

Se dan las estructuras siguientes:

```
struct Tnodo {
    int dato;
    struct Tnodo * siguiente;
} ;

struct TlistaNodos {
    struct Tnodo * inicio;
} ;
```

para definir una lista de nodos circular simplemente enlazados. Implementar las funciones descritas en la sección de prototipos, probando su funcionamiento en un programa.

SOLUCIÓN:

// defines

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

// estructuras

```
typedef struct Tnodo{
    int dato;
    struct Tnodo * siguiente;
}Nodo;

typedef struct TlistaNodos{
    Nodo * inicio;
}ListaNodo;
```

// prototipos

```
void iniciarLista(struct TlistaNodos *);
```

```
void insertarListaCircular (int, struct TlistaNodos *);
// crea un nodo, cuyo campo dato es un integer e inserta dicho nodo en una lista
```

```
void recorrerListaCircular (struct TlistaNodos);
// recorre una lista y printa el contenido del campo dato de cada nodo
```

```
// main
```

```
ListaNodo lista;  
int i = 0;  
  
iniciarLista(&lista);  
//Comprobacion si funciona  
for(i = 10; i > 0; i--){  
    insertarListaCircular(i, &lista);  
}  
recorrerListaCircular(&lista);
```

```
// implementacion de las funciones
```

```
void iniciarLista(ListaNodo * listaNodos){  
    listaNodos -> inicio = NULL;  
}  
  
void insertarListaCircular (int pos, ListaNodo * listaNodos){  
    Nodo * nuevo;  
  
    nuevo = malloc(sizeof(struct Tnodo));  
    (* nuevo).dato = pos;  
    (* nuevo).siguiente = (* listaNodos).inicio;  
    (* listaNodos).inicio = nuevo;  
}  
  
void recorrerListaCircular (ListaNodo * listaNodos){  
    Nodo * actual;  
  
    actual = listaNodos -> inicio;  
    while(actual != NULL){  
        printf ("%d ", actual -> dato);  
        actual = actual -> siguiente;  
    }  
}
```