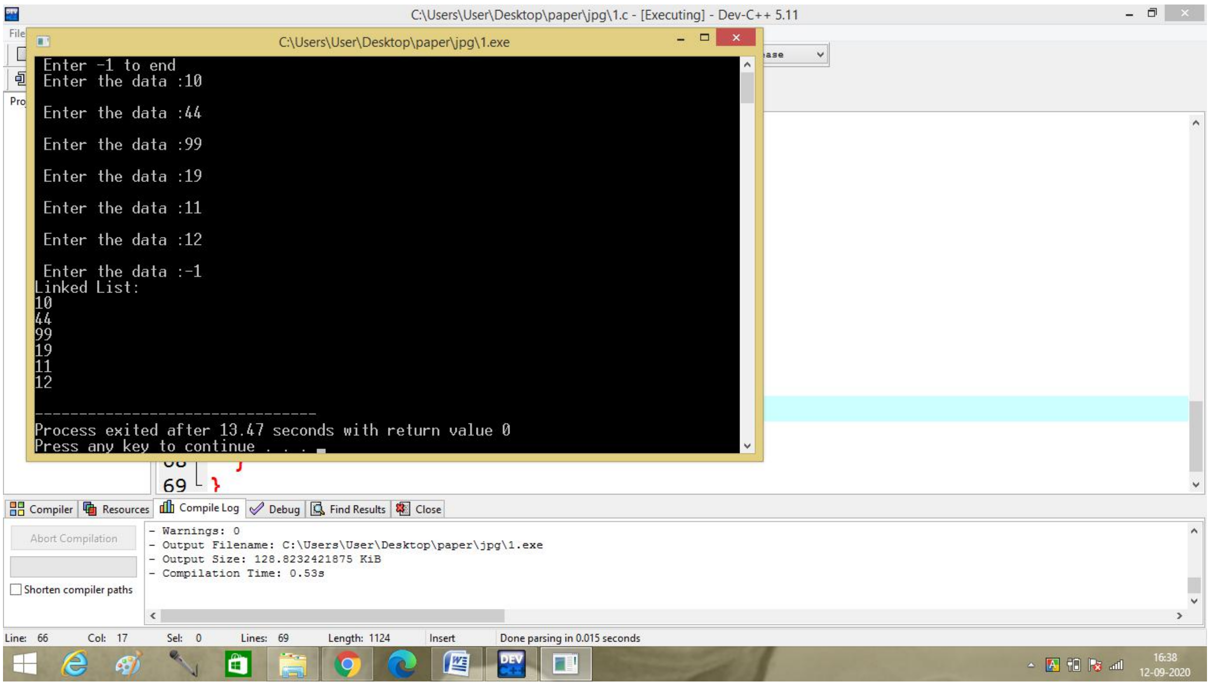
1.WAP to create a double linked list of n nodes and display the linked list by using suitable user defined functions for create and display operations.

```
#include<stdio.h>
#include<malloc.h>
struct node
{
  int data;
  struct node *prev;
  struct node *next;
};
struct node *start=NULL;
struct node * create(struct node *);
void display(struct node *);
void main()
{
start=create(start);
display(start);
}
struct node * create(struct node *start)
{

  struct node *new_node,*ptr;
  int num;
  printf("\n Enter -1 to end");
  printf("\n Enter the data :");
  scanf("%d",&num);

  while(num != -1)
  {
  new_node=(struct node *)malloc(sizeof(struct node));
   new_node->data=num;
   if(start==NULL)
   {
     new_node->next=NULL;
     new_node->prev=NULL;
     start=new_node;
   }
   else
   {
   ptr=start;
     while(ptr->next != NULL)
        ptr=ptr->next;
     ptr->next=new_node;
     new_node->prev=ptr;
     new_node->next=NULL;
    }
    printf("\n Enter the data :");
    scanf("%d",&num);
   }
 return start;
}
void display(struct node *start)
{
  struct node *ptr;
  ptr=start;
  printf("Linked List:\n");
  while(ptr != NULL)
  {
    printf("%d",ptr->data);
    ptr=ptr->next;
  }
}
```

OUTPUT :



```
 Enter -1 to end
 Enter the data :10

 Enter the data :44

 Enter the data :99

 Enter the data :19

 Enter the data :11

 Enter the data :12

 Enter the data :-1
Linked List:
10
44
99
19
11
12

--------------------------------
Process exited after 13.47 seconds with return value 0
Press any key to continue . . .
```

```
- Warnings: 0
- Output Filename: C:\Users\User\Desktop\paper\jpg\1.exe
- Output Size: 128.8232421875 KiB
- Compilation Time: 0.53s
```

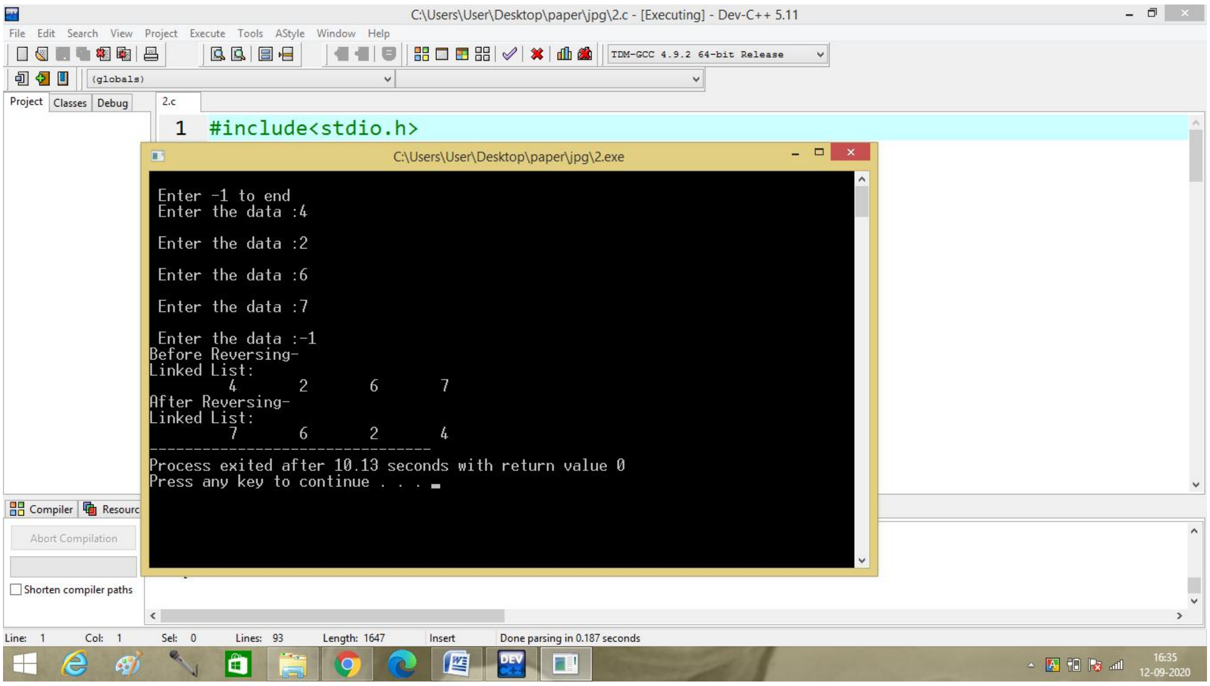
Line: 66 Col: 17 Sel: 0 Lines: 69 Length: 1124 Insert Done parsing in 0.015 seconds

2.WAP to reverse the sequence elements in a double linked list.

```
#include<stdio.h>
#include<malloc.h>

struct node
{
  int data;
  struct node *prev;
  struct node *next;
};
struct node *start=NULL;
struct node * create(struct node *);
void display(struct node *);
struct node * reverse(struct node *);
void main()
{
start=create(start);
printf("Before Reversing-\n");
```

```
display(start);
printf("\n");
start=reverse(start);
printf("After Reversing-\n");
display(start);
}
struct node * reverse(struct node *start)
{
    struct node *current,*temp;
    temp=NULL;
    current=start;

    while(current!=NULL)
    {
   temp = current->prev;
        current->prev = current->next;
        current->next = temp;
        current = current->prev;

    }
    if(temp != NULL )
        start = temp->prev;
    return start;
};
struct node * create(struct node *start)
{
  struct node *new_node,*ptr;
  int num;
  printf("\n Enter -1 to end");
  printf("\n Enter the data :");
  scanf("%d",&num);
  while(num != -1)
  {
   new_node=(struct node *)malloc(sizeof(struct node));
   new_node->data=num;
   if(start==NULL)
   {
     new_node->next=NULL;
     new_node->prev=NULL;
     start=new_node;
   }
   else
   {
     ptr=start;
     while(ptr->next != NULL)
        ptr=ptr->next;
     ptr->next=new_node;
     new_node->prev=ptr;
     new_node->next=NULL;
    }
    printf("\n Enter the data :");
    scanf("%d",&num);
   }
 return start;
}
void display(struct node *start)
{
  struct node *ptr;
  ptr=start;
  printf("Linked List:\n");
  while(ptr != NULL)
  {
    printf("\t %d",ptr->data);
    ptr=ptr->next;
  }
}
```

OUTPUT :


C:\Users\User\Desktop\paper\jpg\2.c - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Release
(globals)
Project Classes Debug
2.c
C:\Users\User\Desktop\paper\jpg\2.exe
Compiler Resources
Abort Compilation
Shorten compiler paths
Line: 1 Col: 1 Sel: 0 Lines: 93 Length: 1647 Insert Done parsing in 0.187 seconds
16:35
12-09-2020

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

```
 Enter -1 to end
 Enter the data :4

 Enter the data :2

 Enter the data :6

 Enter the data :7

 Enter the data :-1
Before Reversing-
Linked List:
         4       2       6       7
After Reversing-
Linked List:
         7       6       2       4
--------------------------------
Process exited after 10.13 seconds with return value 0
Press any key to continue . . .
```

3.Write a menu driven program to perform the following operations in a double linked list by using suitable user defined functions for each case.

a) Traverse the list forward

b) Traverse the list backward

c) Check if the list is empty

d) Insert a node at the certain position (at beginning/end/any position)

e) Delete a node at the certain position (at beginning/end/any position)

f) Delete a node for the given key

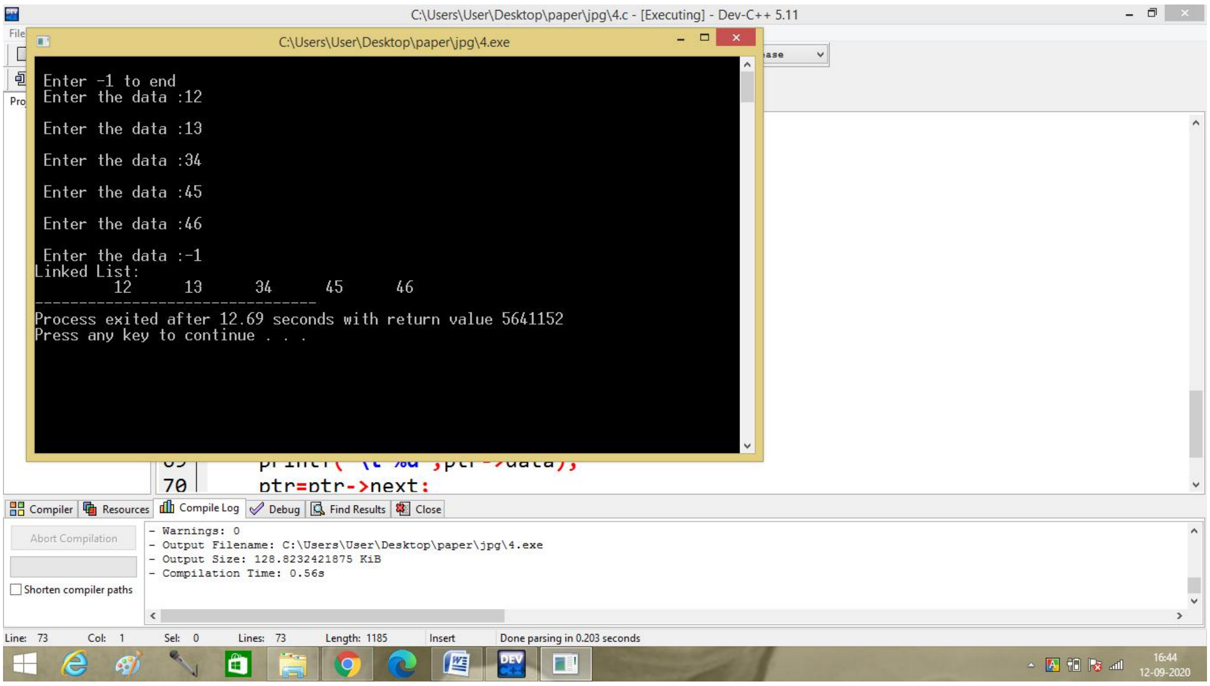
g) Count the total number of nodes

h) Search for an element in the linked list Verify & validate each function from main method

4.WAP to create a single circular double linked list of n nodes and display the linked list by using suitable user defined functions for create and display operations.

```
#include<stdio.h>

#include<malloc.h>


struct node

{

  int data;

  struct node *prev;

  struct node *next;

};

struct node *start=NULL;


struct node * create(struct node *);

void display(struct node *);


void main()

{

start=create(start);

display(start);

}

struct node * create(struct node *start)

{

  struct node *new_node,*ptr;

  int num;

  printf("\n Enter -1 to end");

  printf("\n Enter the data :");

  scanf("%d",&num);


  while(num != -1)

  {
   new_node=(struct node *)malloc(sizeof(struct node));
   new_node->data=num;

   if(start==NULL)
   {
     new_node->next=NULL;
     new_node->prev=NULL;
     start=new_node;
   }
   else
   {
     ptr=start;
     while(ptr->next != NULL)
        ptr=ptr->next;
     ptr->next=new_node;
     new_node->prev=ptr;
     new_node->next=NULL;
    }
    printf("\n Enter the data :");
    scanf("%d",&num);
   }
   start->prev=new_node;
   new_node->next=start;
   return start;
}
void display(struct node *start)
{
  struct node *ptr;
  ptr=start;
  printf("Linked List:\n");
  do
  {
    printf("\t %d",ptr->data);
    ptr=ptr->next;
  }
while(ptr!=start);
}
```

OUTPUT :



C:\Users\User\Desktop\paper\jpg\4.c - [Executing] - Dev-C++ 5.11

C:\Users\User\Desktop\paper\jpg\4.exe

```
 Enter -1 to end
 Enter the data :12

 Enter the data :13

 Enter the data :34

 Enter the data :45

 Enter the data :46

 Enter the data :-1
Linked List:
          12      13      34      45      46
--------------------------------
Process exited after 12.69 seconds with return value 5641152
Press any key to continue . . .
```

```
- Warnings: 0
- Output Filename: C:\Users\User\Desktop\paper\jpg\4.exe
- Output Size: 128.8232421875 KiB
- Compilation Time: 0.56s
```

5.WAP to remove the duplicates in a sorted double linked list.

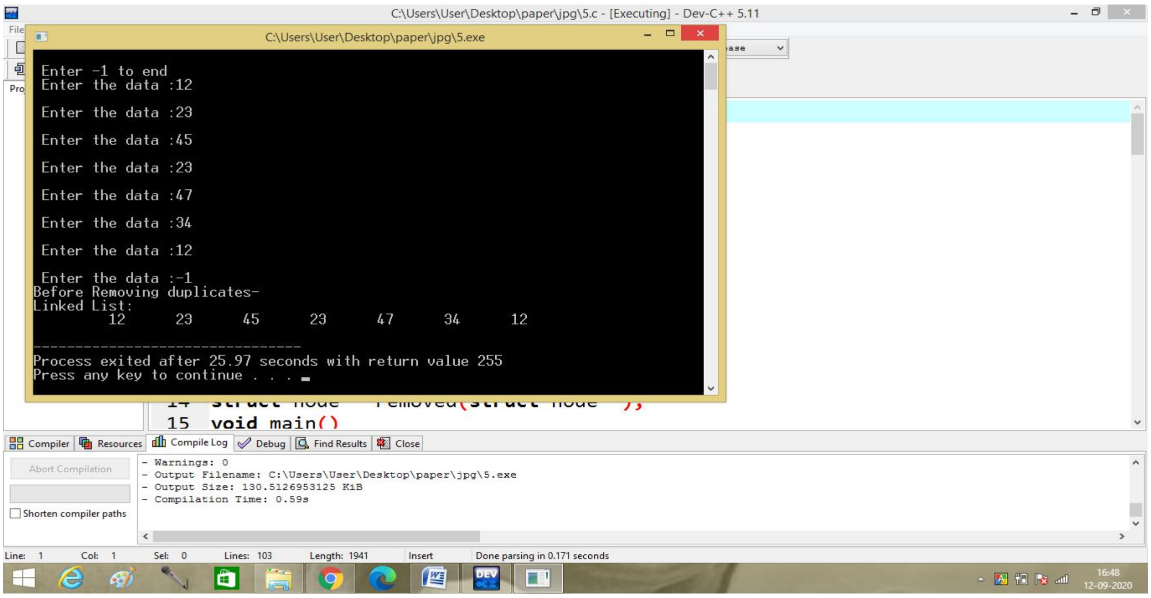
```
#include<stdio.h>
#include<malloc.h>
struct node
{
  int data;
  struct node *prev;
  struct node *next;
};
struct node *start=NULL;
struct node * create(struct node *);
void display(struct node *);
```

```
struct node * removed(struct node *);
void main()
{
start=create(start);
printf("Before Removing duplicates-\n");
display(start);
printf("\n");
start=removed(start);
printf("After Removing duplicates-\n");
display(start);
}
struct node * removed(struct node *start)
{
   struct node *ptr,*p,*q;
   ptr=start;
    while(ptr->next!=NULL)
    {
   p=ptr->next;
        while(p!=NULL)
        {
            if(ptr->data==p->data)
             {
                q=p->next;
                if(ptr->next==NULL)
                 {
                     p->prev->next=NULL;
                 }
             else
             {
              p->prev->next=p->next;
                 p->next->prev=p->prev;
             }
```

```
                p=q;
            }
            else
            p=p->next;
        }
        ptr=ptr->next;
    }
 return start;
}
struct node * create(struct node *start)
{
  struct node *new_node,*ptr;
  int num;
  printf("\n Enter -1 to end");
  printf("\n Enter the data :");
  scanf("%d",&num);
  while(num != -1)
  {
   new_node=(struct node *)malloc(sizeof(struct node));
   new_node->data=num;
   if(start==NULL)
   {
     new_node->next=NULL;
     new_node->prev=NULL;
     start=new_node;
   }
   else
   {
     ptr=start;
     while(ptr->next != NULL)
        ptr=ptr->next;
```

```
         ptr->next=new_node;
         new_node->prev=ptr;
         new_node->next=NULL;
        }
        printf("\n Enter the data :");
        scanf("%d",&num);
      }
  return start;
}
void display(struct node *start)
{
    struct node *ptr;
    ptr=start;
    printf("Linked List:\n");
    while(ptr != NULL)
    {
        printf("\t %d",ptr->data);
        ptr=ptr->next;
    }
}
```

OUTPUT :



C:\Users\User\Desktop\paper\jpg\5.c - [Executing] - Dev-C++ 5.11
C:\Users\User\Desktop\paper\jpg\5.exe
Compiler
Resources
Compile Log
Debug
Find Results
Close
Abort Compilation
□ Shorten compiler paths
Line: 1 Col: 1 Sel: 0 Lines: 103 Length: 1941 Insert Done parsing in 0.171 seconds
16:48
12-09-2020

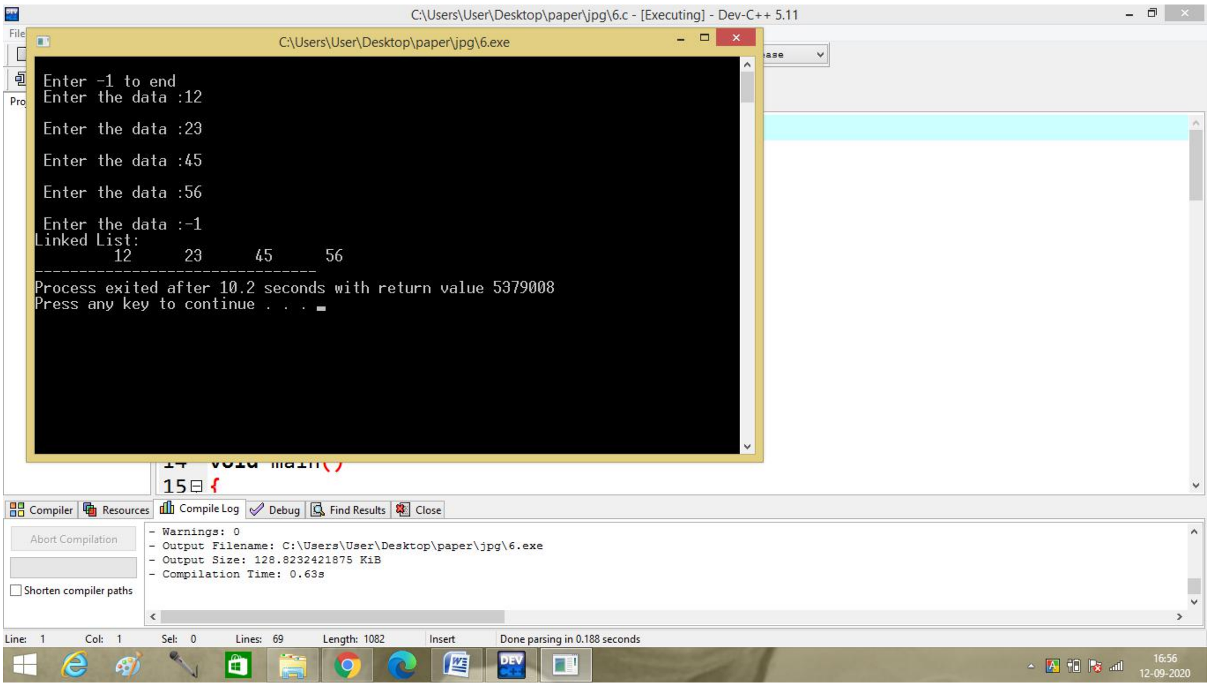
```
 Enter -1 to end
 Enter the data :12

 Enter the data :23

 Enter the data :45

 Enter the data :23

 Enter the data :47

 Enter the data :34

 Enter the data :12

 Enter the data :-1
Before Removing duplicates-
Linked List:
         12      23      45      23      47      34      12

--------------------------------
Process exited after 25.97 seconds with return value 255
Press any key to continue . . .
```

```
- Warnings: 0
- Output Filename: C:\Users\User\Desktop\paper\jpg\5.exe
- Output Size: 130.5126953125 KiB
- Compilation Time: 0.59s
```

6.WAP to convert a given singly linked list to a circular list.

```
#include<stdio.h>
#include<malloc.h>
struct node
{
  int data;
  struct node *next;
};
struct node *start=NULL;
struct node * create(struct node *);
void display(struct node *);
void main()
{
start=create(start);
display(start);
}
struct node * create(struct node *start)
{
  struct node *new_node,*ptr;
  int num;
  printf("\n Enter -1 to end");
  printf("\n Enter the data :");
  scanf("%d",&num);
  while(num != -1)
  {
   new_node=(struct node *)malloc(sizeof(struct node));
   new_node->data=num;
   if(start==NULL)
   {
     new_node->next=NULL;
     start=new_node;
   }
   else
   {
     ptr=start;
     while(ptr->next != NULL)
        ptr=ptr->next;
     ptr->next=new_node;
     new_node->next=NULL;
    }
    printf("\n Enter the data :");
    scanf("%d",&num);
   }
   new_node->next=start;

 return start;
}
void display(struct node *start)
{
  struct node *ptr;
  ptr=start;
  printf("Linked List:\n");
  do
  {
    printf("\t %d",ptr->data);
    ptr=ptr->next;
  }
while(ptr!=start);
}
```

OUTPUT :



```
 Enter -1 to end
 Enter the data :12

 Enter the data :23

 Enter the data :45

 Enter the data :56

 Enter the data :-1
Linked List:
         12       23       45       56
--------------------------------
Process exited after 10.2 seconds with return value 5379008
Press any key to continue . . .
```

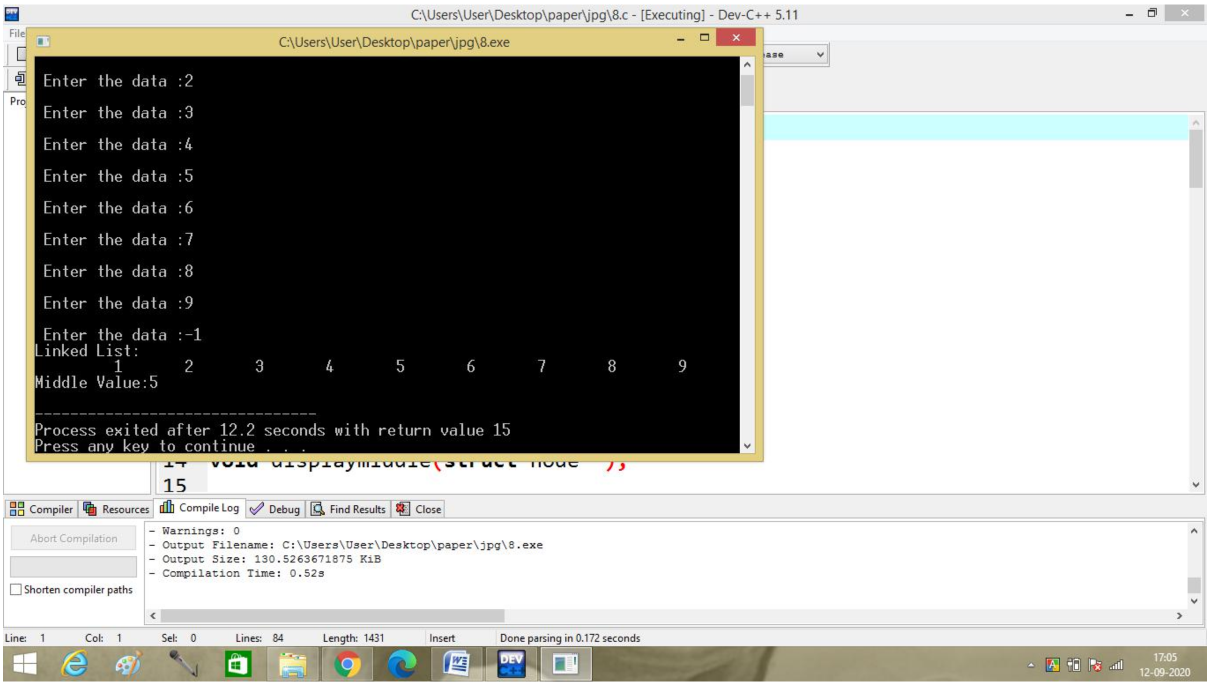
```
- Warnings: 0
- Output Filename: C:\Users\User\Desktop\paper\jpg\6.exe
- Output Size: 128.8232421875 KiB
- Compilation Time: 0.63s
```

8.WAP to print the middle of a double linked list.

```
#include<stdio.h>
#include<malloc.h>
struct node
{
  int data;
  struct node *prev;
  struct node *next;
};
struct node *start=NULL;
struct node * create(struct node *);
void display(struct node *);
void displaymiddle(struct node *);
void main()
{
start=create(start);
```

```
display(start);
printf("\n");
displaymiddle(start);
}
struct node * create(struct node *start)
{
  struct node *new_node,*ptr;
  int num;
  printf("\n Enter -1 to end");
  printf("\n Enter the data :");
  scanf("%d",&num);
  while(num != -1)
  {
   new_node=(struct node *)malloc(sizeof(struct node));
   new_node->data=num;
   if(start==NULL)
   {
     new_node->next=NULL;
     new_node->prev=NULL;
     start=new_node;
   }
   else
   {
     ptr=start;
     while(ptr->next != NULL)
        ptr=ptr->next;
     ptr->next=new_node;
     new_node->prev=ptr;
     new_node->next=NULL;
    }
    printf("\n Enter the data :");
    scanf("%d",&num);
   }
 return start;
}
void display(struct node *start)
{
  struct node *ptr;
  ptr=start;
  printf("Linked List:\n");
  while(ptr != NULL)
  {
    printf("\t %d",ptr->data);
    ptr=ptr->next;
  }
}
void displaymiddle(struct node *start)
{
struct node *fast,*slow;
fast=start;
slow=start;
while(fast!=NULL&&fast->next!=NULL)
{
    fast=fast->next->next;
    slow=slow->next;
}
printf("Middle Value:%d\n",slow->data);
}
```

OUTPUT :



C:\Users\User\Desktop\paper\jpg\8.c - [Executing] - Dev-C++ 5.11
C:\Users\User\Desktop\paper\jpg\8.exe
Compiler
Resources
Compile Log
Debug
Find Results
Close
Abort Compilation
Shorten compiler paths
Line: 1 Col: 1 Sel: 0 Lines: 84 Length: 1431 Insert Done parsing in 0.172 seconds
17:05
12-09-2020

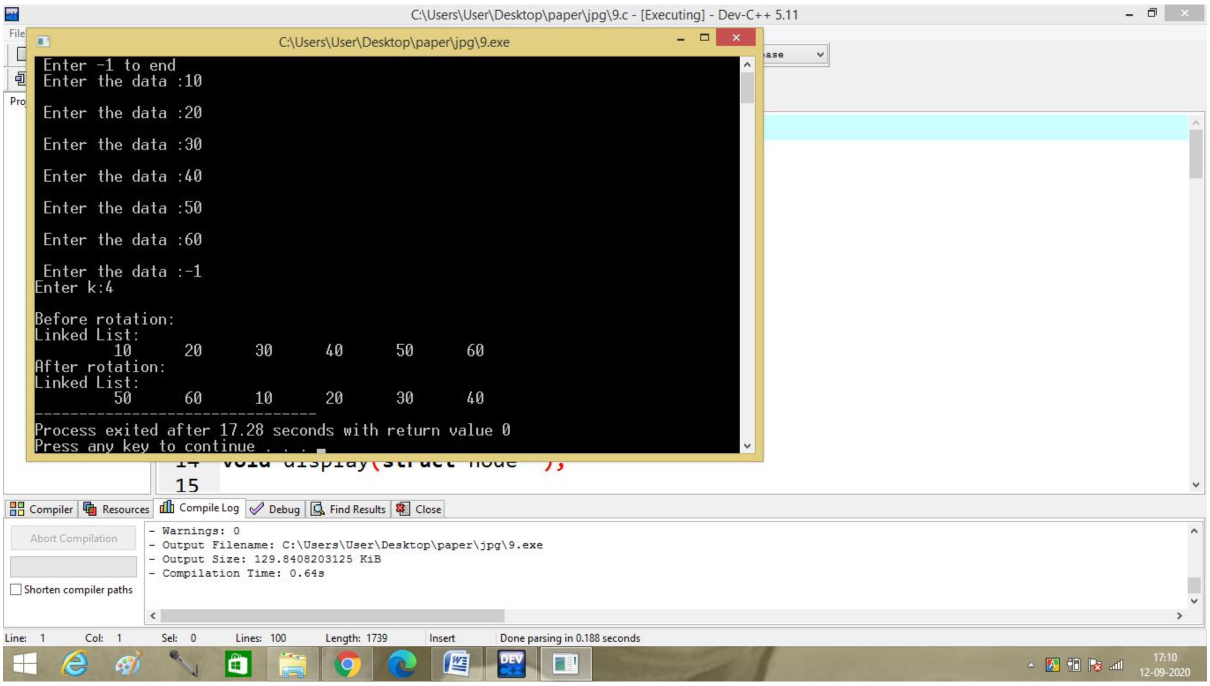
```
 Enter the data :2

 Enter the data :3

 Enter the data :4

 Enter the data :5

 Enter the data :6

 Enter the data :7

 Enter the data :8

 Enter the data :9

 Enter the data :-1
Linked List:
          1       2       3       4       5       6       7       8       9
Middle Value:5

--------------------------------
Process exited after 12.2 seconds with return value 15
Press any key to continue . . .
```

```
- Warnings: 0
- Output Filename: C:\Users\User\Desktop\paper\jpg\8.exe
- Output Size: 130.5263671875 KiB
- Compilation Time: 0.52s
```

Given a double linked list, rotate the linked list counter-clockwise by k nodes. Where k is a given positive integer. For example, if the given linked list is 10->20->30->40->50->60 and k is 4, the list should be modified to 50->60->10->20->30->40. Assume that k is smaller than the count of nodes in linked list.

```
#include<stdio.h>

#include<malloc.h>


struct node

{

  int data;

  struct node *prev;

  struct node *next;

};

struct node *start=NULL;


struct node * create(struct node *);

struct node * rotate(struct node *,int);

void display(struct node *);
void main()
{int k;
start=create(start);
printf("Enter k:");
scanf("%d",&k);
printf("\nBefore rotation:\n");
display(start);
start=rotate(start,k);
printf("\nAfter rotation:\n");
display(start);
}
struct node * create(struct node *start)
{

  struct node *new_node,*ptr;
  int num;
  printf("\n Enter -1 to end");
  printf("\n Enter the data :");
  scanf("%d",&num);

  while(num != -1)
  {

   new_node=(struct node *)malloc(sizeof(struct node));
   new_node->data=num;

   if(start==NULL)
   {
     new_node->next=NULL;
     new_node->prev=NULL;
     start=new_node;
   }
   else
   {

     ptr=start;

     while(ptr->next != NULL)
        ptr=ptr->next;

     ptr->next=new_node;
     new_node->prev=ptr;
     new_node->next=NULL;
    }
    printf("\n Enter the data :");
    scanf("%d",&num);
   }
 return start;
}
struct node * rotate(struct node* start,int k)
{
    struct node *ptr,*p;
    int count=0;
    ptr=start;
    p=start;
    while(p->next!=NULL)
    {p=p->next;
    }
    while(ptr!=NULL)
    {
        count++;
        if(count==k)
        {
            p->next=start;
            start=ptr->next;
            ptr->next=NULL;

        }
        else
        ptr=ptr->next;
    }
    return start;
}
void display(struct node *start)
{
  struct node *ptr;
  ptr=start;
  printf("Linked List:\n");
  while(ptr != NULL)
  {

    printf("\t %d",ptr->data);
    ptr=ptr->next;
  }
}
```

OUTPUT :



```
 Enter -1 to end
 Enter the data :10

 Enter the data :20

 Enter the data :30

 Enter the data :40

 Enter the data :50

 Enter the data :60

 Enter the data :-1
Enter k:4

Before rotation:
Linked List:
         10      20      30      40      50      60
After rotation:
Linked List:
         50      60      10      20      30      40
--------------------------------
Process exited after 17.28 seconds with return value 0
Press any key to continue . . .
```

```
- Warnings: 0
- Output Filename: C:\Users\User\Desktop\paper\jpg\9.exe
- Output Size: 129.8408203125 KiB
- Compilation Time: 0.64s
```

```
Line: 1    Col: 1    Sel: 0    Lines: 100    Length: 1739    Insert    Done parsing in 0.188 seconds
```