# PRACTICAL NO : 8

**AIM** :- Implement a Circular Single Linked List and Perform the operation : Create, Traverse, Insert_beg, Insert_end, Delete_beg, Delete_end using Menu Driver Program.

PROGRAM :

```
#include<stdio.h>
#include<stdlib.h>
struct node{
   int data;
   struct node*next;
};
struct node *s, *p, *q, *a, *t;

void create(){
   printf("Creating the Circular Linked List(CLL).\nEnter data for the First node:\t");
   p = (struct node *)malloc(sizeof(struct node));
   scanf("%d", &p -> data);
   p -> next = p;
   s = p;
}

void Traverse(){
   printf("\nTraversing the linked list:\t");
   t = s;
   do{
      printf("%d\t", t -> data);
      t = t -> next;
   } while (t != s);
```

```
}



void Insert_Beg()
{   printf("\nInserting node at beggining.\nEnter the data:\t");
    p = (struct node *)malloc(sizeof(struct node));
    scanf("%d", &(p -> data));

    if(s == NULL){
        p -> next = p;
        s = p;
    }else{
        t = s;
        while (t -> next != s)
        {
            t = t-> next;
        }
        p->next = s;
        t->next = p;
        s = p;
    }



}

void Insert_End()
{
    t =s;
```

```
   while(t -> next != s){
      t = t -> next;
   }

   p = (struct node*)malloc(sizeof(struct node));
   printf("\nEnter data of last node:\t");
   scanf("%d", &(p -> data));
   p -> next = s;
   t -> next =  p;
}

void Delete_Beg()
{   printf("\nDeleting the node at beggining..\n");
if (s == NULL)
{
  printf("The linked list is empty..\n");
}

   t = s;
   while(t -> next != s){
      t = t-> next;
   }
   q = s -> next;
   t -> next = q;
   free(s);
   s = q;

}
```

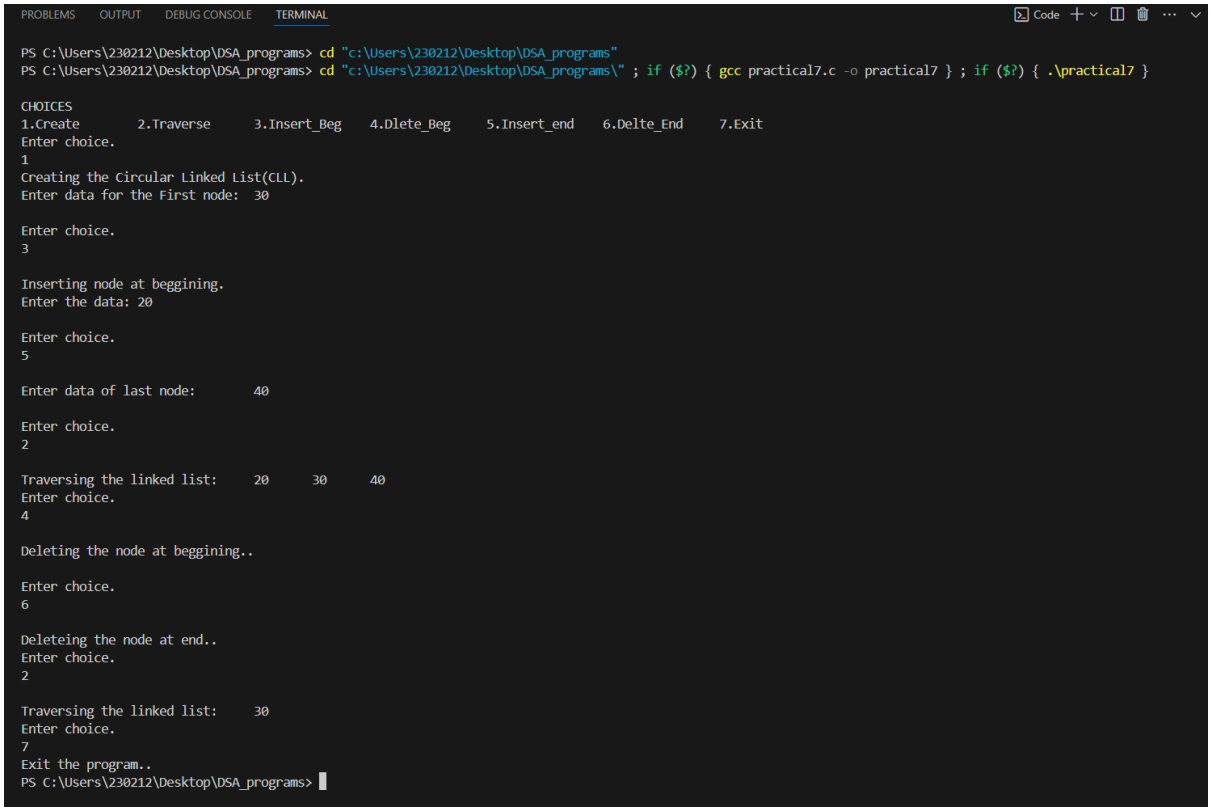
# PRACTICAL NO : 8

```
void Delete_End()
{   printf("\nDeleteing the node at end..");
   t = s;
   while(t -> next != s){
      q = t;
      t = t-> next;
   }
   q -> next = s;
   free(t);
}
int main(){
   int choice;


printf("\nCHOICES\n1.Create\t2.Traverse\t3.Insert_Beg\n4.Dlete_Beg\t5.Insert_end\t6.
Delte_End\t7.Exit");
   do{
      printf("\nEnter valid choice.\n");
      scanf("%d", &choice);
      switch (choice)
      {
      case 1:
         create();
         break;

      case 2:
         Traverse();
         break;
```

```
        case 3:
           Insert_Beg();
           break;

        case 4:
           Delete_Beg();
           break;

        case 5:
           Insert_End();
           break;

        case 6:
           Delete_End();
           break;

        case 7:
           printf("Exit the program..");
           break;

        default:
        printf("\nPlease enter a valid choice\n");
           break;
        }
    }while(choice != 7);

}
```

# PRACTICAL NO : 8

## OUTPUT




GITHUB LINK : https://github.com/AmolNagargoje04/Data-Structure-practical