**III Sem – Sec A**

**DSPD Assignment – 1**

**Group Members:-**

* **Roll No. 52: Niladri Sen**
* **Roll No. 59: Saurabh Thakre**

**Question:**

Implement Bubble Sort on Doubly Linked List.

**Program:**

#include<stdio.h>

#include<stdlib.h>

struct dnode

{

struct dnode \*lpoint;

int data;

struct dnode \*rpoint;

};

void bsort(int);

int append(int num);

void display(struct dnode\*);

struct dnode \*start=NULL;

void main()

{

int n=0,a,m;

do{

printf("press 1 for insertion press 2 for sorting press 0 for exit :");

scanf("%d",&m);

switch(m)

{

case 1:printf("enter the element to be inserted:");

scanf("%d",&a);

n++;

append(a);

display(start);

break;

case 2:printf("sorted list:\n");

bsort(n);

display(start);

break;

case 0:exit(0);

}

}while(m!=0);

}

int append(int num)

{

struct dnode \*r,\*q=start;

if(start==NULL)

{

start=(struct dnode\*)malloc(sizeof (struct dnode));

start->lpoint=NULL;

start->data=num;

start->rpoint=NULL;

}

else

{

while(q->rpoint!=NULL)

q=q->rpoint;

r=(struct dnode\*)malloc(sizeof(struct dnode));

r->data=num;

r->rpoint=NULL;

r->lpoint=q;

q->rpoint=r;

}

return 0;

}

void bsort(int n)

{

int i,t;

struct dnode \*temp;

if(start==NULL)

return;

temp=start;

for(i=0;i<n-1;i++)

{

while (temp->rpoint !=NULL)

{

if ((temp->data )> ((temp->rpoint)->data) )

{

t=temp->data;

temp->data=((temp->rpoint)->data) ;

((temp->rpoint)->data) =t;

}

temp = temp->rpoint;

}

temp=start;

}

}

void display(struct dnode \*q)

{

while(q!=NULL)

{

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

q=q->rpoint;

}

printf("\n");

}

**Output:**

sat123@ubuntu:~$ gcc ass1.c

sat123@ubuntu:~$ ./a.out

press 1 for insertion press 2 for sorting press 0 for exit :1

enter the element to be inserted:5

5

press 1 for insertion press 2 for sorting press 0 for exit :1

enter the element to be inserted:4

5 4

press 1 for insertion press 2 for sorting press 0 for exit :1

enter the element to be inserted:8

5 4 8

press 1 for insertion press 2 for sorting press 0 for exit :1

enter the element to be inserted:2

5 4 8 2

press 1 for insertion press 2 for sorting press 0 for exit :1

enter the element to be inserted:7

5 4 8 2 7

press 1 for insertion press 2 for sorting press 0 for exit :1

enter the element to be inserted:3

5 4 8 2 7 3

press 1 for insertion press 2 for sorting press 0 for exit :1

enter the element to be inserted:9

5 4 8 2 7 3 9

press 1 for insertion press 2 for sorting press 0 for exit :2

sorted list:

2 3 4 5 7 8 9

press 1 for insertion press 2 for sorting press 0 for exit :0

sat123@ubuntu:~$

**Question:**

Write a C program to create a linked list from a already given linked list. The new linked list must contain every alternate element of the existing link list.

**Program:**

#include<stdio.h>

#include<stdlib.h>

struct node listtwo(struct node \*);

struct node listthree(struct node \*);

struct node display(struct node \*);

struct node

{

int data;

struct node \*next;

};

int n;

struct node \*temp,\*temp1,\*temp2,\*x,\*w,\*y,\*z,\*v,\*start2=NULL,\*start1=NULL,\*start=NULL;

void main()

{

int data1,i;

printf("Enter Size:");

scanf("%d",&n);

for(i=0;i<n;i++)

{

printf("Insert Element:");

scanf("%d",&data1);

x=(struct node \*)malloc(sizeof(struct node));

x->data=data1;

x->next=NULL;

if(start==NULL)

{

start=x;

}

else

{

temp=start;

while(temp->next!=NULL)

{

temp=temp->next;

}

temp->next=x;

}

}

printf("Linked List 1:\n");

display(start);

printf("Linked List 2:\n");

listtwo(start);

printf("Linked List 3:\n");

listthree(start);

}

struct node listtwo(struct node \*w)

{

if(n%2==0)

{

while(w!=NULL)

{

z=(struct node \*)malloc(sizeof(struct node));

z->data=w->data;

z->next=NULL;

w=w->next;

w=w->next;

if(start1==NULL)

{

start1=z;

}

else

{

temp1=start1;

while(temp1->next!=NULL)

{

temp1=temp1->next;

}

temp1->next=z;

}

}

}

else

{

while(w->next!=NULL)

{

z=(struct node \*)malloc(sizeof(struct node));

z->data=w->data;

z->next=NULL;

w=w->next;

w=w->next;

if(start1==NULL)

{

start1=z;

}

else

{

temp1=start1;

while(temp1->next!=NULL)

{

temp1=temp1->next;

}

temp1->next=z;

}

}

temp1=start1;

z=(struct node \*)malloc(sizeof(struct node));

z->data=w->data;

while(temp1->next!=NULL)

{

temp1=temp1->next;

}

z->next=NULL;

temp1->next=z;

}

display(start1);

}

struct node listthree(struct node \*w)

{

w=w->next;

if(n%2==0)

{

while(w->next!=NULL)

{

v=(struct node \*)malloc(sizeof(struct node));

v->data=w->data;

v->next=NULL;

w=w->next;

w=w->next;

if(start2==NULL)

{

start2=v;

}

else

{

temp2=start2;

while(temp2->next!=NULL)

{

temp2=temp2->next;

}

temp2->next=v;

}

}

temp2=start2;

v=(struct node \*)malloc(sizeof(struct node));

v->data=w->data;

while(temp2->next!=NULL)

{

temp2=temp2->next;

}

v->next=NULL;

temp2->next=v;

}

else

{

while(w!=NULL)

{

v=(struct node \*)malloc(sizeof(struct node));

v->data=w->data;

v->next=NULL;

w=w->next;

w=w->next;

if(start2==NULL)

{

start2=v;

}

else

{

temp2=start2;

while(temp2->next!=NULL)

{

temp2=temp2->next;

}

temp2->next=v;

}

}

}

display(start2);

}

struct node display(struct node \*y)

{

while(y!=NULL)

{

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

y=y->next;

}

printf("NULL");

printf("\n");

}

**Output:**

sat123@ubuntu:~$ gcc daq6.c

sat123@ubuntu:~$ ./a.out

Enter Size:10

Insert Element:11

Insert Element:12

Insert Element:13

Insert Element:14

Insert Element:15

Insert Element:16

Insert Element:17

Insert Element:18

Insert Element:19

Insert Element:20

Linked List 1:

11->12->13->14->15->16->17->18->19->20->NULL

Linked List 2:

11->13->15->17->19->NULL

Linked List 3:

12->14->16->18->20->NULL

sat123@ubuntu:~$