CIRCULAR DOUBLY LINK LIST

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

struct Node
    {
```

```
    int data;
    struct Node* prev;
    struct Node* next;

  };


void insertbegin(struct Node** head)
  {
    struct Node* temp;
    struct Node* newnode;
    int key;
    printf("Enter the value for inserting at begining:\n");
    scanf("%d",&key);
    newnode=(struct Node*)malloc(sizeof(struct Node));
    newnode->data=key;
    newnode->prev=NULL;
    newnode->next=NULL;


    if(*head==NULL)
    {  *head=newnode;
       newnode->prev=*head;
       newnode->next=*head;
       printf("\n--first element--\n(%d)",(*head)->data);
       printf("\n---0000--\n");
```

```
        }
        else if(*head!=NULL)
        {

            newnode->next=*head;
            newnode->prev=(*head)->prev;
            (*head)->prev->next=newnode;
            (*head)->prev=newnode;
            *head=newnode;
            printf("\n--1do--\n");

        }
        else
            {

            }

    }

void deletbegin(struct Node**head)
    {
        struct Node* temp;
        temp=NULL;
        if(*head==NULL)
        {
            printf("linklist is underflow:\n");
        }
        else if((*head)->next == (*head))
        {
            *head = NULL;
            free(*head);
            printf("\nnode deleted\n");
        }

        else
        {
            temp=*head;
            while(temp->next!=(*head))
            {
                temp=temp->next;
            }
            temp->next=(*head)->next;
            (*head)->next->prev=temp;
            free(*head);
            *head=temp->next;




        }
    }
```

```
void printlist(struct Node** head)
  {



       struct Node* temp=*head;
       if((*head)==NULL)
       {
            printf("link list is underflow:\n");
       }
       else
        {

           printf("\nfirst value(%d):\n",(*head)->data);

            printf("\nvalue:\n");
          do
            {
              printf("\n%d \n",temp->data);
              temp=temp->next;
              printf("\n");
            }while(temp!=*head);
        }
  }

int main()
    {   struct Node* head=NULL;
    insertbegin(&head);
    insertbegin(&head);
    insertbegin(&head);

    printlist(&head);
    deletbegin(&head);
    printlist(&head);
    deletbegin(&head);
    printlist(&head);
    deletbegin(&head);
    printlist(&head);
    printlist(&head);

    return 0;
}
```