DS LAB-APPLICATIONS OF LINKED LIST-1

1BM19CS079

LIKITHA B

Addition of 2 long positive integers-

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
#include<string.h>
struct NODE
{
int info;
struct NODE*link;
};
typedef struct NODE*node;
node getnode()
{
node x;
x=(node)malloc(sizeof(struct NODE));
if(x==NULL)
{
printf("out of mamory\n");
```

```
exit(0);
}
return x;
}
node ins_front(node first,int item)
{
node temp;
temp=getnode();
temp->info=item;
temp->link=first;
return temp;
}
node extract(char *s,node head)
{
 int i,n;
 for(i=0;i<strlen(s);i++)
 {
 n=s[i]-'0';
 head=ins_front(head,n);
 }
 return head;
}
```

```
node addlong(node head1,node head2,node head3)
{
 int temp,sum,carry=0;
 node cur1,cur2;
 cur1=head1;
 cur2=head2;
 while(cur1!=NULL&&cur2!=NULL)
 {
 temp=cur1->info+cur2->info+carry;
 if(temp>9)
 {
 sum=temp%10;
 carry=temp/10;
 }
 else
 {
 sum=temp;
 carry=0;
 }
 head3=ins_front(head3,sum);
 cur1=cur1->link;
 cur2=cur2->link;
```

```
}
while(cur1!=NULL)
{
temp=cur1->info+carry;
if(temp>9)
{
sum=temp%10;
carry=temp/10;
}
else
{
sum=temp;
carry=0;
}
head3=ins_front(head3,sum);
cur1=cur1->link;
}
while(cur2!=NULL)
{
temp=cur2->info+carry;
if(temp>9)
{
```

```
sum=temp%10;
carry=temp/10;
}
else
{
sum=temp;
carry=0;
}
head3=ins_front(head3,sum);
cur2=cur2->link;
}
if(cur1==NULL&&cur2==NULL)
{
if(carry==1)
head3=ins_front(head3,carry);
}
return head3;
}
void display(node first)
{
node cur;
if(first==NULL)
```

```
{
printf("Empty\n");
return;
}
cur=first;
while(cur!=NULL)
{
printf("%d\t",cur->info);
cur=cur->link;
}
}
int main()
{
 int ch;
 node head1=NULL;
 node head2=NULL;
 node head3=NULL;
 char s1[30],s2[30];
 printf("\nEnter first integer\n");
 scanf("%s",s1);
 head1=extract(s1,head1);
 display(head1);
```

```
 printf("\nEnter second integer\n");
 scanf("%s",s2);
 head2=extract(s2,head2);
 display(head2);
 head3=addlong(head1,head2,head3);
 printf("\nThe result is\n");
 display(head3);
return 0;
}
```