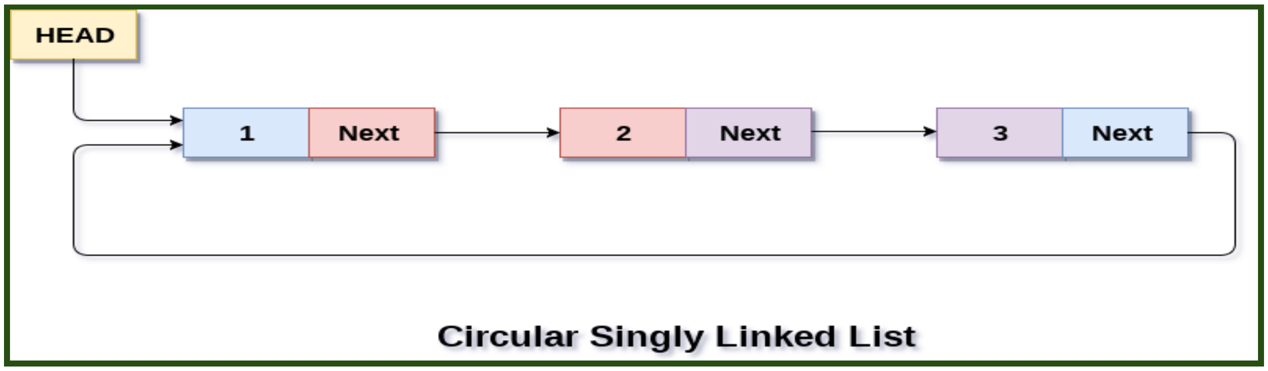
Circular Singly Linked List

In a **circular Singly linked list**, **the last node of the list contains a pointer to the first node of the list.** We can have **circular singly linked list** as well as **circular doubly linked list.**

We traverse a circular singly linked list until we reach the same node where we started. **The circular singly liked list has no beginning and no ending. There is no null value present in the next part of any of the nodes.**

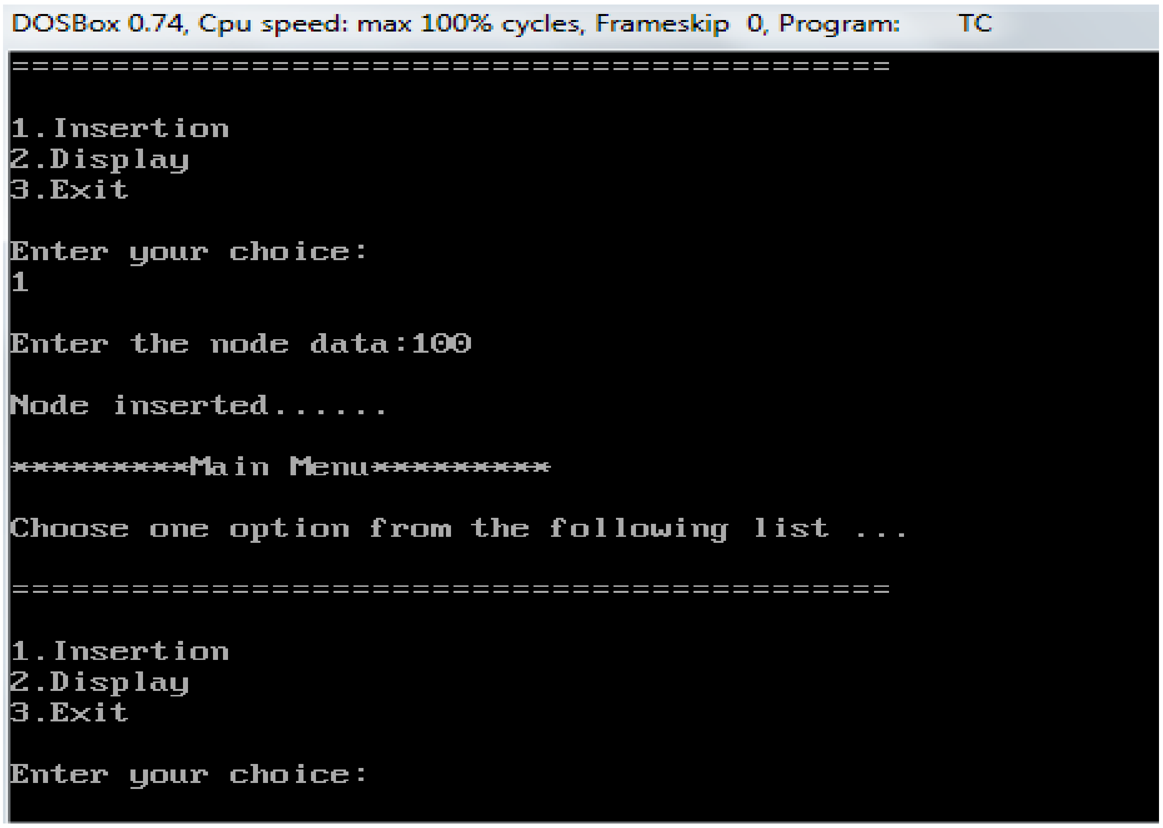
The following image shows a circular singly linked list.



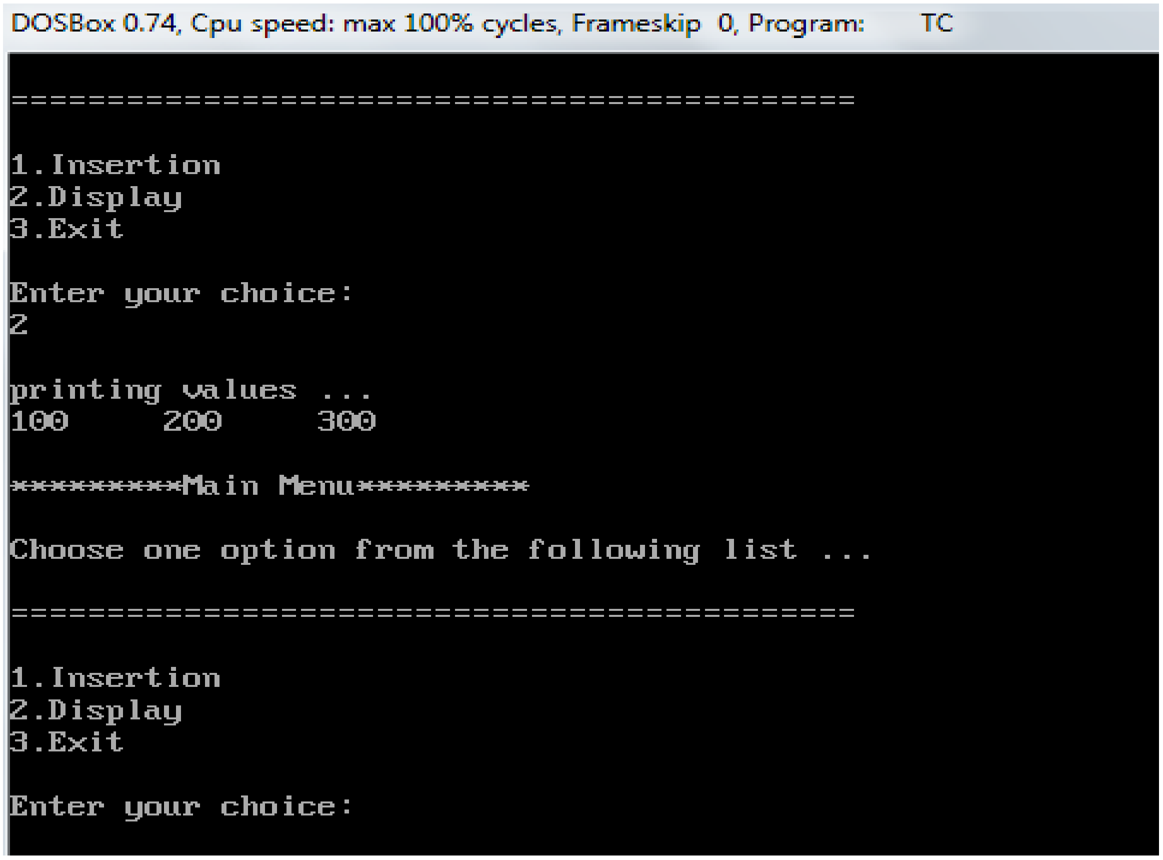
Example

```
#include<stdio.h>
#include<stdlib.h>
struct node
{
   int data;
   struct node *next;
}*head=NULL;
struct node *newnode,*temp;
void create();
void display();
void main ()
{
   int choice;
   clrscr();
   while(1)
   {
      printf("\n*********Main Menu*********\n");
      printf("\nChoose one option from the following list ...\n");
      printf("\n======================================\n");
      printf("\n1.Insertion \n2.Display \n3.Exit\n");
      printf("\nEnter your choice:\n");
      scanf("\n%d",&choice);
      switch(choice)
      {
      case 1:
         create();
         break;
      case 2:
         display();
         break;
      case 3:
         exit(0);
         break;
      default:
         printf("\nPlease enter valid choice..");
      }
   }
}
void create()
{
   newnode = (struct node *)malloc(sizeof(struct node));
   printf("\nEnter the node data:");
   scanf("%d",&newnode->data);
```

```
    newnode->next=0;
    if(head==NULL)
    {
        head=temp=newnode;
    }
    else
    {
        temp->next=newnode;
        temp=newnode;
    }
    temp->next=head;
    printf("\nNode inserted......\n");
}
void display()
{
    struct node *temp;
    temp=head;
    if(head==NULL)
    {
        printf("\nNothing to print");
    }
    else
    {
        printf("\nprinting values ... \n");

        while(temp->next!=head)
        {
            printf("%d\t",temp->data);
            temp=temp->next;
        }
        printf("%d\n",temp->data);
    }
}
```


DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC

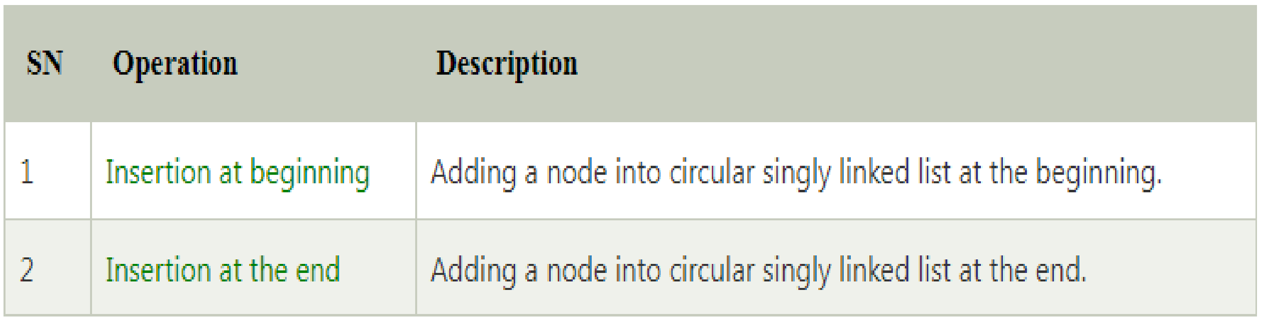
```
============================================

1.Insertion
2.Display
3.Exit

Enter your choice:
1

Enter the node data:100

Node inserted......

**********Main Menu**********

Choose one option from the following list ...

============================================

1.Insertion
2.Display
3.Exit

Enter your choice:
```


DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC

```
============================================

1.Insertion
2.Display
3.Exit

Enter your choice:
2

printing values ...
100     200     300

**********Main Menu**********

Choose one option from the following list ...

============================================

1.Insertion
2.Display
3.Exit

Enter your choice:
```

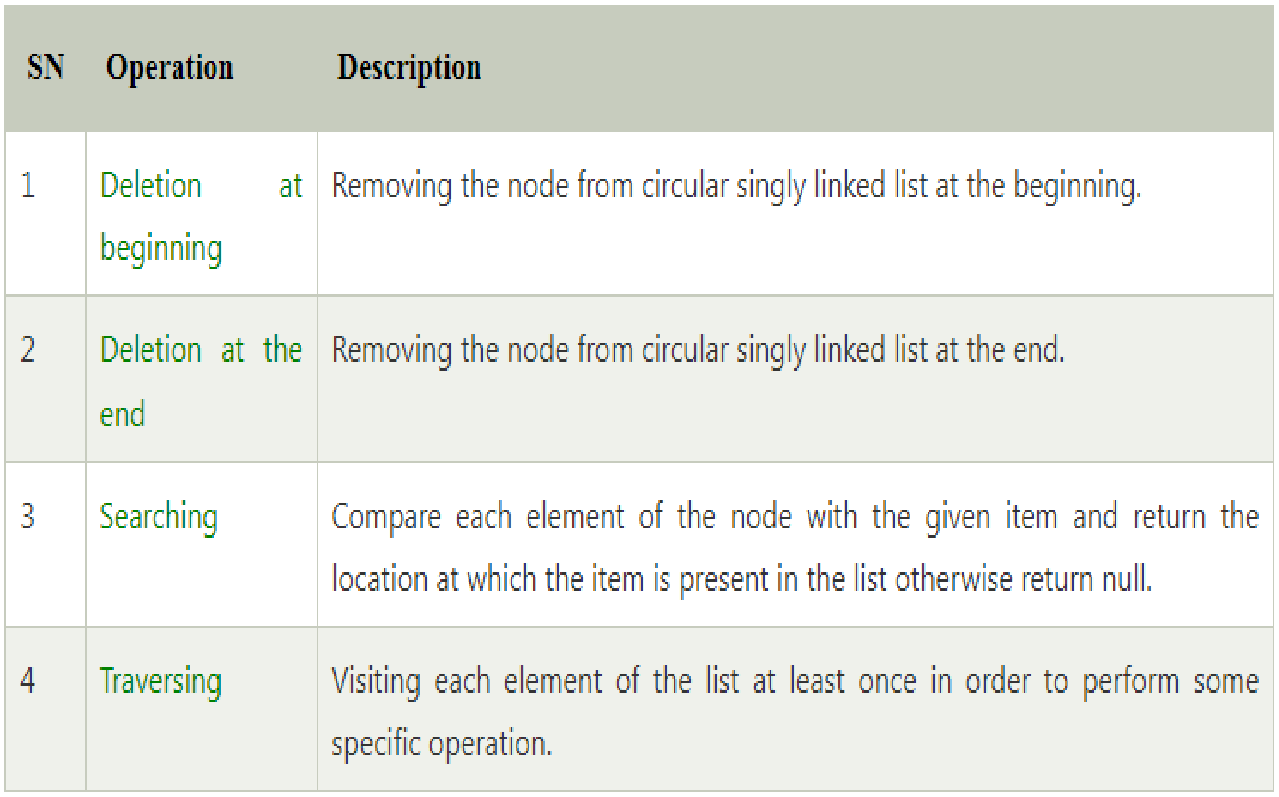
Operations on Circular Singly linked list:

Insertion



SN	Operation	Description
1	Insertion at beginning	Adding a node into circular singly linked list at the beginning.
2	Insertion at the end	Adding a node into circular singly linked list at the end.

Deletion & Traversing

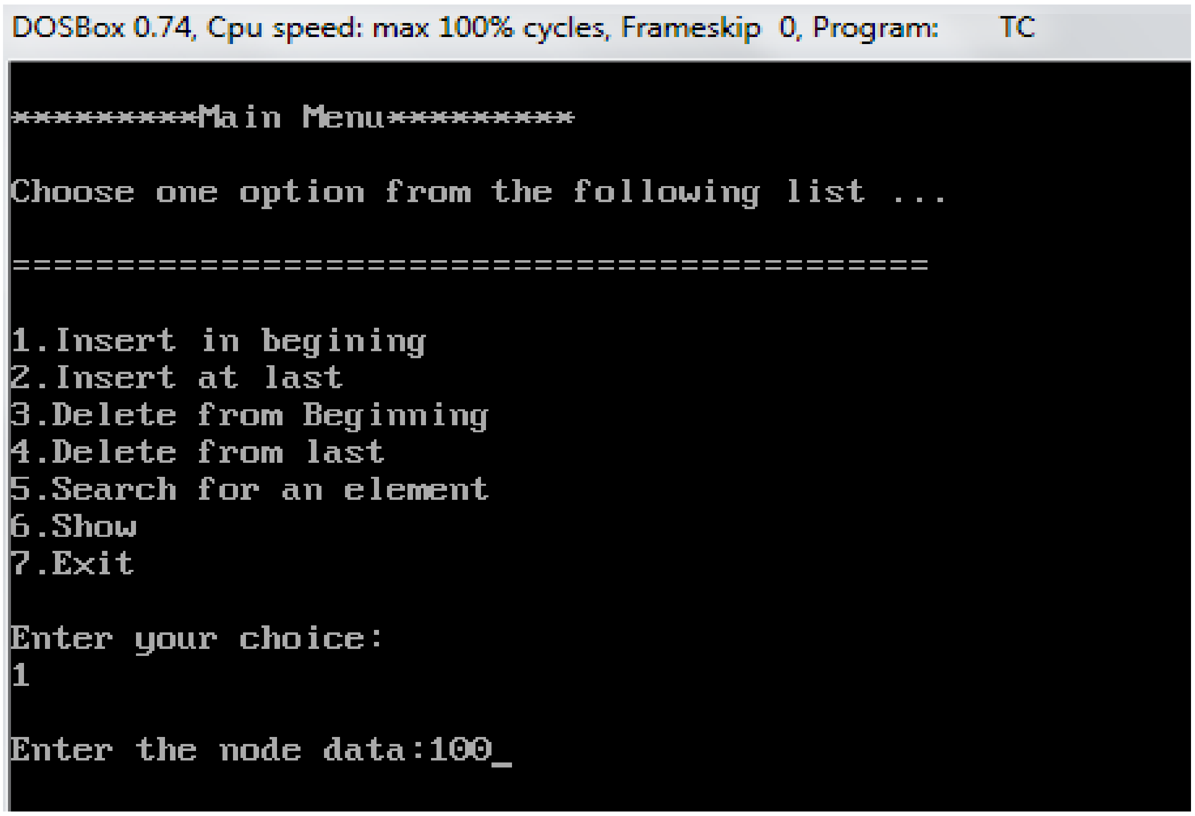


SN	Operation	Description
1	Deletion at beginning	Removing the node from circular singly linked list at the beginning.
2	Deletion at the end	Removing the node from circular singly linked list at the end.
3	Searching	Compare each element of the node with the given item and return the location at which the item is present in the list otherwise return null.
4	Traversing	Visiting each element of the list at least once in order to perform some specific operation.

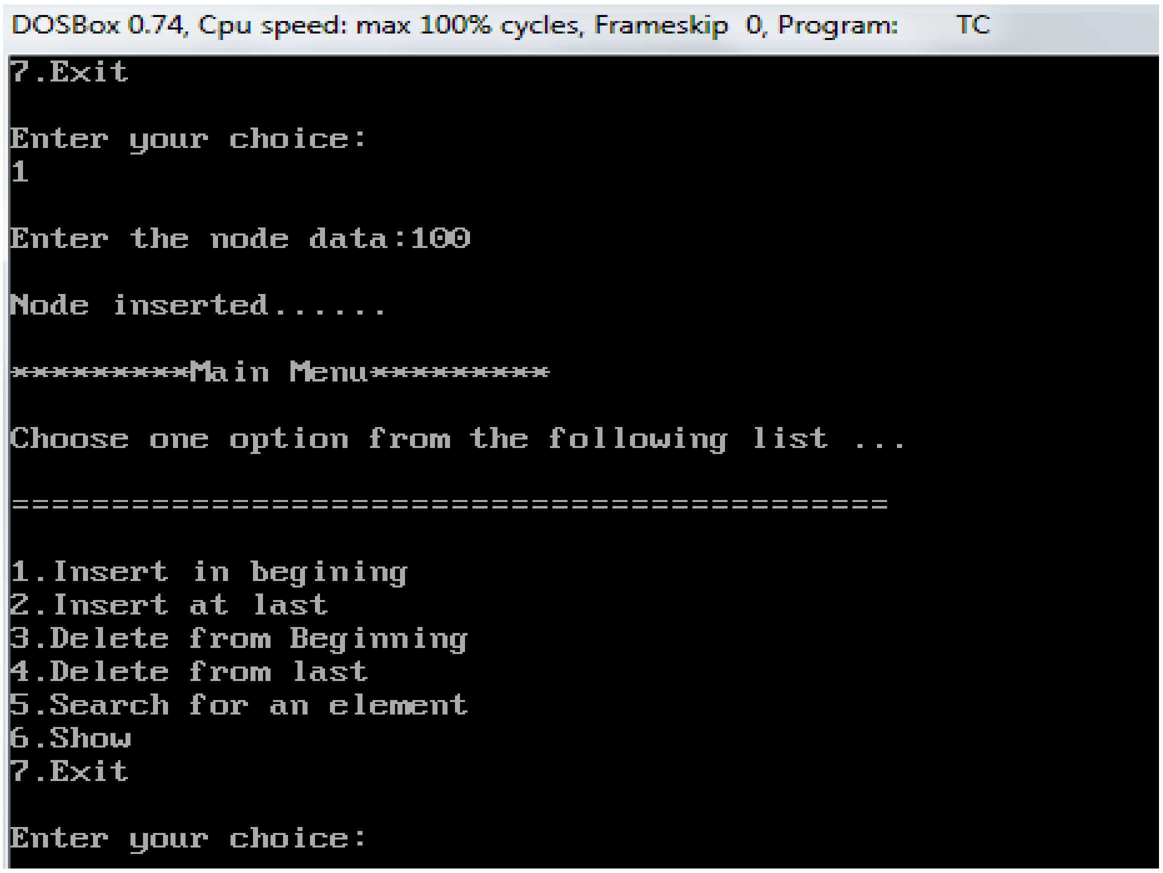
Example

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

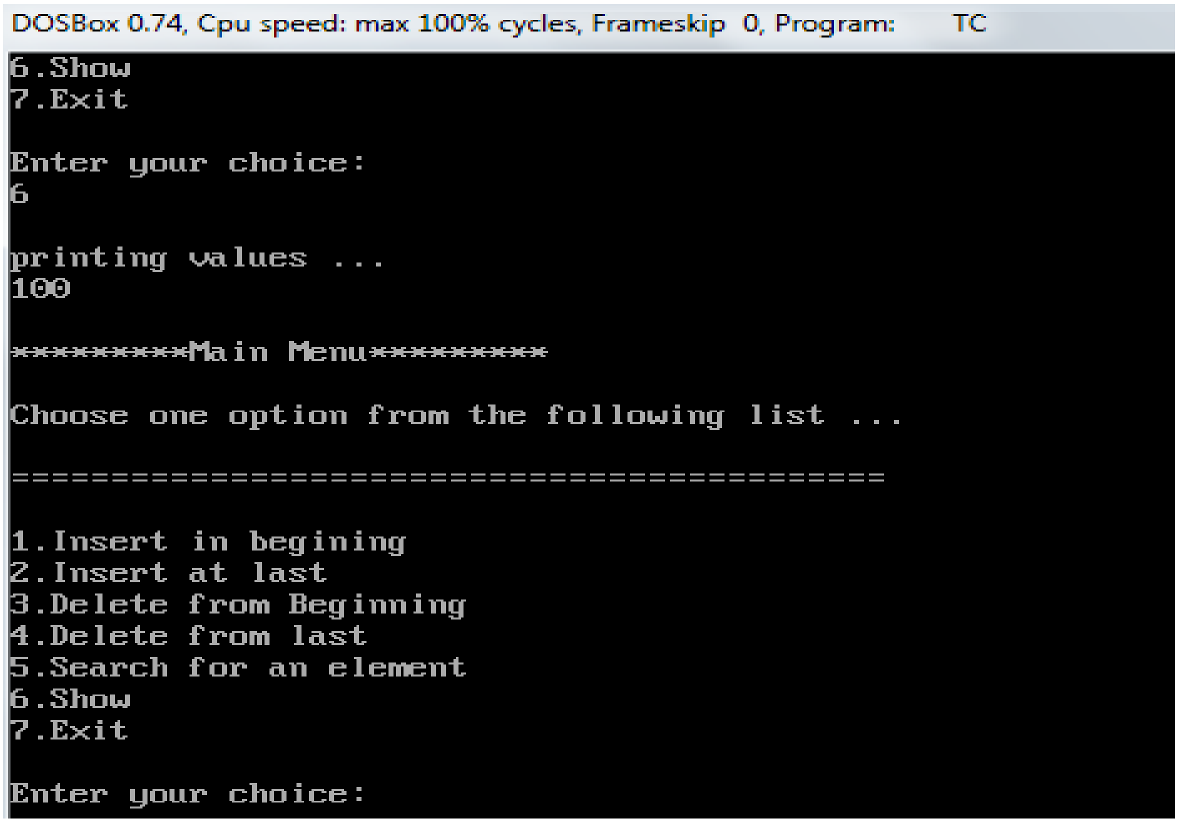
```
struct node
{
   int data;
   struct node *next;
}*head=NULL;
void beginsert ();
void lastinsert ();
void randominsert();
void begin_delete();
void last_delete();
void random_delete();
void display();
void search();
void main ()
{
   int choice;
   clrscr();
   while(1)
   {
      printf("\n*********Main Menu*********\n");
      printf("\nChoose one option from the following list ...\n");
      printf("\n===============================================\n");
      printf("\n1.Insert in begining\n2.Insert at last\n3.Delete
from Beginning\n4.Delete from last\n5.Search for an element\n6.Show
\n7.Exit\n");
      printf("\nEnter your choice:\n");
      scanf("\n%d",&choice);
      switch(choice)
      {
      case 1:
         beginsert();
         break;
      case 2:
         lastinsert();
         break;
      case 3:
         begin_delete();
         break;
      case 4:
         last_delete();
         break;
      case 5:
         search();
         break;
      case 6:
         display();
         break;
      case 7:
         exit(0);
         break;
      default:
         printf("\nPlease enter valid choice..");
      }
   }
}
void beginsert()
{
   struct node *newnode,*temp;
   newnode = (struct node *)malloc(sizeof(struct node));
   printf("\nEnter the node data:");
   scanf("%d",&newnode->data);
   if(head == NULL)
   {
      head = newnode;
      newnode->next=head;
   }
   else
   {
      temp = head;
      while(temp->next != head)
         temp = temp->next;
      newnode->next = head;
      temp -> next = newnode;
      head = newnode;
   }
   printf("\nNode inserted......\n");
}
void lastinsert()
{
   struct node *newnode,*temp;
   int item;
   newnode=(struct node *)malloc(sizeof(struct node));
   printf("\nEnter Data:");
   scanf("%d",&newnode->data);
   if(head==NULL)
   {
         head=newnode;
         newnode->next=head;
   }
   else
   {
         temp=head;
         while(temp->next!=head)
         {
            temp=temp->next;
         }
         temp->next=newnode;
         newnode->next=head;
   }
   printf("\nNode inserted......\n");
}
void begin_delete()
{
   struct node *temp;
   if(head == NULL)
   {
       printf("\nUNDERFLOW");
   }
   else if(head->next == head)
   {
       head = NULL;
       free(head);
       printf("\nNode deleted.....\n");
   }
   else
   {
         temp=head;
       while(temp->next!=head)
           temp=temp->next;
       temp->next=head->next;
       free(head);
       head=temp->next;
       printf("\nNode deleted.....\n");
   }
}
void last_delete()
{
   struct node *temp, *temp1;
   if(head==NULL)
   {
       printf("\nUNDERFLOW......");
   }
   else if(head->next==head)
   {
       head=NULL;
       free(head);
       printf("\nNode deleted......\n");
   }
   else
   {
       temp=head;
       while(temp->next!=head)
       {
           temp1=temp;
           temp=temp->next;
       }
       temp1->next=temp->next;
       free(temp);
       printf("\nNode deleted.....\n");
   }
}
void search()
{
   struct node *temp;
   int item,i=0,flag=1;
   temp= head;
   printf("\nEnter item which you want to search:\n");
   scanf("%d",&item);
   if(head->data==item)
   {
       printf("item found at location %d",i+1);
       flag=0;
   }
   else
   {
       while(temp->next!= head)
       {
           if(temp->data==item)
           {
               printf("item found at location:%d ",i+1);
               flag=0;
               break;
           }
           else
           {
               flag=1;
           }
           i++;
           temp=temp->next;
       }
   }
   if(flag != 0)
   {
       printf("\nItem not found.....\n");
   }
}
void display()
{
   struct node *temp;
   temp=head;
   if(head==NULL)
   {
       printf("\nNothing to print");
   }
   else
   {
       printf("\nprinting values ... \n");
       while(temp->next!=head)
       {
           printf("%d\t",temp->data);
           temp=temp->next;
       }
       printf("%d\n",temp->data);
   }
}
```



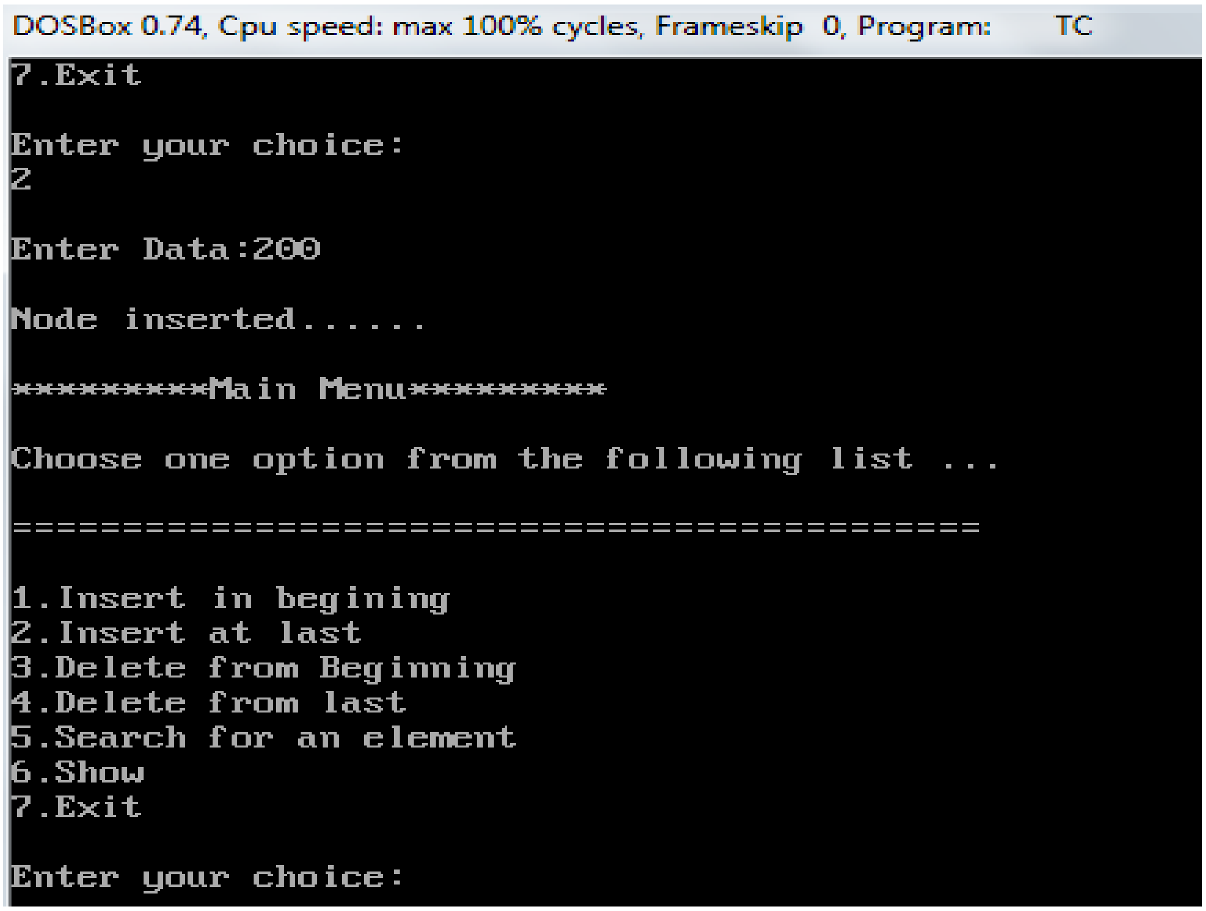
```
DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip  0, Program:     TC

**********Main Menu**********

Choose one option from the following list ...

===============================================

1.Insert in begining
2.Insert at last
3.Delete from Beginning
4.Delete from last
5.Search for an element
6.Show
7.Exit

Enter your choice:
1

Enter the node data:100_
```


DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC

```
7.Exit

Enter your choice:
1

Enter the node data:100

Node inserted......

**********Main Menu**********

Choose one option from the following list ...

==============================================

1.Insert in begining
2.Insert at last
3.Delete from Beginning
4.Delete from last
5.Search for an element
6.Show
7.Exit

Enter your choice:
```


DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC

```
6.Show
7.Exit

Enter your choice:
6

printing values ...
100

**********Main Menu**********

Choose one option from the following list ...

==============================================

1.Insert in begining
2.Insert at last
3.Delete from Beginning
4.Delete from last
5.Search for an element
6.Show
7.Exit

Enter your choice:
```



DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC

```
7.Exit

Enter your choice:
2

Enter Data:200

Node inserted......

**********Main Menu**********

Choose one option from the following list ...

============================================

1.Insert in begining
2.Insert at last
3.Delete from Beginning
4.Delete from last
5.Search for an element
6.Show
7.Exit

Enter your choice:
```