Doubly Linked lists

Discuss with examples

Various Operations:

1. Insert a node at the beginning
2. Insert a node at the end
3. Print data of all the nodes
4. Delete a node at the beginning
5. Delete a node at the end

```
typedef struct dnode
{
        struct dnode *prev;
       int info;
        struct dnode *next;
}dnode;

dnode *head=NULL;
```

```
struct dnode* insert_begin(struct dnode *head)
{
        struct dnode *temp=malloc(sizeof(struct dnode));
        printf("\nInfo: ");       scanf("%d",&temp->info);
        temp->prev=temp->next=NULL;
        if(head==NULL)
        {
                head=temp;
                return head;
        }
        head->prev=temp;
        temp->next=head;
        head=temp;
        return head;
}
```

```
struct dnode *insert_end(struct dnode *head)
{          struct dnode *temp=malloc(sizeof(struct dnode));
          printf("\nInfo: ");  scanf("%d",&temp->info);
          temp->prev=temp->next=NULL;

          if(head==NULL)
          {         head=temp; return head;  }

          dnode *cur=head;
          while(cur->next!=NULL) cur=cur->next;
          cur->next=temp;
          temp->prev=cur;
          return head;
}
```

```
void print(struct dnode *head)
{
        struct dnode *h=head;
        while(h!=NULL)
        {
                cout<<h->info<<"->";
                h=h->next;
        }
}
```

```
dnode *dnode::delete_begin(dnode *head)
{
        dnode *temp;
        if(head==NULL)
        {       cout<<“DLL is empty”;
                return head;
         }
        temp=head;
        head=head->next; head->prev=NULL;
        free(temp);
        return head;
}
```

```
struct dnode * delete_end(dnode *head)
{       struct dnode *temp;
        if(head==NULL)
        {
                printf("DLL is empty");
                return head;
         }
if(head->next==NULL){ free(head); head=NULL;  return(head);}
         dnode *cur=head;
        while(cur->next!=NULL) cur=cur-> next;
        cur->prev->next=NULL;
        free(cur);
        return head;
}
```

```
int main()
{       struct dnode *head=NULL;    int c,ele, flag=1;
        while(flag==1)
        {       printf("1.Insert begin \n 2.Insert end\n 3.Print \n 4.delete begin\n 5. Delete end\n";
                scanf("%d",&c);
                switch(c)
                {       case 1:head=insert_begin(head);         break;
                        case 2:head=insert_end(head);           break;
                        case 3: print(head); break;
                        case 4: delete_begin(head); break;
                        case 5:  delete_end(head); break;
                        default: flag=0;
                }
        }
        return 0;
}
```

Exercises

Exercises:

Write functions to:

- Concatenate two SLL
- Find the length of a circular linked list
- Delete a node whose data field(key) is given from a DLL
- Insert a node after and before the node whose key is given in a DLL
- Exchange two nodes in a DLL

Circular Doubly Linked List

Discuss with examples

Operations:

- Create
- Insert a node at the beginning
- Insert a node at the end
- Delete a node at the beginning
- Delete a node at the end
- Etc.

```
void list::reverse()
{
        list *prev1,*curr;    prev1=curr=NULL;
        while(first!=NULL)
        {
                prev1=curr;
                curr=first;
                first=first->next;
                curr->next=prev1;
                if(prev1) prev1->prev=curr;
        }
        first=prev1;  printf("\nreversed node is:“);     traverse();
}
```