EXPERIMENT 3

Singly Circular Linked List

Program:

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
struct node
{
int data;
struct node *next;
}*head=NULL;
void create(int n)
{
int data,i=1;
struct node *newNode,*temp;
printf("\nEnter Your Data For Node %d:\t",i);
scanf("%d",&data);
newNode=(struct node*)malloc(sizeof(struct node));
if(newNode==NULL)
{
 printf("Cannot Create Starting Node\n");
 exit(0);
}
```

```
newNode->data=data;
newNode->next=NULL;
newNode->next=newNode;
head=newNode;
temp=head;
for(i=2;i<=n;i++)
{
 printf("\nEnter Your Data For Node %d:\t",i);
 scanf("%d",&data);
 newNode=(struct node*)malloc(sizeof(struct node*));
 if(newNode==NULL)
 {
 printf("Cannot Create Node %d\n",i);
 exit(0);
 }
 newNode->data=data;
 newNode->next=head;
 temp->next=newNode;
 temp=temp->next;
}
}
void print()
{
int i=1;
struct node* temp;
```

```
temp=head;
if(temp==NULL)
{
 printf("Linked List Not Available\n");
 exit(0);
}
printf("Linked List elements are\n");
while(temp->next!=head)
{
 printf("Value At Node %d is %d and address of next Node is
%d\n",i,temp->data,temp->next);
 temp=temp->next;
 i++;
}
printf("Value At Node %d is %d and address of next Node is
%d\n",i,temp->data,temp->next);
}
void insert_beg()
{
struct node *newNode,*temp;
int data;
temp=head;
if(temp==NULL)
{
 printf("Linked List Not Available and thus creating a new linked list
with single node\n");
```

```
 printf("\nEnter Data For New Node:\t");
 scanf("%d",&data);
 newNode=(struct node*)malloc(sizeof(struct node));
 if(newNode==NULL)
 {
 printf("Cannot Create a new Node\n");
 exit(0);
 }
 head=newNode;
 head->next=head;
 printf("\nUpdated Linked List is:\t");
 print();
}
else
{
 printf("\nEnter Data For New Node:\t");
 scanf("%d",&data);
 newNode=(struct node*)malloc(sizeof(struct node));
 if(newNode==NULL)
 {
 printf("Cannot Create New Node\n");
 exit(0);
 }
 newNode->data=data;
 newNode->next=head;
```

```
 while(temp->next!=head)
 {
 temp=temp->next;
 }
 temp->next=newNode;
 head=newNode;
 printf("\nUpdated Linked List is:\t");
 print();
}
}
void insert_end()
{
struct node *temp,*newNode;
int data;
temp=head;
if(temp==NULL)
{
 printf("Linked List not available thus creating a new Linked List with
single node\n");
 printf("\nEnter Data For New Node:\t");
 scanf("%d",&data);
 newNode=(struct node*)malloc(sizeof(struct node));
 if(newNode==NULL)
 {
 printf("Cannot Create a New Node thus exiting the program\n");
```

```
 exit(0);
 }
 newNode->data=data;
 head=newNode;
 head->next=head;
 printf("\nUpdated Linked List is :\t");
 print();
}
else
{
 printf("\nEnter Data For New Node:\t");
 scanf("%d",&data);
 newNode=(struct node*)malloc(sizeof(struct node));
 if(newNode==NULL)
 {
 printf("Cannot Create a New Node thus exiting the program\n");
 exit(0);
 }
 newNode->data=data;
 newNode->next=head;
 while(temp->next!=head)
 {
 temp=temp->next;
 }
 temp->next=newNode;
```

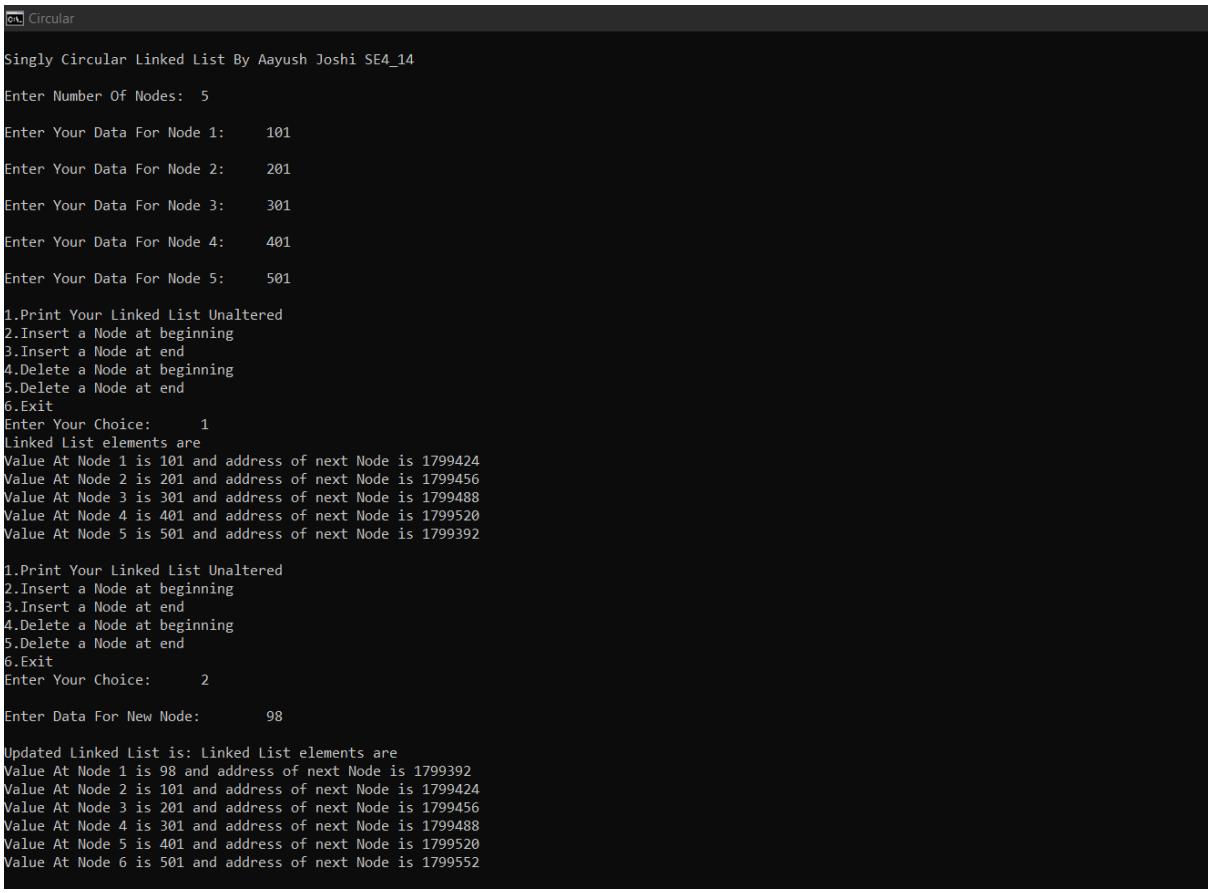
```
 printf("\nUpdated Linked List is:\t");
 print();
}
}
void delete_beg()
{
struct node* temp;
int data;
temp=head;
if(temp==NULL)
{
 printf("Linked List Not Available thus exiting the program\n");
 exit(0);
}
data=temp->data;
printf("Deleted Value From Linked List is %d\n",data);
while(temp->next!=head)
{
 temp=temp->next;
}
temp->next=head->next;
temp=head;
head=temp->next;
free(temp);
printf("\nUpdated Linked List is:\t");
```

```
print();
}
void delete_end()
{
struct node *temp,*t;
int data;
temp=head;
if(temp==NULL)
{
 printf("Linked List Not Available thus exiting the program\n");
 exit(0);
}
while(temp->next!=head)
{
 t=temp;
 temp=temp->next;
}
data=temp->data;
printf("Deleted Value From Linked List is %d\n",data);
t->next=temp->next;
free(temp);
printf("\nUpdated Linked List:\t");
print();
}
void main()
```

```
{
int n,ch;
printf("\nSingly Circular Linked List By Aayush Joshi SE4_14\n");
printf("\nEnter Number Of Nodes:\t");
scanf("%d",&n);
create(n);
while(1)
{
printf("\n1.Print Your Linked List Unaltered\t");
printf("\n2.Insert a Node at beginning\t");
printf("\n3.Insert a Node at end\t");
printf("\n4.Delete a Node at beginning\t");
printf("\n5.Delete a Node at end\t");
printf("\n6.Exit\t");
printf("\nEnter Your Choice:\t");
scanf("%d",&ch);
switch(ch)
{
 case 1:print();
break;
 case 2:insert_beg();
break;
 case 3:insert_end();
break;
 case 4:delete_beg();
```

```
break;
 case 5:delete_end();
break;
 case 6:exit(0);
break;
 default:printf("Wrong Choice\n");
}
}
getch();
}
```

Output:



```
1.Print Your Linked List Unaltered
2.Insert a Node at beginning
3.Insert a Node at end
4.Delete a Node at beginning
5.Delete a Node at end
6.Exit
Enter Your Choice:      3

Enter Data For New Node:        601

Updated Linked List is: Linked List elements are
Value At Node 1 is 98 and address of next Node is 1799392
Value At Node 2 is 101 and address of next Node is 1799424
Value At Node 3 is 201 and address of next Node is 1799456
Value At Node 4 is 301 and address of next Node is 1799488
Value At Node 5 is 401 and address of next Node is 1799520
Value At Node 6 is 501 and address of next Node is 1799584
Value At Node 7 is 601 and address of next Node is 1799552

1.Print Your Linked List Unaltered
2.Insert a Node at beginning
3.Insert a Node at end
4.Delete a Node at beginning
5.Delete a Node at end
6.Exit
Enter Your Choice:      4
Deleted Value From Linked List is 98

Updated Linked List is: Linked List elements are
Value At Node 1 is 101 and address of next Node is 1799424
Value At Node 2 is 201 and address of next Node is 1799456
Value At Node 3 is 301 and address of next Node is 1799488
Value At Node 4 is 401 and address of next Node is 1799520
Value At Node 5 is 501 and address of next Node is 1799584
Value At Node 6 is 601 and address of next Node is 1799392
```

```
1.Print Your Linked List Unaltered
2.Insert a Node at beginning
3.Insert a Node at end
4.Delete a Node at beginning
5.Delete a Node at end
6.Exit
Enter Your Choice:      5
Deleted Value From Linked List is 601

Updated Linked List:    Linked List elements are
Value At Node 1 is 101 and address of next Node is 1799424
Value At Node 2 is 201 and address of next Node is 1799456
Value At Node 3 is 301 and address of next Node is 1799488
Value At Node 4 is 401 and address of next Node is 1799520
Value At Node 5 is 501 and address of next Node is 1799392

1.Print Your Linked List Unaltered
2.Insert a Node at beginning
3.Insert a Node at end
4.Delete a Node at beginning
5.Delete a Node at end
6.Exit
Enter Your Choice:      6

Press any key to continue . . .
```