Write a C Program to implement circular Queue Operations Using Arrays.

```
#include<stdio.h>
#include<stdlib.h>
#define maxsize 5
void insert();
void delete(); void display();
int front = -1, rear = -1; int queue[maxsize];
void main ()
{
int choice; while(choice != 3)
{

printf("\n1.insert an element\n2.Delete an element\n3.Display the queue\n"); printf("\nEnter your choice ?");
scanf("%d",&choice); switch(choice)
{
case 1:
insert(); break; case 2:
delete(); break; case 3:
display(); break;
default:
printf("\nEnter valid choice??\n");
}
}
}
void insert()
{
int item;
printf("\nEnter the element\n"); scanf("%d",&item); if((rear+1)%maxsize == front)
{
printf("\nOVERFLOW"); return;
}
else if(front == -1 && rear == -1)
{
front = 0;
rear = 0;
}
else if(rear == maxsize -1 && front != 0)
{
rear = 0;
}
else
{
rear = (rear+1)%maxsize;
```

```
}
queue[rear] = item; printf("\nValue inserted ");
}
void delete()
{
int item;
if(front == -1 & rear == -1)
{
printf("\nUNDERFLOW\n"); return;
}
else if(front == rear)
{
front = -1;
rear = -1;
}
else if(front == maxsize -1)
{
front = 0;
}
else
front = front + 1;
}

void display()
{
int i;
if(front == -1)
printf("\nCircular Queue is Empty!!!\n");
else
{
i = front;
printf("\nCircular Queue Elements are : \n"); if(front <= rear){
while(i <= rear)
printf("%d %d %d\n",queue[i++],front,rear);
}
else{
while(i <= maxsize - 1)
printf("%d %d %d\n", queue[i++],front,rear); i = 0;
while(i <= rear)
printf("%d %d %d\n",queue[i++],front,rear);
}
}
}
```

Q. Write a program in C to delete the node from the beginning of a Singly Linked List.

```
#include<stdlib.h>
#include <stdio.h>
void create();
void display();
struct node
{
 int info;
 struct node *next;
};
struct node *start=NULL;
int main()
{
 int choice;
 while(1){
 printf("\n 1.Create \n");
 printf("\n 2.Display \n");
 printf("\n 3. delete from beg \n");
 printf("Enter your choice:\t");
 scanf("%d",&choice);
 switch(choice)
 {
 case 1:
 create();
 break;
 case 2:
 display();
 break;
 case 3:
 delete_begin();
 break;
 default:
 printf("\n Wrong Choice:n");
 break;
 }
 }
 return 0;
}
void create()
{
 struct node *temp,*ptr;
```

```
 temp=(struct node *)malloc(sizeof(struct node));

 printf("\nEnter the data value for the node:\t");
 scanf("%d",&temp->info);
 temp->next=NULL;
 if(start==NULL)
 {
 start=temp;
 }
 else
 {
 ptr=start;
 while(ptr->next!=NULL)
 {
 ptr=ptr->next;
 }
 ptr->next=temp;
 }
}


void delete_begin()
{
 struct node *ptr;
 if(ptr==NULL)
 {
 printf("\n List is Empty:\n");
 return;
 }
 else
 {
 ptr=start;
 start=start->next ;
 printf("\n The deleted element is :%d ",ptr->info);
 free(ptr);
 }
}

void display()
{
 struct node *ptr;

 printf("\n The List elements are:\t");
 for(ptr=start;ptr!=NULL;ptr=ptr->next)
```

```
 printf("%d \n",ptr->info );  }
```

Q. Write a program in C to delete the node from the end of a Singly Linked List.

```
#include<stdlib.h>
#include <stdio.h>
void create();

void display();
struct node
{
 int info;
 struct node *next;
};
struct node *start=NULL;
int main()
{
 int choice;

 while(1){
 printf("\n 1.Create \n");
 printf("\n 2.Display \n");
 printf("\n 3. delete from end \n");
 printf("Enter your choice:\t");
 scanf("%d",&choice);
 switch(choice)
 {
 case 1:create();
break;
 case 2:display();
break;
 case 3:delete_end();
break;
 default:
 printf("\n Wrong Choice:n");break;
 }
 }
 return 0;
}
void create()
{
 struct node *temp,*ptr;
 temp=(struct node *)malloc(sizeof(struct node));
```

```
 printf("\nEnter the data value for the node:\t");
 scanf("%d",&temp->info);
 temp->next=NULL;
 if(start==NULL)
 {
 start=temp;
 }
 else
 {
 ptr=start;
 while(ptr->next!=NULL)
 {
 ptr=ptr->next;
 }
 ptr->next=temp;
 }
}


void delete_end()
{
 struct node *temp,*ptr;
 if(start==NULL)
 {
 printf("\n List is Empty:");
 exit(0);
 }
 else if(start->next ==NULL)
 {
 ptr=start;
 start=NULL;
 printf("\n The deleted element is:%d ",ptr->info);
 free(ptr);
 } else
 {
 ptr=start;
 while(ptr->next!=NULL)
 {
 temp=ptr;
 ptr=ptr->next;
 }
 temp->next=NULL;
 printf("\n The deleted element is:%d ",ptr->info);
 free(ptr);
```

```
 } }
void display()
{
 struct node *ptr;

 printf("\n The List elements are:\t");
 for(ptr=start;ptr!=NULL;ptr=ptr->next)
 printf("%d \n",ptr->info );
}
```

Q. Write a program in C to delete the node from the random given position of a Singly Linked List.

```
#include<stdlib.h>
#include <stdio.h>
void create();
void display();
struct node
{
 int info;
 struct node *next;
};
struct node *start=NULL;
int main()
{
 int choice;
 while(1){
 printf("\n 1.Create \n");
 printf("\n 2.Display \n");
 printf("\n 3. delete from given position \n");
 printf("Enter your choice:\t");
 scanf("%d",&choice);
 switch(choice)
 {
 case 1:create();break;
 case 2:display();break;
 case 3:delete_pos();break;
 default:
 printf("\n Wrong Choice:n");break;
 }
 }
 return 0;
}
void create()
```

```
{
 struct node *temp,*ptr;
 temp=(struct node *)malloc(sizeof(struct node));

 printf("\nEnter the data value for the node:\t");
 scanf("%d",&temp->info);
 temp->next=NULL;
 if(start==NULL)
 {
 start=temp;
 }
 else
 {
 ptr=start;
 while(ptr->next!=NULL)
 {
 ptr=ptr->next;
 }
 ptr->next=temp;
 }
}
 void delete_pos()
{
 int i,pos;
 struct node *temp,*ptr;
 if(start==NULL)
 {
 printf("\n The List is Empty:\n");
 exit(0);
 }
 else
 {
 printf("\n Enter the position of the node to be deleted: ");
 scanf("%d",&


pos);
 if(pos==0)
 {
 ptr=start;
 start=start->next ;
 printf("\n The deleted element is:%d ",ptr->info );
 free(ptr);
 }
```

```
 else
 {
 ptr=start;
 for(i=0;i<pos;i++) { temp=ptr; ptr=ptr->next ;
 if(ptr==NULL)
 {
 printf("\n Position not Found:\n");
 return;
 }}
 temp->next =ptr->next ;
 printf("\n The deleted element is:%d ",ptr->info );
 free(ptr); } }}
```

Q. Write a program in C to insert a new node from the begining of a Linked List.

```
#include<stdlib.h>
#include <stdio.h>
void create();
void display();
struct node
{
 int info;
 struct node *next;
};
struct node *start=NULL;
int main()
{
 int choice;
 while(1){
 printf("\n 1.Create \n");
 printf("\n 2.Display \n");
 printf("\n 3. insert new node in begining \n");
 printf("Enter your choice:\t");
 scanf("%d",&choice);
 switch(choice)
 {
 case 1:create();break;
 case 2:display();break;
 case 3:insert_begin();break;
 default:
 printf("\n Wrong Choice:n");break;
 }
 }
 return 0;
```

```
}
void create()
{
 struct node *temp,*ptr;
 temp=(struct node *)malloc(sizeof(struct node));

 printf("\nEnter the data value for the node:\t");
 scanf("%d",&temp->info);
 temp->next=NULL;
 if(start==NULL)
 {
 start=temp;
 }
 else
 {
 ptr=start;
 while(ptr->next!=NULL)
 {
 ptr=ptr->next;
 }
 ptr->next=temp;
 }
}

 void insert_begin()
{
 struct node *temp;
 temp=(struct node *)malloc(sizeof(struct node));
 if(temp==NULL)
 {
 printf("\n Out of Memory Space:\n ");
 return;
 }
 printf("\n Enter the data value for the node: " );
 scanf("%d",&temp->info);
 temp->next =NULL;
 if(start==NULL)
 {
 start=temp;
 }
 else
 {
 temp->next=start;
 start=temp;
```

```
 }
}
void display()
{
 struct node *ptr;

 printf("\n The List elements are:\t");
 for(ptr=start;ptr!=NULL;ptr=ptr->next)
 printf("%d \n",ptr->info );
}
```

Q. Write a program in C to insert a new node from the end of a Linked List.

```
#include<stdlib.h>
#include <stdio.h>
void create();
void display();
struct node
{
 int info;
 struct node *next;
};
struct node *start=NULL;
int main()
{
 int choice;
 while(1){
 printf("\n 1.Create \n");
 printf("\n 2.Display \n");
 printf("\n 3. insert from end \n");
 printf("Enter your choice:\t");
 scanf("%d",&choice);
 switch(choice)
 {
 case 1:create();break;
 case 2:display();break;
 case 3:insert_end();break;
 default:
 printf("\n Wrong Choice:n");break;
 }
 }
 return 0;
}
void create()
```

```
{
 struct node *temp,*ptr;
 temp=(struct node *)malloc(sizeof(struct node));

 printf("\nEnter the data value for the node:\t");
 scanf("%d",&temp->info);
 temp->next=NULL;
 if(start==NULL)
 {
 start=temp;
 }
 else
 {
 ptr=start;
 while(ptr->next!=NULL)
 {
 ptr=ptr->next;
 }
 ptr->next=temp;
 }
}
void insert_end()
{
 struct node *temp,*ptr;
 temp=(struct node *)malloc(sizeof(struct node));
 if(temp==NULL)
 {
 printf("\n Out of Memory Space:\n");
 return;
 }
 printf("\n Enter the data value for the node: " );
 scanf("%d",&temp->info );
 temp->next =NULL;
 if(start==NULL)
 {
 start=temp;
 }
 else
 {
 ptr=start;
 while(ptr->next !=NULL)
 {
 ptr=ptr->next ;
 }
```

```
 ptr->next =temp;
 }
}
void display()
{
 struct node *ptr;

 printf("\n The List elements are:\t");
 for(ptr=start;ptr!=NULL;ptr=ptr->next)
 printf("%d \n",ptr->info );
}
void display()
{
 struct node *ptr;

 printf("\n The List elements are:\t");
 for(ptr=start;ptr!=NULL;ptr=ptr->next)
 printf("%d \n",ptr->info );
}
```

Q. Write a program in C to insert a new node on a given position of a Linked List.

```
#include<stdlib.h>
#include <stdio.h>
void create();
void display();
struct node
{
 int info;
 struct node *next;
};
struct node *start=NULL;
int main()
{
 int choice;
 while(1){
 printf("\n 1.Create \n");
 printf("\n 2.Display \n");
 printf("\n 3. insert new node on a given postion \n");
 printf("Enter your choice:\t");
 scanf("%d",&choice);
 switch(choice)
 {
```

```
 case 1:create();break;
 case 2:display();break;
 case 3:insert_pos();break;
 default:
 printf("\n Wrong Choice:n");break;
 }
 }
 return 0;
}
void create()
{
 struct node *temp,*ptr;
 temp=(struct node *)malloc(sizeof(struct node));

 printf("\nEnter the data value for the node:\t");
 scanf("%d",&temp->info);
 temp->next=NULL;
 if(start==NULL)
 {
 start=temp;
 }
 else
 {
 ptr=start;
 while(ptr->next!=NULL)
 {
 ptr=ptr->next;
 }
 ptr->next=temp;
 }
}

 void insert_pos()
{
 struct node *ptr,*temp;
 int i,pos;
 temp=(struct node *)malloc(sizeof(struct node));
 if(temp==NULL)
 {
    printf("\n Out of Memory Space:\n");
 return;
 }
 printf("\n Enter the position for the new node to be inserted: ");
 scanf("%d",&pos);
```

```
 printf("\n Enter the data value of the node: ");
 scanf("%d",&temp->info) ;
 temp->next=NULL;
 if(pos==0)
 {
 temp->next=start;
 start=temp;
 }
 else
 {
 for(i=0,ptr=start;i<pos-1;i++)
{ ptr=ptr->next;
 if(ptr==NULL)
 {
 printf("\n Position not found:[Handle with care]\n");
 return;
 }
 }
 temp->next =ptr->next ;
 ptr->next=temp;
 }
}
void display()
{
 struct node *ptr;

 printf("\n The List elements are:\t");
 for(ptr=start;ptr!=NULL;ptr=ptr->next)
 printf("%d \n",ptr->info );
}
```