
        scanf ("%d", &item);
        last = insert_rightpos (item, head);
        break;
    }
  case 7: search (head);
          break;
  default: exit(0);
```

```
    }}
getch();
return(0);
}
```

C:\Users\sohan\Desktop\C Programs\Data Structures Lab\doubly linked list\doubly_linked_lists.cpp - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

(globals)

doubly_linked_lists.cpp

```
#include<stdio.h>
#include<conio.h>
#include<process.h>
#include<stdlib.h>
#include <malloc.h>
struct node
{
      int info;
      struct node *llink;
      struct node *rlink;
      };
typedef struct node *NODE;


NODE getnode()
{
     NODE x;
     x=(NODE)malloc(sizeof(struct node));
     if(x==NULL)
     {
         printf("mem full\n");
         exit(0);
         }
     return x;
     }


void freenode(NODE x)
{
     free(x);
}


NODE dinsert_front(int item,NODE head)
{
NODE temp,cur;
temp=getnode();
```

Compiler Resources Compile Log Debug Find Results

Line: 126 Col: 8 Sel: 0 Lines: 279 Length: 5035 Insert Done parsing in 0.016 seconds

Type here to search

ENG 18:29 15-12-2020

C:\Users\sohan\Desktop\C Programs\Data Structures Lab\doubly linked list\doubly_linked_lists.cpp - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

(globals)

doubly_linked_lists.cpp

```
37  temp=getnode();
38  temp->info=item;
39  cur=head->rlink;
40  head->rlink=temp;
41  temp->llink=head;
42  temp->rlink=cur;
43  cur->llink=temp;
44  return head;
45  }
46
47
48  NODE dinsert_rear(int item,NODE head)
49  {
50  NODE temp,cur;
51  temp=getnode();
52  temp->info=item;
53  cur=head->llink;
54  head->llink=temp;
55  temp->rlink=head;
56  temp->llink=cur;
57  cur->rlink=temp;
58  return head;
59  }
60
61
62  NODE ddelete_front(NODE head)
63  {
64  NODE cur,next;
65  if(head->rlink==head)
66  {
67  printf("dq empty\n");
68  return head;
69  }
70  cur=head->rlink;
71  next=cur->rlink;
72  head->rlink=next;
73  next->llink=head;
```

Compiler Resources Compile Log Debug Find Results

Line: 126 Col: 8 Sel: 0 Lines: 279 Length: 5035 Insert Done parsing in 0.016 seconds

Type here to search

ENG 18:29 15-12-2020

C:\Users\sohan\Desktop\C Programs\Data Structures Lab\doubly linked list\doubly_linked_lists.cpp - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

(globals)

doubly_linked_lists.cpp

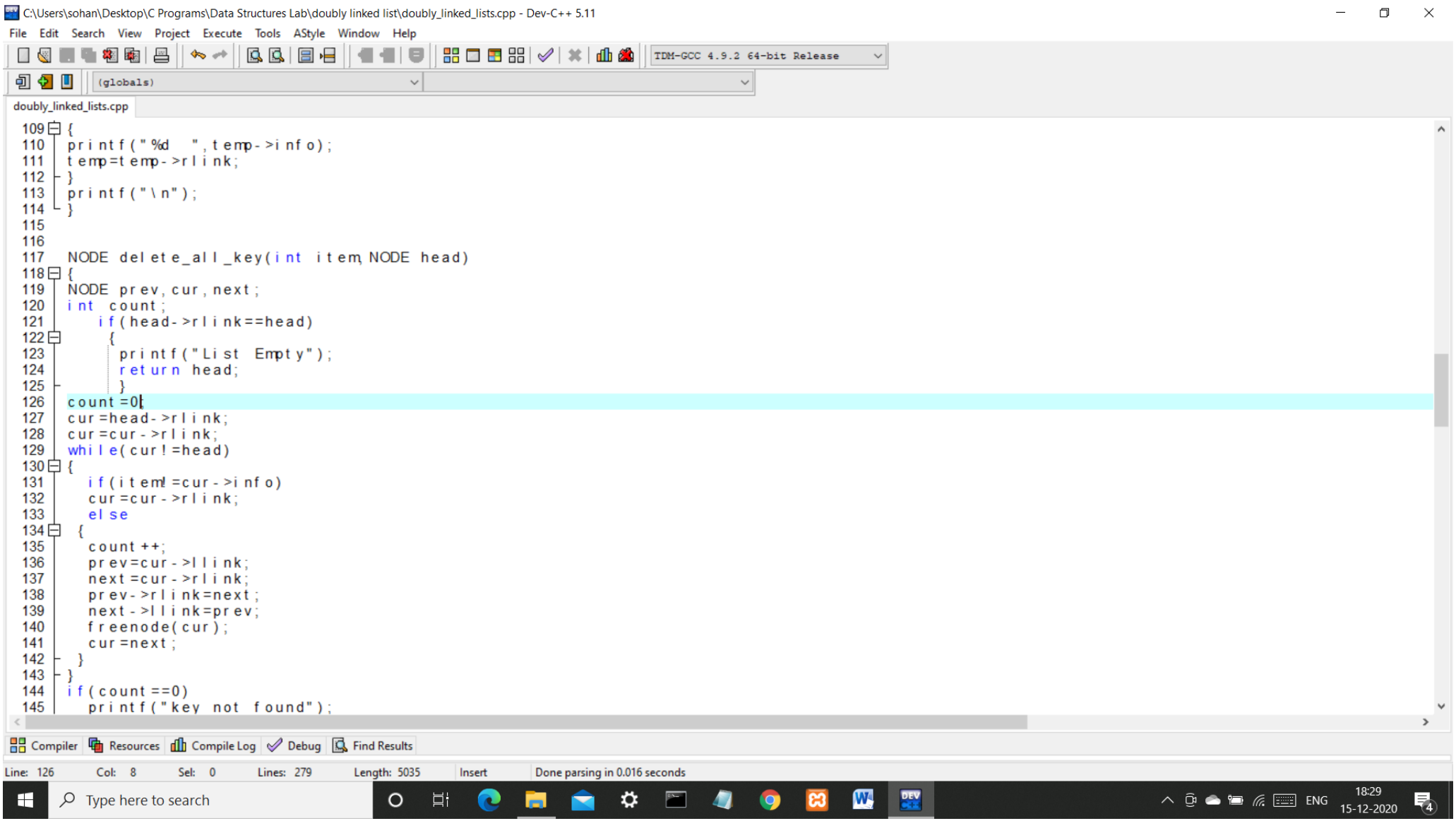
```
 72  head->rlink=next;
 73  next->llink=head;
 74  printf("the node deleted is %d",cur->info);
 75  freenode(cur);
 76  return head;
 77  }
 78
 79
 80  NODE ddelete_rear(NODE head)
 81  {
 82  NODE cur,prev;
 83  if(head->rlink==head)
 84  {
 85  printf("dq empty\n");
 86  return head;
 87  }
 88  cur=head->llink;
 89  prev=cur->llink;
 90  head->llink=prev;
 91  prev->rlink=head;
 92  printf("the node deleted is %d",cur->info);
 93  freenode(cur);
 94  return head;
 95  }
 96
 97
 98  void display(NODE head)
 99  {
100  NODE temp;
101  if(head->rlink==head)
102  {
103  printf("dq empty\n");
104  return;
105  }
106  printf("contents of dq\n");
107  temp=head->rlink;
108  while(temp!=head)
```

Compiler Resources Compile Log Debug Find Results

Line: 126 Col: 8 Sel: 0 Lines: 279 Length: 5035 Insert Done parsing in 0.016 seconds

Type here to search

ENG 18:29 15-12-2020


C:\Users\sohan\Desktop\C Programs\Data Structures Lab\doubly linked list\doubly_linked_lists.cpp - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Release
(globals)
doubly_linked_lists.cpp
{
printf("%d ",temp->info);
temp=temp->rlink;
}
printf("\n");
}
NODE delete_all_key(int item,NODE head)
{
NODE prev,cur,next;
int count;
if(head->rlink==head)
{
printf("List Empty");
return head;
}
count=0;
cur=head->rlink;
cur=cur->rlink;
while(cur!=head)
{
if(item!=cur->info)
cur=cur->rlink;
else
{
count++;
prev=cur->llink;
next=cur->rlink;
prev->rlink=next;
next->llink=prev;
freenode(cur);
cur=next;
}
}
if(count==0)
printf("key not found");
Compiler Resources Compile Log Debug Find Results
Line: 126 Col: 8 Sel: 0 Lines: 279 Length: 5035 Insert Done parsing in 0.016 seconds
Type here to search
ENG
18:29
15-12-2020

C:\Users\sohan\Desktop\C Programs\Data Structures Lab\doubly linked list\doubly_linked_lists.cpp - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

(globals)

doubly_linked_lists.cpp

```
146     else
147      printf("key found at %d positions and are deleted\n", count);
148
149   return head;
150
151
152   }
153   NODE insert_leftpos(int item,NODE head){
154   NODE temp,cur,prev;
155   if(head->rlink==head){
156   printf("Empty \n");
157   return head;
158   }
159   cur=head->rlink;
160   while(cur!=head){
161   if(item==cur->info)break;
162   cur=cur->rlink;
163   }
164   if(cur==head){
165   printf("key nt found \n");
166   return head;
167   }
168   prev=cur->llink;
169   printf("Enter towards left of %d:",item);
170   temp=getnode();
171   scanf("%d",&temp->info);
172   prev->rlink=temp;
173   temp->llink=prev;
174   cur->llink=temp;
175   temp->rlink=cur;
176   return head;}
177   NODE insert_rightpos(int item,NODE head){
178   NODE temp,cur,prev;
179   if(head->llink==head){
180   printf("Empty \n");
181   return head;
182   }
```

Compiler Resources Compile Log Debug Find Results

Line: 126 Col: 8 Sel: 0 Lines: 279 Length: 5035 Insert Done parsing in 0.016 seconds

Type here to search

ENG 18:29 15-12-2020

C:\Users\sohan\Desktop\C Programs\Data Structures Lab\doubly linked list\doubly_linked_lists.cpp - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

(globals)

doubly_linked_lists.cpp

```
152 }
153 NODE insert_leftpos(int item,NODE head){
154 NODE temp,cur,prev;
155 if(head->rlink==head){
156 printf("Empty \n");
157 return head;
158 }
159 cur=head->rlink;
160 while(cur!=head){
161 if(item==cur->info)break;
162 cur=cur->rlink;
163 }
164 if(cur==head){
165 printf("key nt found \n");
166 return head;
167 }
168 prev=cur->llink;
169 printf("Enter towards left of %d:",item);
170 temp=getnode();
171 scanf("%d",&temp->info);
172 prev->rlink=temp;
173 temp->llink=prev;
174 cur->llink=temp;
175 temp->rlink=cur;
176 return head;}
177 NODE insert_rightpos(int item,NODE head){
178 NODE temp,cur,prev;
179 if(head->llink==head){
180 printf("Empty \n");
181 return head;
182 }
183 cur=head->llink;
184 while(cur!=head){
185 if(item==cur->info)break;
186 cur=cur->llink;
187 }
188 if(cur==head){
```

Compiler Resources Compile Log Debug Find Results

Line: 126 Col: 8 Sel: 0 Lines: 279 Length: 5035 Insert Done parsing in 0.016 seconds

Type here to search

ENG 18:29 15-12-2020

C:\Users\sohan\Desktop\C Programs\Data Structures Lab\doubly linked list\doubly_linked_lists.cpp - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

(globals)

doubly_linked_lists.cpp

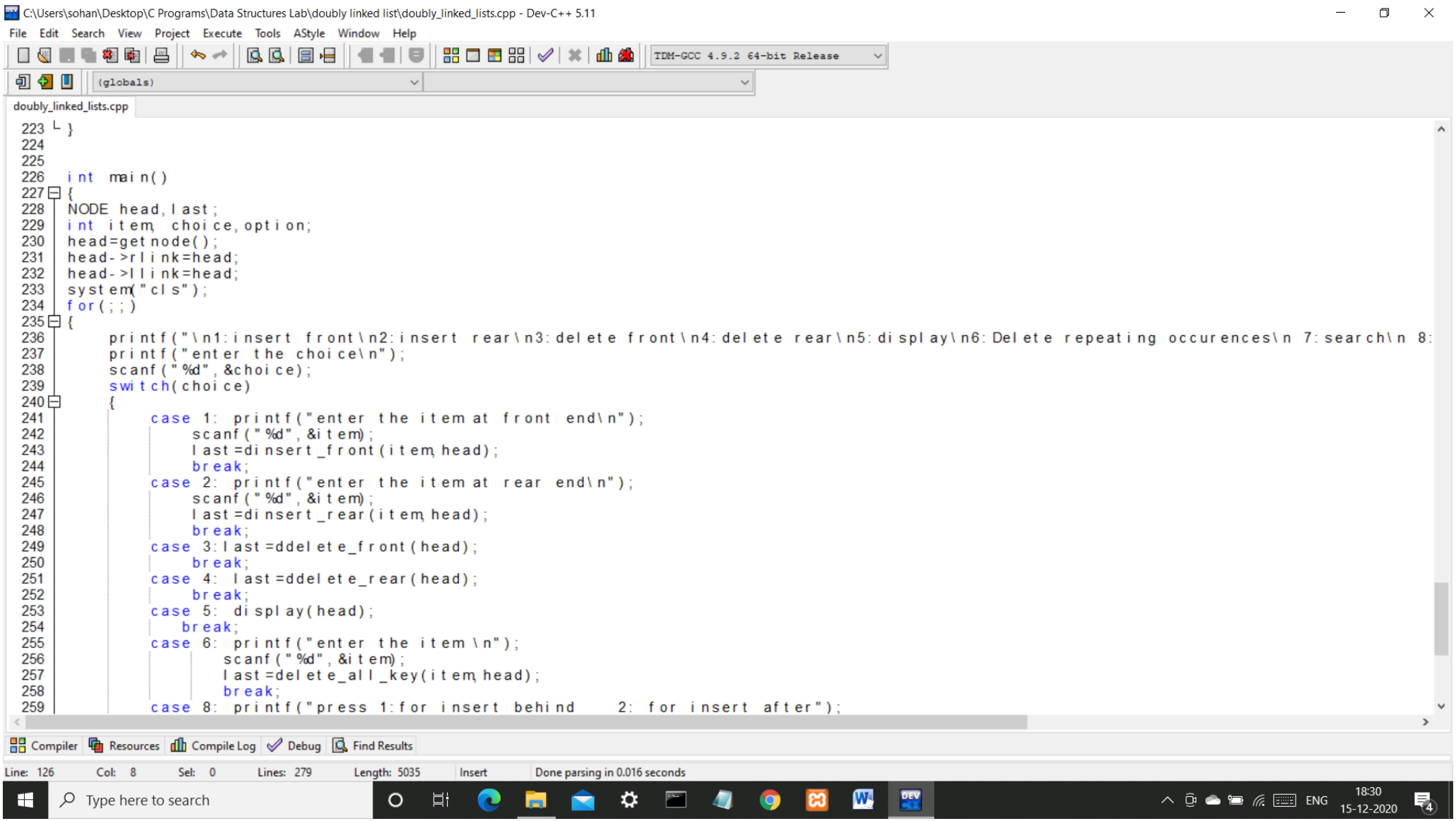
```
187 }
188 if(cur==head){
189 printf("key nt found \n");
190 return head;
191 }
192 prev=cur->rlink;
193 printf("Enter towards right of %d:",item);
194 temp=getnode();
195 scanf("%d",&temp->info);
196 prev->llink=temp;
197 temp->rlink=prev;
198 cur->rlink=temp;
199 temp->llink=cur;
200 return head;}
201
202
203 NODE search(NODE head)
204 {
205     NODE temp=head->rlink;
206     int count=0,key,flag=0;
207     printf("Enter the key :");
208     scanf("%d",&key);
209     while(temp!=head)
210     {          count++;
211         if(temp->info==key)
212         {
213             flag=1;
214             printf("key %d found in position %d",key,count);
215         }temp=temp->rlink;
216     }
217     if(flag==0)
218     {
219         printf("Key is not found in list");
220     }
221     return head;
222
223 }
```

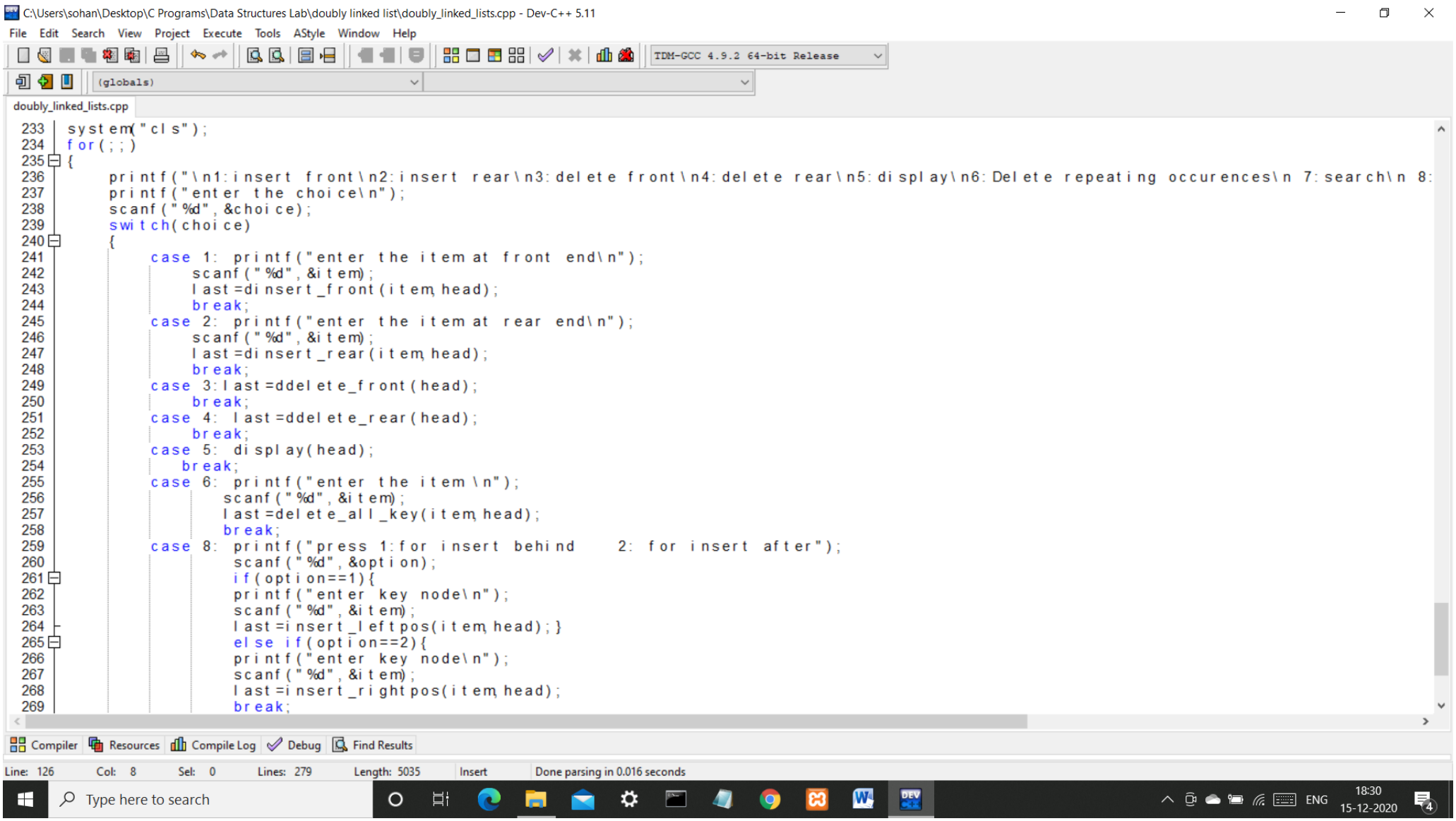
Compiler Resources Compile Log Debug Find Results

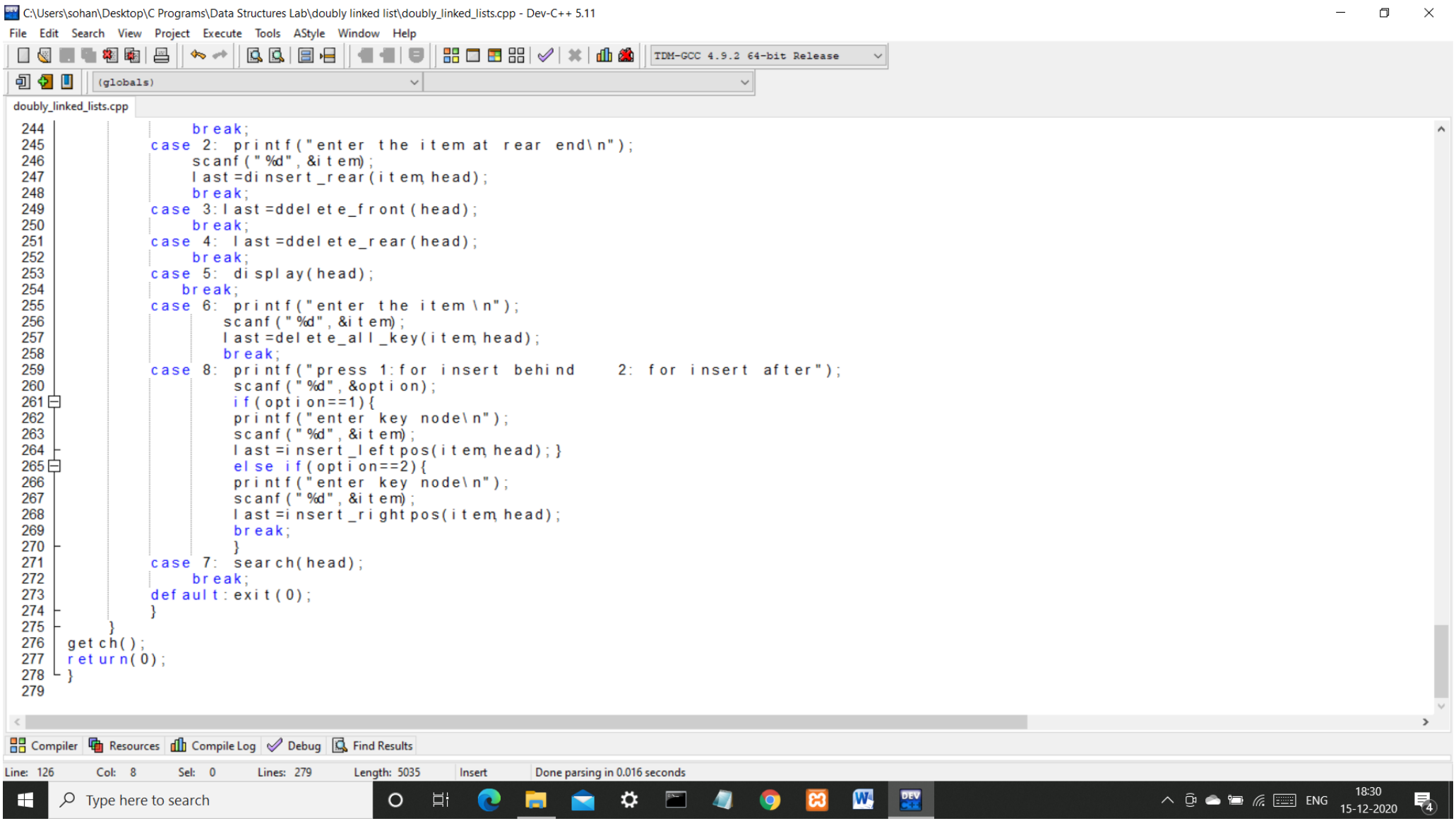
Line: 126 Col: 8 Sel: 0 Lines: 279 Length: 5035 Insert Done parsing in 0.016 seconds

Type here to search

ENG 18:30 15-12-2020


C:\Users\sohan\Desktop\C Programs\Data Structures Lab\doubly linked list\doubly_linked_lists.cpp - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Release
(globals)
doubly_linked_lists.cpp
223 }
224
225
226 int main()
227 {
228 NODE head,last;
229 int item, choice,option;
230 head=getnode();
231 head->rlink=head;
232 head->llink=head;
233 system("cls");
234 for(;;)
235 {
236 printf("\n1:insert front\n2:insert rear\n3:delete front\n4:delete rear\n5:display\n6:Delete repeating occurences\n 7:search\n 8:
237 printf("enter the choice\n");
238 scanf("%d",&choice);
239 switch(choice)
240 {
241 case 1: printf("enter the item at front end\n");
242 scanf("%d",&item);
243 last=dinsert_front(item,head);
244 break;
245 case 2: printf("enter the item at rear end\n");
246 scanf("%d",&item);
247 last=dinsert_rear(item,head);
248 break;
249 case 3:last=ddelete_front(head);
250 break;
251 case 4: last=ddelete_rear(head);
252 break;
253 case 5: display(head);
254 break;
255 case 6: printf("enter the item\n");
256 scanf("%d",&item);
257 last=delete_all_key(item,head);
258 break;
259 case 8: printf("press 1:for insert behind 2: for insert after");
Compiler Resources Compile Log Debug Find Results
Line: 126 Col: 8 Sel: 0 Lines: 279 Length: 5035 Insert Done parsing in 0.016 seconds
Type here to search
ENG
18:30
15-12-2020


C:\Users\sohan\Desktop\C Programs\Data Structures Lab\doubly linked list\doubly_linked_lists.cpp - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Release
(globals)
doubly_linked_lists.cpp
233 system("cls");
234 for(;;)
235 {
236 printf("\n1:insert front\n2:insert rear\n3:delete front\n4:delete rear\n5:display\n6:Delete repeating occurences\n 7:search\n 8:
237 printf("enter the choice\n");
238 scanf("%d",&choice);
239 switch(choice)
240 {
241 case 1: printf("enter the item at front end\n");
242 scanf("%d",&item);
243 last=dinsert_front(item,head);
244 break;
245 case 2: printf("enter the item at rear end\n");
246 scanf("%d",&item);
247 last=dinsert_rear(item,head);
248 break;
249 case 3:last=ddelete_front(head);
250 break;
251 case 4: last=ddelete_rear(head);
252 break;
253 case 5: display(head);
254 break;
255 case 6: printf("enter the item\n");
256 scanf("%d",&item);
257 last=delete_all_key(item,head);
258 break;
259 case 8: printf("press 1:for insert behind 2: for insert after");
260 scanf("%d",&option);
261 if(option==1){
262 printf("enter key node\n");
263 scanf("%d",&item);
264 last=insert_leftpos(item,head);}
265 else if(option==2){
266 printf("enter key node\n");
267 scanf("%d",&item);
268 last=insert_rightpos(item,head);
269 break;
Compiler Resources Compile Log Debug Find Results
Line: 126 Col: 8 Sel: 0 Lines: 279 Length: 5035 Insert Done parsing in 0.016 seconds
Type here to search
ENG
18:30
15-12-2020


C:\Users\sohan\Desktop\C Programs\Data Structures Lab\doubly linked list\doubly_linked_lists.cpp - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Release
(globals)
doubly_linked_lists.cpp
break;
case 2: printf("enter the item at rear end\n");
scanf("%d",&item);
last=dinsert_rear(item,head);
break;
case 3:last=ddelete_front(head);
break;
case 4: last=ddelete_rear(head);
break;
case 5: display(head);
break;
case 6: printf("enter the item\n");
scanf("%d",&item);
last=delete_all_key(item,head);
break;
case 8: printf("press 1:for insert behind 2: for insert after");
scanf("%d",&option);
if(option==1){
printf("enter key node\n");
scanf("%d",&item);
last=insert_leftpos(item,head);}
else if(option==2){
printf("enter key node\n");
scanf("%d",&item);
last=insert_rightpos(item,head);
break;
}
case 7: search(head);
break;
default:exit(0);
}
}
getch();
return(0);
}
Compiler Resources Compile Log Debug Find Results
Line: 126 Col: 8 Sel: 0 Lines: 279 Length: 5035 Insert Done parsing in 0.016 seconds
Type here to search
ENG
18:30
15-12-2020

C:\Users\sohan\Desktop\C Programs\Data Structures Lab\doubly linked list\doubly_linked_lists.exe

```
1:insert front
2:insert rear
3:delete front
4:delete rear
5:display
6:Delete repeating occurences
 7:search
 8:Inserting node before/after key node
9:Exit
enter the choice
1
enter the item at front end
34

1:insert front
2:insert rear
3:delete front
4:delete rear
5:display
6:Delete repeating occurences
 7:search
 8:Inserting node before/after key node
9:Exit
enter the choice
1
enter the item at front end
55

1:insert front
2:insert rear
3:delete front
4:delete rear
5:display
6:Delete repeating occurences
 7:search
 8:Inserting node before/after key node
9:Exit
enter the choice
1
enter the item at front end
67

1:insert front
2:insert rear
3:delete front
4:delete rear
5:display
6:Delete repeating occurences
 7:search
```

C:\Users\sohan\Desktop\C Programs\Data Structures Lab\doubly linked list\doubly_linked_lists.exe

```
1:insert front
2:insert rear
3:delete front
4:delete rear
5:display
6:Delete repeating occurences
 7:search
 8:Inserting node before/after key node
9:Exit
enter the choice
1
enter the item at front end
34

1:insert front
2:insert rear
3:delete front
4:delete rear
5:display
6:Delete repeating occurences
 7:search
 8:Inserting node before/after key node
9:Exit
enter the choice
1
enter the item at front end
55

1:insert front
2:insert rear
3:delete front
4:delete rear
5:display
6:Delete repeating occurences
 7:search
 8:Inserting node before/after key node
9:Exit
enter the choice
1
enter the item at front end
67

1:insert front
2:insert rear
3:delete front
4:delete rear
5:display
6:Delete repeating occurences
 7:search
```

C:\Users\sohan\Desktop\C Programs\Data Structures Lab\doubly linked list\doubly_linked_lists.exe

```
5:display
6:Delete repeating occurences
 7:search
 8:Inserting node before/after key node
9:Exit
enter the choice
2
enter the item at rear end
9

1:insert front
2:insert rear
3:delete front
4:delete rear
5:display
6:Delete repeating occurences
 7:search
 8:Inserting node before/after key node
9:Exit
enter the choice
2
enter the item at rear end
13

1:insert front
2:insert rear
3:delete front
4:delete rear
5:display
6:Delete repeating occurences
 7:search
 8:Inserting node before/after key node
9:Exit
enter the choice
5
contents of dq
67  55  34  9  13

1:insert front
2:insert rear
3:delete front
4:delete rear
5:display
6:Delete repeating occurences
 7:search
 8:Inserting node before/after key node
9:Exit
enter the choice
7
Enter the key :55
```

C:\Users\sohan\Desktop\C Programs\Data Structures Lab\doubly linked list\doubly_linked_lists.exe

```
7
Enter the key :55
key 55 found in position 2
1:insert front
2:insert rear
3:delete front
4:delete rear
5:display
6:Delete repeating occurences
 7:search
 8:Inserting node before/after key node
9:Exit
enter the choice
2
enter the item at rear end
33

1:insert front
2:insert rear
3:delete front
4:delete rear
5:display
6:Delete repeating occurences
 7:search
 8:Inserting node before/after key node
9:Exit
enter the choice
1
enter the item at front end
1

1:insert front
2:insert rear
3:delete front
4:delete rear
5:display
6:Delete repeating occurences
 7:search
 8:Inserting node before/after key node
9:Exit
enter the choice
5
contents of dq
1  67  55  34  9  13  33

1:insert front
2:insert rear
3:delete front
4:delete rear
5:display
```

C:\Users\sohan\Desktop\C Programs\Data Structures Lab\doubly linked list\doubly_linked_lists.exe

```
2:insert rear
3:delete front
4:delete rear
5:display
6:Delete repeating occurences
 7:search
 8:Inserting node before/after key node
9:Exit
enter the choice
3
the node deleted is 1
1:insert front
2:insert rear
3:delete front
4:delete rear
5:display
6:Delete repeating occurences
 7:search
 8:Inserting node before/after key node
9:Exit
enter the choice
4
the node deleted is 33
1:insert front
2:insert rear
3:delete front
4:delete rear
5:display
6:Delete repeating occurences
 7:search
 8:Inserting node before/after key node
9:Exit
enter the choice
5
contents of dq
67  55  34  9  13

1:insert front
2:insert rear
3:delete front
4:delete rear
5:display
6:Delete repeating occurences
 7:search
 8:Inserting node before/after key node
9:Exit
enter the choice
2
enter the item at rear end
9
```

C:\Users\sohan\Desktop\C Programs\Data Structures Lab\doubly linked list\doubly_linked_lists.exe

```
4:delete rear
5:display
6:Delete repeating occurences
 7:search
 8:Inserting node before/after key node
9:Exit
enter the choice
2
enter the item at rear end
9

1:insert front
2:insert rear
3:delete front
4:delete rear
5:display
6:Delete repeating occurences
 7:search
 8:Inserting node before/after key node
9:Exit
enter the choice
1
enter the item at front end
9

1:insert front
2:insert rear
3:delete front
4:delete rear
5:display
6:Delete repeating occurences
 7:search
 8:Inserting node before/after key node
9:Exit
enter the choice
5
contents of dq
9  67  55  34  9  13  9

1:insert front
2:insert rear
3:delete front
4:delete rear
5:display
6:Delete repeating occurences
 7:search
 8:Inserting node before/after key node
9:Exit
enter the choice
6
```

C:\Users\sohan\Desktop\C Programs\Data Structures Lab\doubly linked list\doubly_linked_lists.exe

```
9:Exit
enter the choice
6
enter the item
9
key found at 2 positions and are deleted

1:insert front
2:insert rear
3:delete front
4:delete rear
5:display
6:Delete repeating occurences
 7:search
 8:Inserting node before/after key node
9:Exit
enter the choice
5
contents of dq
9  67  55  34  13

1:insert front
2:insert rear
3:delete front
4:delete rear
5:display
6:Delete repeating occurences
 7:search
 8:Inserting node before/after key node
9:Exit
enter the choice
8
press 1:for insert behind    2: for insert after1
enter key node
67
Enter towards left of 67:12
Enter the key :12
key 12 found in position 2
1:insert front
2:insert rear
3:delete front
4:delete rear
5:display
6:Delete repeating occurences
 7:search
 8:Inserting node before/after key node
9:Exit
enter the choice
5
contents of dq
```

C:\Users\sohan\Desktop\C Programs\Data Structures Lab\doubly linked list\doubly_linked_lists.exe

```
enter the choice
8
press 1:for insert behind    2: for insert after1
enter key node
67
Enter towards left of 67:12
Enter the key :12
key 12 found in position 2
1:insert front
2:insert rear
3:delete front
4:delete rear
5:display
6:Delete repeating occurences
 7:search
 8:Inserting node before/after key node
9:Exit
enter the choice
5
contents of dq
9  12  67  55  34  13

1:insert front
2:insert rear
3:delete front
4:delete rear
5:display
6:Delete repeating occurences
 7:search
 8:Inserting node before/after key node
9:Exit
enter the choice
9

--------------------------------
Process exited after 150.6 seconds with return value 0
Press any key to continue . . .
```