# DATA STRUTURE AND ALGORITHM LAB

## ASSIGNMENT 2

## **DATE OF ASSIGNMENT:- 18.07.18**

## **DATE OF SUBMISSION:- 25.07.18**

**2.1 Write a C program to Create a circular Linked list.**

#include<stdio.h>

#include<stdlib.h>

typedef struct Node{

int data;

struct Node \*next;

}node;

node \*head;

void create();

void display();

int main()

{

int ch;

head=NULL;

while(1)

{

printf("\n1.Create\n2.Display\nExit\nEnter your choice :- ");

scanf("%d",&ch);

switch(ch)

{

case 1:

create();

break;

case 2:

display();

break;

case 3:

exit(0);

break;

default:

printf("Invalid Input!!!!");

}

}

}

void create()

{

node \*nw,\*ptr;

nw = (node \*)malloc(sizeof(node));

printf("Enter a number = ");

scanf("%d",&nw->data);

nw->next = NULL;

if(head==NULL)

{

head=nw;

nw->next=head;

}

else

{

ptr = head;

while(ptr->next!=head)

ptr=ptr->next;

ptr->next = nw;

nw->next=head;

}

}

void display()

{

node \*ptr;

if(head==NULL)

printf("Linked list does not exists!!!");

else

{

ptr=head;

do

{

printf("%d\t",ptr->data);

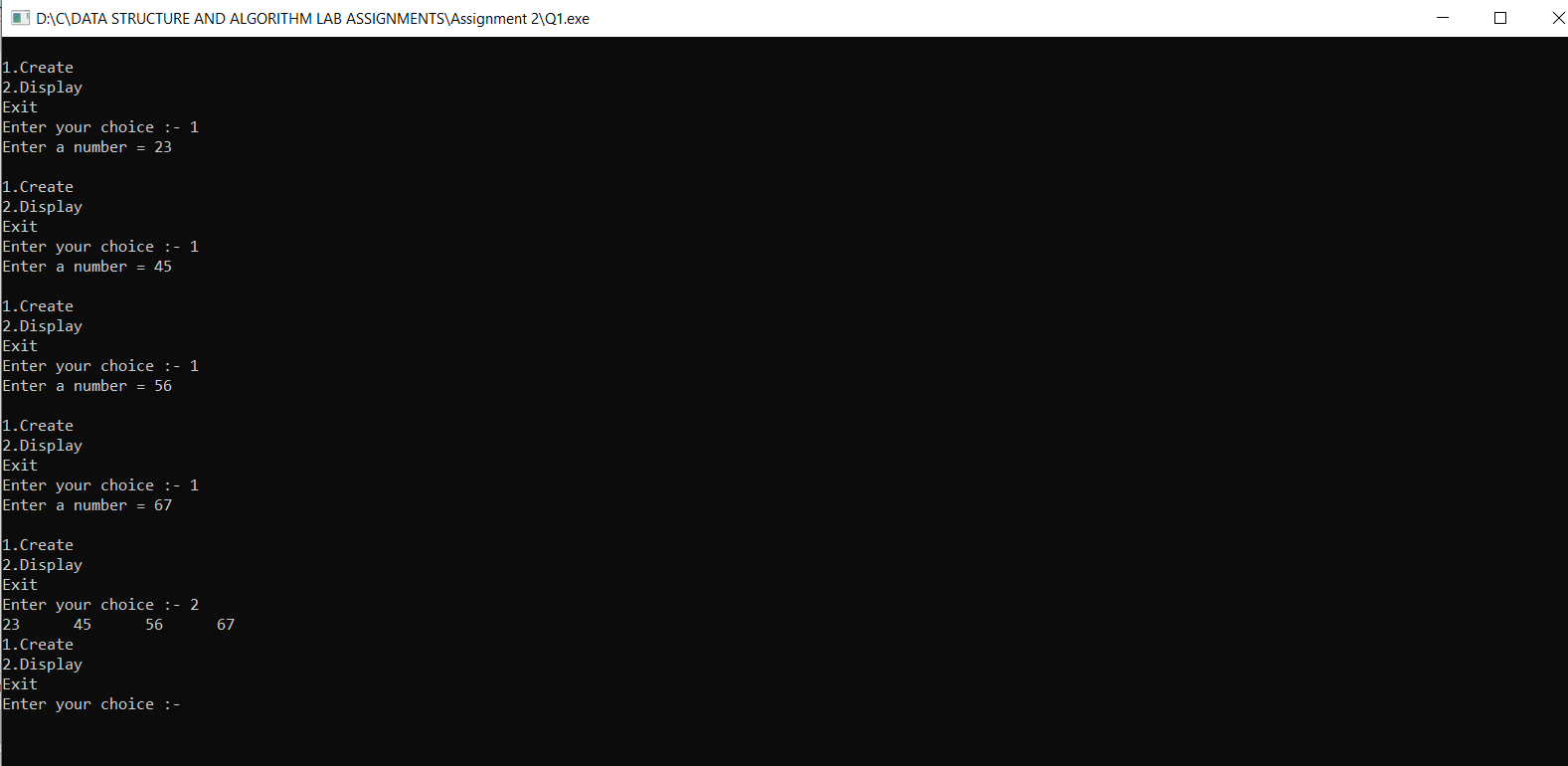
ptr = ptr->next;

}while(ptr!=head);

}

}

**OUTPUT:-**



**2.2 Write a C program to display the elements of a circular linked list.**

#include<stdio.h>

#include<stdlib.h>

typedef struct Node{

int data;

struct Node \*next;

}node;

node \*head;

void create();

void display();

int main()

{

int ch;

head=NULL;

while(1)

{

printf("\n1.Create\n2.Display\nExit\nEnter your choice :- ");

scanf("%d",&ch);

switch(ch)

{

case 1:

create();

break;

case 2:

display();

break;

case 3:

exit(0);

break;

default:

printf("Invalid Input!!!!");

}

}

}

void create()

{

node \*nw,\*ptr;

nw = (node \*)malloc(sizeof(node));

printf("Enter a number = ");

scanf("%d",&nw->data);

nw->next = NULL;

if(head==NULL)

{

head=nw;

nw->next=head;

}

else

{

ptr = head;

while(ptr->next!=head)

ptr=ptr->next;

ptr->next = nw;

nw->next=head;

}

}

void display()

{

node \*ptr;

if(head==NULL)

printf("Linked list does not exists!!!");

else

{

ptr=head;

do

{

printf("%d\t",ptr->data);

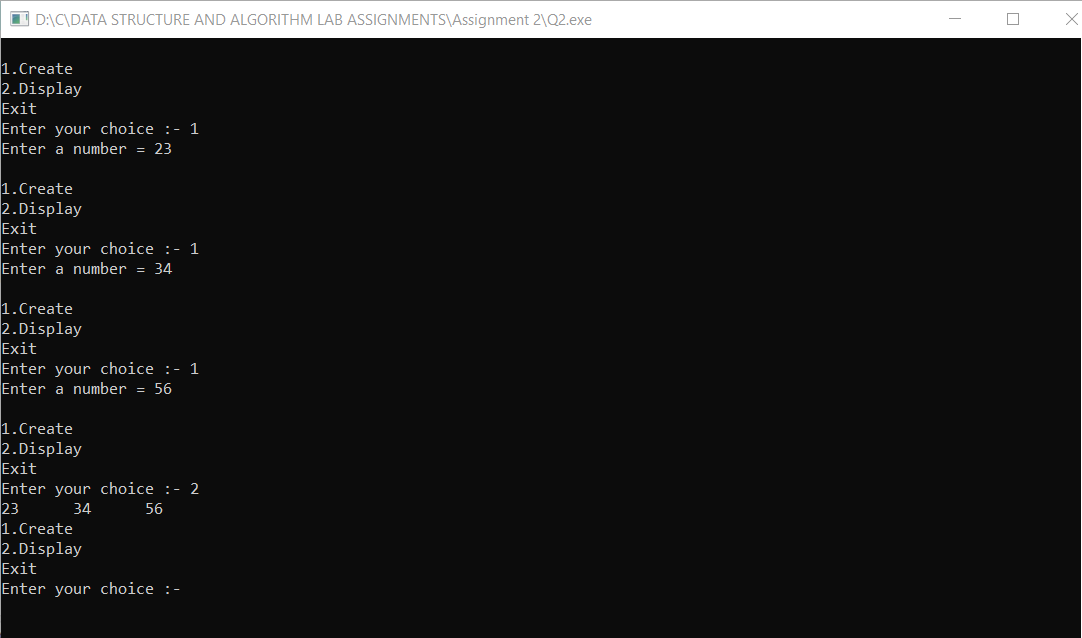
ptr = ptr->next;

}while(ptr!=head);

}

}

**OUTPUT:-**



**2.3 Write a C program to search an user given element in a Circular linked list.**

#include<stdio.h>

#include<stdlib.h>

typedef struct Node{

int data;

struct Node \*next;

}node;

node \*head;

void create();

void display();

void search();

int main()

{

int ch;

head=NULL;

while(1)

{

printf("\n1.Create\n2.Display\n3.Search an element\n4.Exit\nEnter your choice :- ");

scanf("%d",&ch);

switch(ch)

{

case 1:

create();

break;

case 2:

display();

break;

case 3:

search();

break;

case 4:

exit(0);

break;

default:

printf("Invalid Input!!!!");

}

}

}

void create()

{

node \*nw,\*ptr;

nw = (node \*)malloc(sizeof(node));

printf("Enter a number = ");

scanf("%d",&nw->data);

nw->next = NULL;

if(head==NULL)

{

head=nw;

nw->next=head;

}

else

{

ptr = head;

while(ptr->next!=head)

ptr=ptr->next;

ptr->next = nw;

nw->next=head;

}

}

void display()

{

node \*ptr;

if(head==NULL)

printf("Linked list does not exists!!!");

else

{

ptr=head;

do

{

printf("%d\t",ptr->data);

ptr = ptr->next;

}while(ptr!=head);

}

}

void search()

{

node \*ptr;

int item,flag=1;

if(head==NULL)

printf("Linked list does not exists!!!!");

else

{

ptr = head;

printf("Enter the value you want to search = ");

scanf("%d",&item);

do

{

if(ptr->data==item)

{

flag=0;

break;

}

ptr = ptr->next;

}while(ptr!=head);

}

if(flag==0)

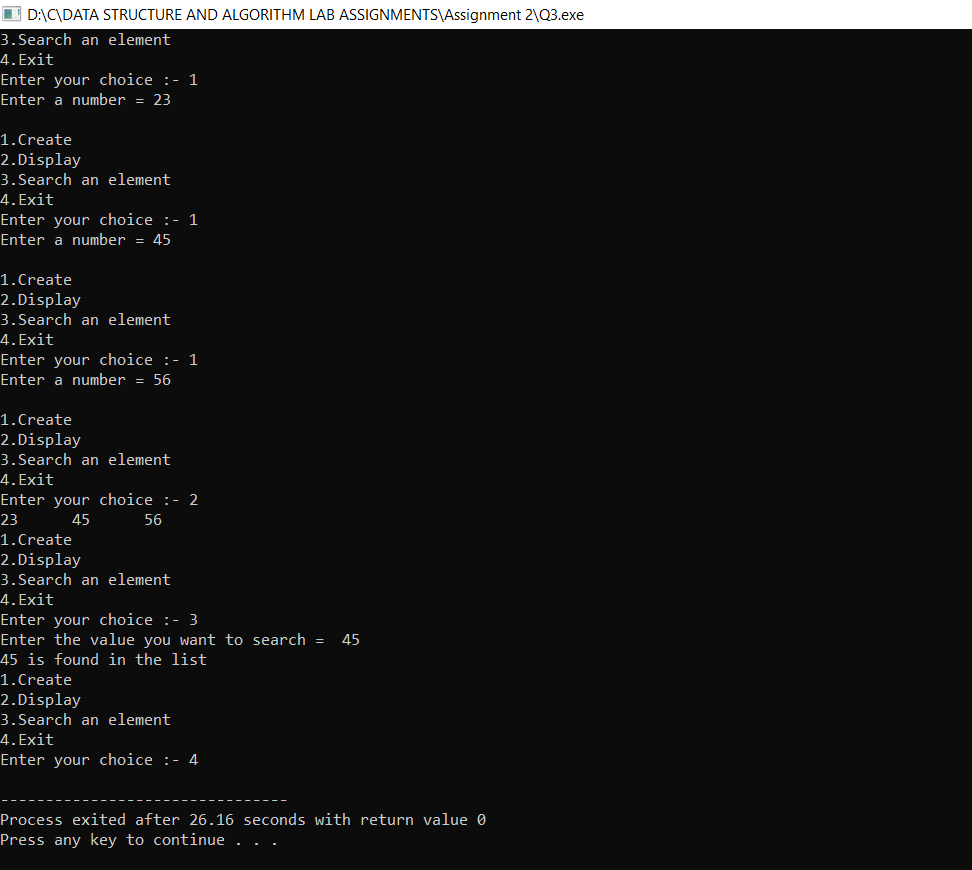
printf("%d is found in the list",item);

else

printf("%d is not found in the list",item);

}

**OUTPUT:-**

****

**2.4 Write a C program to add an element at beginning of a Circular Linked List.**

#include<stdio.h>

#include<stdlib.h>

typedef struct Node{

int data;

struct Node \*next;

}node;

node \*head;

void create();

void display();

void in\_beg();

int main()

{

int ch;

head=NULL;

while(1)

{

printf("\n1.Create\n2.Display\n3.Insert at beginnig\n4.Exit\nEnter your choice :- ");

scanf("%d",&ch);

switch(ch)

{

case 1:

create();

break;

case 2:

display();

break;

case 3:

in\_beg();

break;

case 4:

exit(0);

break;

default:

printf("Invalid Input!!!!");

}

}

}

void create()

{

node \*nw,\*ptr;

nw = (node \*)malloc(sizeof(node));

printf("Enter a number = ");

scanf("%d",&nw->data);

nw->next = NULL;

if(head==NULL)

{

head=nw;

nw->next=head;

}

else

{

ptr = head;

while(ptr->next!=head)

ptr=ptr->next;

ptr->next = nw;

nw->next=head;

}

}

void display()

{

node \*ptr;

if(head==NULL)

printf("Linked list does not exists!!!");

else

{

ptr=head;

do

{

printf("%d\t",ptr->data);

ptr = ptr->next;

}while(ptr!=head);

}

}

void in\_beg()

{

node \*nw, \*ptr,\*ptr1;

nw = (node \*)malloc(sizeof(node));

printf("Enter a number = ");

scanf("%d",&nw->data);

nw->next=NULL;

if(head==NULL)

printf("Linked List does not exists!!!");

else

{

ptr=head;

while(ptr->next!=head)

ptr=ptr->next;

nw->next=head;

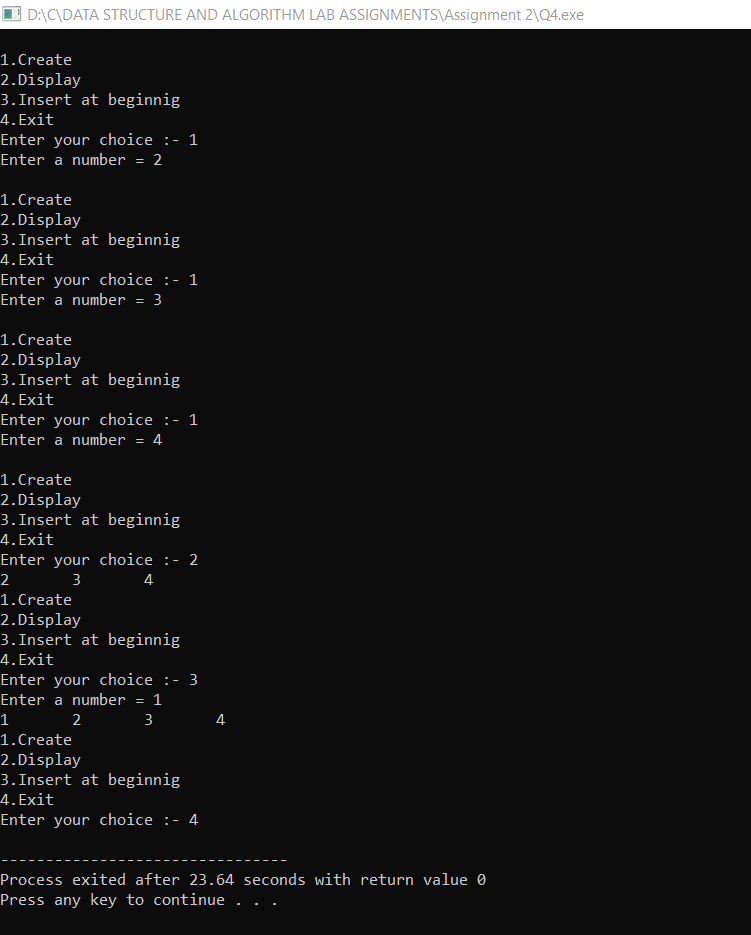
head=nw;

ptr->next=head;

}

}

**OUTPUT:-**



**2.5 Write a C program to add an element at the end of a Circular Linked list.**

#include<stdio.h>

#include<stdlib.h>

typedef struct Node{

int data;

struct Node \*next;

}node;

node \*head;

void create();

void display();

void in\_last();

int main()

{

int ch;

head=NULL;

while(1)

{

printf("\n1.Create\n2.Display\n3.Insert at end\n4.Exit\nEnter your choice :- ");

scanf("%d",&ch);

switch(ch)

{

case 1:

create();

break;

case 2:

display();

break;

case 3:

in\_last();

break;

case 4:

exit(0);

break;

default:

printf("Invalid Input!!!!");

}

}

}

void create()

{

node \*nw,\*ptr;

nw = (node \*)malloc(sizeof(node));

printf("Enter a number = ");

scanf("%d",&nw->data);

nw->next = NULL;

if(head==NULL)

{

head=nw;

nw->next=head;

}

else

{

ptr = head;

while(ptr->next!=head)

ptr=ptr->next;

ptr->next = nw;

nw->next=head;

}

}

void display()

{

node \*ptr;

if(head==NULL)

printf("Linked list does not exists!!!");

else

{

ptr=head;

do

{

printf("%d\t",ptr->data);

ptr = ptr->next;

}while(ptr!=head);

}

}

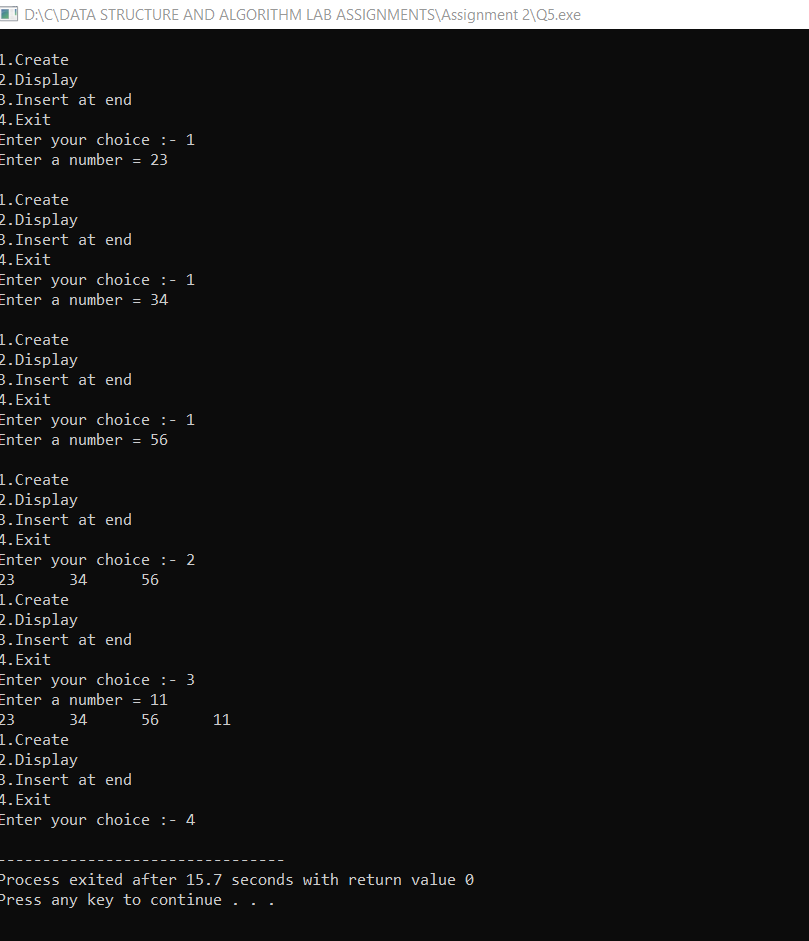
void in\_last()

{

create();

}

**OUTPUT:-**

****

**2.6 Write a C program to add an element at any user given position of a Circular Linked List.**

#include<stdio.h>

#include<stdlib.h>

typedef struct Node{

int data;

struct Node \*next;

}node;

node \*head;

void create();

void display();

void in\_middle();

int main()

{

int ch;

head=NULL;

while(1)

{

printf("\n1.Create\n2.Display\n3.Insert at specific position\n4.Exit\nEnter your choice :- ");

scanf("%d",&ch);

switch(ch)

{

case 1:

create();

break;

case 2:

display();

break;

case 3:

in\_middle();

break;

case 4:

exit(0);

break;

default:

printf("Invalid Input!!!!");

}

}

}

void create()

{

node \*nw,\*ptr;

nw = (node \*)malloc(sizeof(node));

printf("Enter a number = ");

scanf("%d",&nw->data);

nw->next = NULL;

if(head==NULL)

{

head=nw;

nw->next=head;

}

else

{

ptr = head;

while(ptr->next!=head)

ptr=ptr->next;

ptr->next = nw;

nw->next=head;

}

}

void display()

{

node \*ptr;

if(head==NULL)

printf("Linked list does not exists!!!");

else

{

ptr=head;

do

{

printf("%d\t",ptr->data);

ptr = ptr->next;

}while(ptr!=head);

}

}

void in\_middle()

{

node \*nw, \*ptr, \*ptr1;

int pos,i=1;

printf("Enter the position you want to insert = ");

scanf("%d",&pos);

if(head==NULL)

printf("Linked List does not exists!!!");

else

{

nw = (node \*)malloc(sizeof(node));

printf("Enter a number = ");

scanf("%d",&nw->data);

nw->next=NULL;

ptr=head;

ptr1=head;

while(ptr->next!=head&&i!=pos)

{

ptr1=ptr;

ptr = ptr->next;

i++;

}

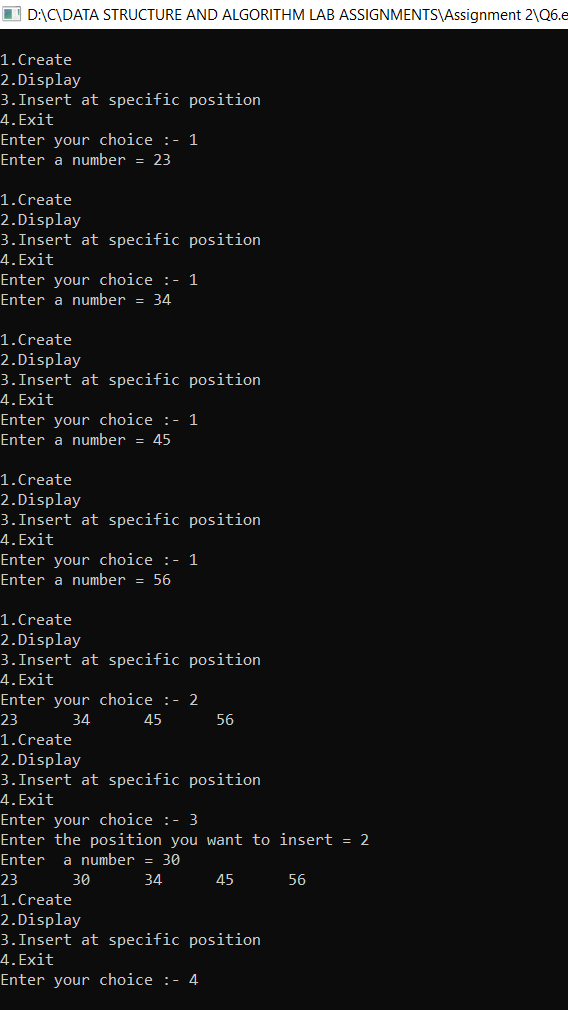
nw->next=ptr;

ptr1->next=nw;

}

}

**OUTPUT:-**

****

**2.7 Write a C program to count the number of nodes in a Circular Linked List.**

#include<stdio.h>

#include<stdlib.h>

typedef struct Node{

int data;

struct Node \*next;

}node;

node \*head;

void create();

void display();

void count\_nodes();

int main()

{

int ch;

head=NULL;

while(1)

{

printf("\n1.Create\n2.Display\n3.Count the number of nodes\n4.Exit\nEnter your choice :- ");

scanf("%d",&ch);

switch(ch)

{

case 1:

create();

break;

case 2:

display();

break;

case 3:

count\_nodes();

break;

case 4:

exit(0);

break;

default:

printf("Invalid Input!!!!");

}

}

}

void create()

{

node \*nw,\*ptr;

nw = (node \*)malloc(sizeof(node));

printf("Enter a number = ");

scanf("%d",&nw->data);

nw->next = NULL;

if(head==NULL)

{

head=nw;

nw->next=head;

}

else

{

ptr = head;

while(ptr->next!=head)

ptr=ptr->next;

ptr->next = nw;

nw->next=head;

}

}

void display()

{

node \*ptr;

if(head==NULL)

printf("Linked list does not exists!!!");

else

{

ptr=head;

do

{

printf("%d\t",ptr->data);

ptr = ptr->next;

}while(ptr!=head);

}

}

void count\_nodes()

{

node \*ptr;

int c=0;

if(head==NULL)

printf("Linked List does not exists!!!");

else

{

ptr=head;

do

{

ptr = ptr->next;

c++;

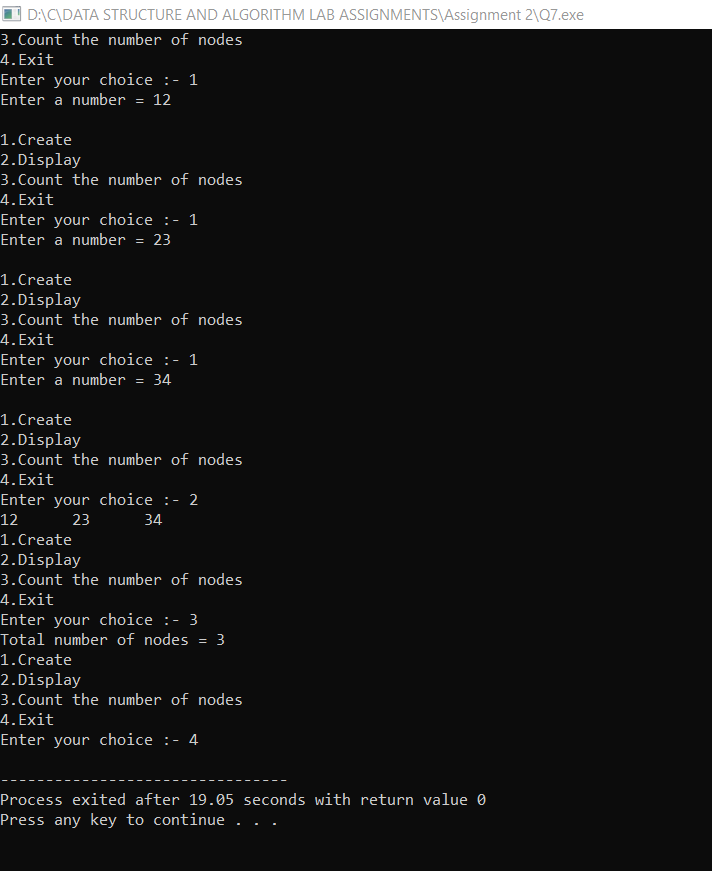
}while(ptr!=head);

}

printf("Total number of nodes = %d",c);

}

**OUTPUT:-**

****

**2.8 Write a C program to delete a node at beginning of a Circular Linked List.**

#include<stdio.h>

#include<stdlib.h>

typedef struct Node{

int data;

struct Node \*next;

}node;

node \*head;

void create();

void display();

void del\_beg();

int main()

{

int ch;

head=NULL;

while(1)

{

printf("\n1.Create\n2.Display\n3.Delete from beginning\n4.Exit\nEnter your choice :- ");

scanf("%d",&ch);

switch(ch)

{

case 1:

create();

break;

case 2:

display();

break;

case 3:

del\_beg();

break;

case 4:

exit(0);

break;

default:

printf("Invalid Input!!!!");

}

}

}

void create()

{

node \*nw,\*ptr;

nw = (node \*)malloc(sizeof(node));

printf("Enter a number = ");

scanf("%d",&nw->data);

nw->next = NULL;

if(head==NULL)

{

head=nw;

nw->next=head;

}

else

{

ptr = head;

while(ptr->next!=head)

ptr=ptr->next;

ptr->next = nw;

nw->next=head;

}

}

void display()

{

node \*ptr;

if(head==NULL)

printf("Linked list does not exists!!!");

else

{

ptr=head;

do

{

printf("%d\t",ptr->data);

ptr = ptr->next;

}while(ptr!=head);

}

}

void del\_beg()

{

node \*ptr,\*ptr1;

ptr=head;

ptr1=head;

do

{

ptr=ptr->next;

}while(ptr->next!=head);

printf("Delete item is = %d",ptr1->data);

head = ptr1->next;

free(ptr1);

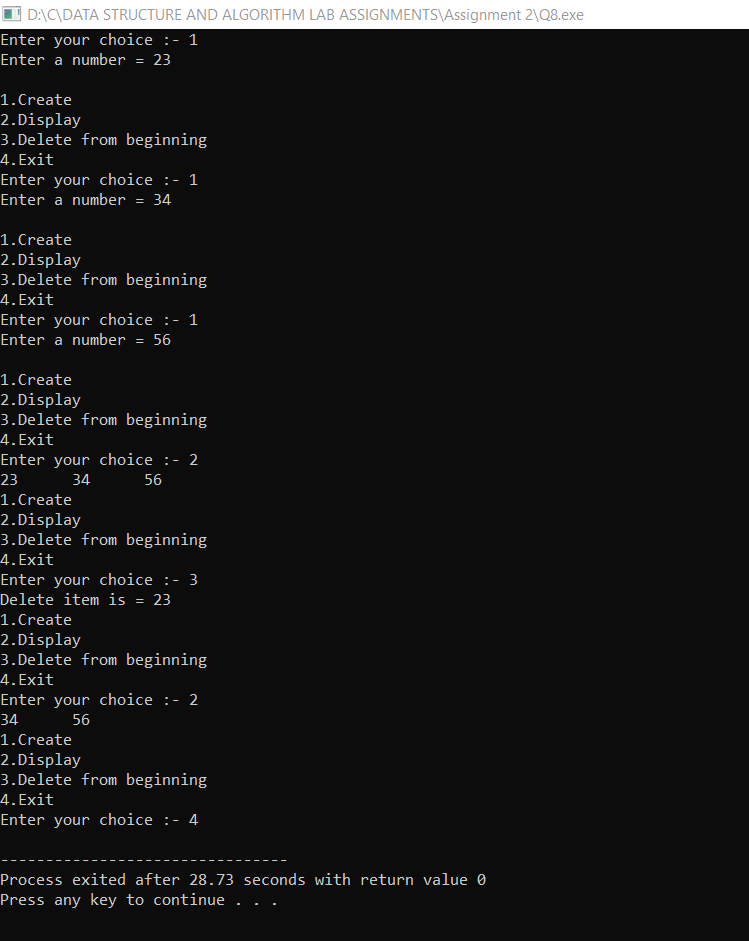
ptr->next=head;

if(ptr1==ptr)

head=NULL;

}

**OUTPUT:-**

****

**2.9 Write a C program to delete a node at end of a Circular Linked list.**

#include<stdio.h>

#include<stdlib.h>

typedef struct Node{

int data;

struct Node \*next;

}node;

node \*head;

void create();

void display();

void del\_end();

int main()

{

int ch;

head=NULL;

while(1)

{

printf("\n1.Create\n2.Display\n3.Delete from end\n4.Exit\nEnter your choice :- ");

scanf("%d",&ch);

switch(ch)

{

case 1:

create();

break;

case 2:

display();

break;

case 3:

del\_end();

break;

case 4:

exit(0);

break;

default:

printf("Invalid Input!!!!");

}

}

}

void create()

{

node \*nw,\*ptr;

nw = (node \*)malloc(sizeof(node));

printf("Enter a number = ");

scanf("%d",&nw->data);

nw->next = NULL;

if(head==NULL)

{

head=nw;

nw->next=head;

}

else

{

ptr = head;

while(ptr->next!=head)

ptr=ptr->next;

ptr->next = nw;

nw->next=head;

}

}

void display()

{

node \*ptr;

if(head==NULL)

printf("Linked list does not exists!!!");

else

{

ptr=head;

do

{

printf("%d\t",ptr->data);

ptr = ptr->next;

}while(ptr!=head);

}

}

void del\_end()

{

node \*ptr1, \*ptr;

ptr=head;

ptr1=head;

while(ptr->next!=head)

{

ptr1=ptr;

ptr = ptr->next;

}

printf("Delete item is = %d",ptr->data);

ptr1->next=head;

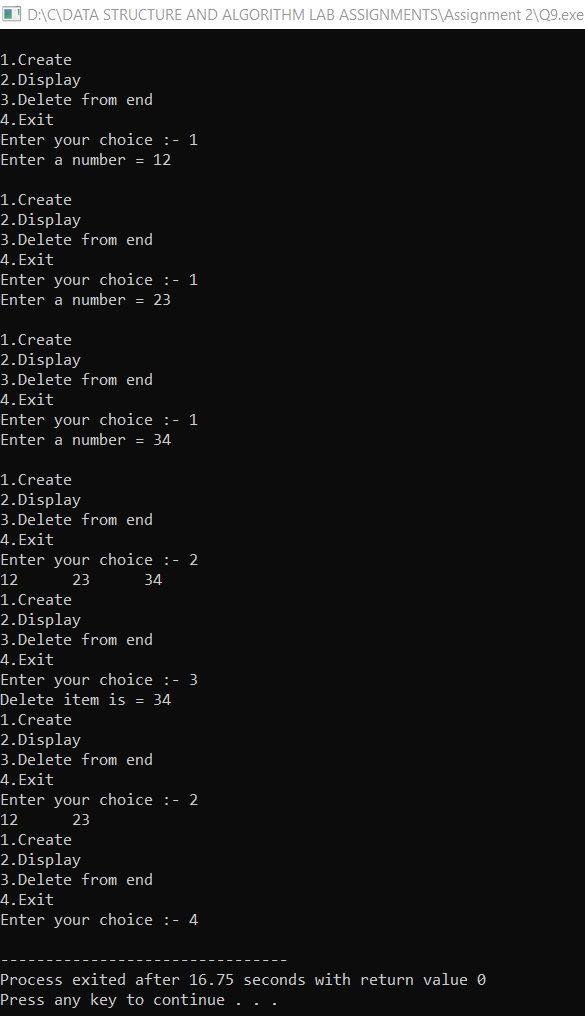
free(ptr);

if(ptr1==ptr)

head=NULL;

}

**OUTPUT:-**

****

**2.10 Write a C program to reverse a Circular Linked List.**

#include<stdio.h>

#include<stdlib.h>

typedef struct Node{

int data;

struct Node \*next;

}node;

node \*head;

void create();

void display();

void reverse();

int main()

{

int ch;

head=NULL;

while(1)

{

printf("\n1.Create\n2.Display\n3.Reverse\n4.Exit\nEnter your choice :- ");

scanf("%d",&ch);

switch(ch)

{

case 1:

create();

break;

case 2:

display();

break;

case 3:

reverse();

break;

case 4:

exit(0);

break;

default:

printf("Invalid Input!!!!");

}

}

}

void create()

{

node \*nw,\*ptr;

nw = (node \*)malloc(sizeof(node));

printf("Enter a number = ");

scanf("%d",&nw->data);

nw->next = NULL;

if(head==NULL)

{

head=nw;

nw->next=head;

}

else

{

ptr = head;

while(ptr->next!=head)

ptr=ptr->next;

ptr->next = nw;

nw->next=head;

}

}

void display()

{

node \*ptr;

if(head==NULL)

printf("Linked list does not exists!!!");

else

{

ptr=head;

do

{

printf("%d\t",ptr->data);

ptr = ptr->next;

}while(ptr!=head);

}

}

void reverse()

{

node \*rev\_head,\*ptr,\*ptr1;

if(head==NULL)

printf("Linked list does not exists!!!!");

else

{

printf("Original List :- \n");

display();

rev\_head=NULL;

ptr=head;

do

{

ptr1=ptr->next;

ptr->next=rev\_head;

rev\_head=ptr;

ptr=ptr1;

}while(ptr!=head);

head->next=rev\_head;

head=rev\_head;

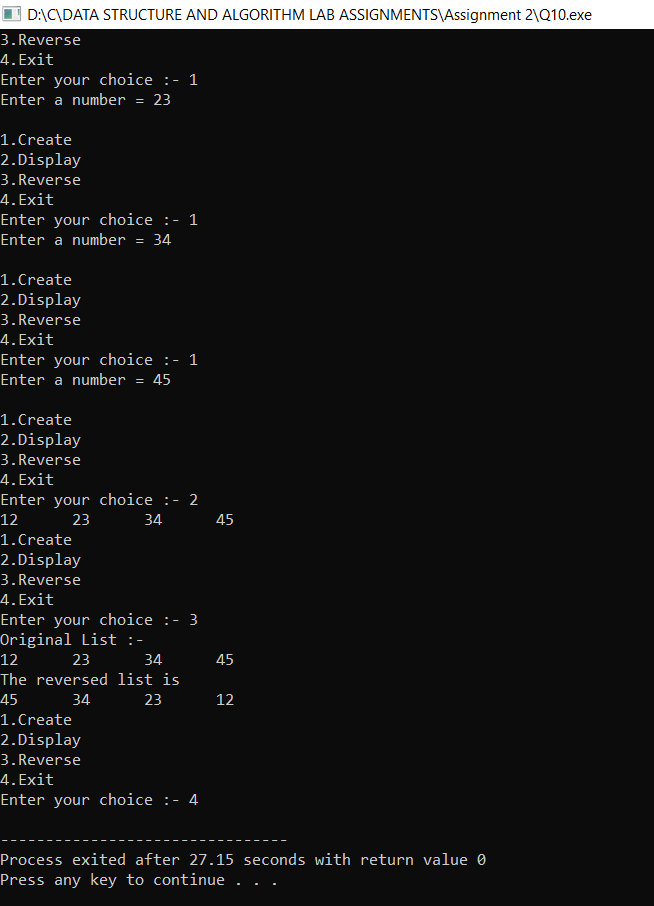
printf("\nThe reversed list is\n");

display();

}

}

**OUTPUT:-**

****