DOUBLY CIRCULAR LINKED LIST--------

#include<stdio.h>

#include<stdlib.h>

struct node{

int data;

struct node \*next;

struct node \*prev;

};

int main()

{

int n , i , item ;

struct node \*p ,\*new\_node , \*head;

printf("Enter the number of nodes");

scanf("%d",&n);

printf("Enter the '0'th index node");

scanf("%d",&item);

new\_node = (struct node \*)malloc(sizeof(struct node \*));

new\_node->data = item;

new\_node->next = NULL;

new\_node->prev = NULL;

head = new\_node;

p = new\_node;

for(i=1;i<n;i++)

{

printf("Enter the next node");

scanf("%d",&item);

new\_node = (struct node \*)malloc(sizeof(struct node \*));

new\_node->data = item;

new\_node->next = NULL;

new\_node->prev = p;

p -> next = new\_node ;

p = p->next;

}

p = head;

while(p!=NULL)

{

printf(" %d",p->data);

p = p->next;

}

}

SINGLE CIRCLAR LINKED LIST-------

#include<stdio.h>

#include<stdlib.h>

typedef struct node

{

int data;

struct node \*next;

}node;

int main()

{

int a , i = 1 , n ,r;

//

i = 0 is also ok....

node \*p,\*q,\*start;

printf("Enter the number of nodes");

scanf("%d",&n);

printf("Enter node %d \n",i);

// you can also start with i = 0

p = (node\*)malloc(sizeof(node));

scanf("%d",&a);

p->data = a;

p->next = NULL;

start = p;

for(i=2;i<=n;i++)

//if i=0 , then for ( i = 1; i < n; i++ )

{

printf("Enter node %d \n",i);

q = (node\*)malloc(sizeof(node));

scanf("%d",&a);

q->data = a;

q->next = NULL;

p->next = q;

p = p->next;

}

p->next = start;

// connect last node to first(start) node

printf("\t %d", start->data);

p = start->next;

while(p!=start)

{

printf("\t %d", p->data);

p = p->next;

}