

Name: M.S.SANJAY

Registration No.:15BCE0517

Title: ARITHMETIC OPERATIONS ON LARGE NUMBERSS

Course Name: Data Structures & Algorithms

Course Code: CSE2003

Semester: Winter 2015-2016

***Problem Statement :***

The aim of the project is that given any two large numbers (of any length) it should be able to find the sum difference and product of those two numbers using the concept of doubly linked list

***Software Used:***

CodeBlocks version 16.01

***Functions / Modules:***

void get()

This function is used to accept any two large numbers.It first accepts the number of digits and then accepts all digits one by one.The numbers are stored using a doubly linked list.

Time complexity: *O*(*n*)

node\* add(node \*topp1,node \*topp2)

This function performs the addition of two numbers stored in doubly linked lists namely topp1 and topp2.The pre requisite condition is that the length of number represented by topp1 should be greater than that of topp2.The function adds the numbers using the standard addition logic i.e. adding them digit by digit and carrying the overflow digit. It returns the pointer to the sum of the input numbers

Time complexity: *O*(*n*)

node\* sub(node \*top1,node \*top2)

This function performs the difference of two numbers stored in doubly linked lists namely top1 and top2.The pre requisite condition is that the length of number represented by top1 should be greater than that of top2.The function subtracts the numbers using the standard subtraction logic i.e. subtracting them digit by digit and carrying the borrow. It returns the pointer to the difference of the input numbers.

Time complexity: *O*(*n*)

void multipli(node \*topp1,node \*topp2)

This function performs the multiplication of two numbers stored in doubly linked lists namely topp1 and topp2.The pre requisite condition is that the length of number represented by topp1 should be greater than that of topp2.The function multiplies the numbers using the standard multiplication logic (grade school algorithm)i.e.multiply each digit of 2nd number by first number and then add all of them .This function doesn’t return any pointer but rather prints the result then and there.

Time complexity:*O*(*n*2)

int main()

This function is used to call all the above functions. It returns 0.

***List of Errors Encountered while coding the project:***

There were a lot of small syntax and semantics error(like missing semicolon,missing brackets,using undefined variables,etc.) while doing the project. But they were corrected easily. The first major encountered while doing the project was logic error. Initially the carry part was ignored and the result was awkward. Then ,the logic was changed to include the carry and borrow. When doing multiplication, the final answer was random because the zeroes that should be appended to the end was not done. Later the code was modified to include this.Even after this there was error in traversing the doubly linked list. The code was actually written for traversing each digit from ones position to higher position.But the code was doing the reverse traversing.Later this was changed and the correct output was obtained.

***Key or challenging logic in the project:***

First the addition and subtraction was done using singly linked list but the whole logic was changed for multiplication as while doing addition and subtraction in singly linked lists,the final result will not be stored whereas multiplication algoritm required the answer to be stored.So for using the same addition function in multiplication,the whole logic was changed to doubly linked lists.Multiplication could have been done using singly linked list but the complexity would have been much higher as it would require separate modules for addition and it would increase complexity of the project.

***Prerequisites to be included or installed:***

Codeblocks IDE

***Complete Code***

#include <iostream>

using namespace std;

struct node

{

int data;

node \*next,\*prev;

}\*ptr2=NULL,\*ptr1=NULL,\*temp=NULL,\*top1=NULL,\*top2=NULL,\*ptr=NULL,\*anss=NULL,\*res=NULL;

int l1,l2;node \*bt=NULL;

node \*prod=new node;

node\* add(node \*topp1,node \*topp2)

{ node \*ans=NULL;

int c=0;

ptr2=topp2;

ptr1=topp1;

if (top1==NULL&&top2==NULL)

cout<<endl<<"both null na\n";

while(ptr2!=NULL)

{

temp=new node;

temp->data=(ptr1->data+ptr2->data+c)%10;

c=(ptr1->data+ptr2->data+c)/10;

temp->next=NULL;temp->prev=NULL;

if (ans==NULL)

{

ans=temp;

}

else

{

ans->next=temp;temp->prev=ans;

ans=temp;

}

ptr1=ptr1->next;

ptr2=ptr2->next;

}

while(ptr1!=NULL)

{

temp=new node;

temp->data=(ptr1->data+c)%10;

c=(ptr1->data+c)/10;

temp->next=NULL;temp->prev=NULL;

ptr1=ptr1->next;

ans->next=temp;temp->prev=ans;

ans=temp;

}

if (c)

{

temp=new node;

temp->data=c;

ans->next=temp;temp->prev=ans;

ans=temp;

}

while(ans->prev!=NULL)

{

ans=ans->prev;

}

return ans;

}

void get()

{int i;

cout<<"Enter 1st number length";

cin>>l1;

cout<<"ENTER 1st number";

for(i=0;i<l1;i++)

{

temp=new node;

cin>>temp->data;

temp->next=NULL;temp->prev=NULL;

if (top1==NULL)

top1=temp;

else

{

temp->next=top1;top1->prev=temp;

top1=temp;

}

}

cout<<"Enter second number length";

cin>>l2;

cout<<"Enter second number";

for(i=0;i<l2;i++)

{

temp=new node;

cin>>temp->data;

temp->next=NULL;temp->prev=NULL;

if (top2==NULL)

top2=temp;

else

{

temp->next=top2;top2->prev=temp;

top2=temp;

}

}

}

node\* sub(node \*top1,node \*top2)

{int c=0;

ptr2=top2;

ptr1=top1;

while(ptr2!=NULL)

{

temp=new node;

if (((ptr1->data)-(ptr2->data)-c)<0)

{temp->data=10+(ptr1->data-ptr2->data-c);

c=1;

}

else

{temp->data=(ptr1->data-ptr2->data-c);

c=0;

}

temp->next=NULL;temp->prev=NULL;

if (anss==NULL)

{

anss=temp;

}

else

{

anss->next=temp;temp->prev=anss;

anss=temp;

}

ptr1=ptr1->next;

ptr2=ptr2->next;

}

while(ptr1!=NULL)

{

temp=new node;

temp->prev=NULL;temp->next=NULL;

if (((ptr1->data)-c)<0)

{temp->data=10+(ptr1->data-c);

c=1;

}

else

{temp->data=(ptr1->data-c);

c=0;

}

ptr1=ptr1->next;

anss->next=temp;temp->prev=anss;

anss=temp;

}

while(anss->prev!=NULL)

anss=anss->prev;

return anss;

}

void multipli(node \*topp1,node \*topp2)

{

int k=0;int c;

//node \*prod=NULL,\*res;

node \*ptrmul=topp2;

while(ptrmul!=NULL)

{int c=0;

res=NULL;

node \*trav=topp1;

for(int i=0;i<k;i++)

{

temp=new node;

temp->data=0;

temp->next=NULL;temp->prev=NULL;

if(res!=NULL)

{res->next=temp;temp->prev=res;

res=temp;

}

else

res=temp;

}

while(trav!=NULL)

{

temp=new node;

temp->data=((ptrmul->data\*trav->data)+c)%10;

c=((ptrmul->data\*trav->data)+c)/10;

temp->next=NULL;temp->prev=NULL;

if(res==NULL)

res=temp;

else

{

res->next=temp;temp->prev=res;

res=temp;

}

trav=trav->next;

}

if(c)

{

temp=new node;

temp->data=c;

res->next=temp;temp->prev=res;

res=temp;

}

while(res->prev!=NULL)

{ res=res->prev;

}

k++;

prod=add(res,prod);

ptrmul=ptrmul->next;

}

while(prod->next!=NULL)

prod=prod->next;

while(prod!=NULL)

{

cout<<prod->data;

prod=prod->prev;

}

}

node\* divi(node \*top1,node \*top2)

{

node \*ansd=new node;

ansd->data=0;

ansd->next=NULL;ansd->prev=NULL;

node \*one=new node;

one->data=1;

one->next=NULL;one->prev=NULL;

int len1=0,len2=0;

ptr1=top1;

ptr2=top2;

while(ptr1->next!=NULL)

{

len1++;ptr1=ptr1->next;

}

while(ptr2->next!=NULL)

{

len2++;ptr2=ptr2->next;

}

if (l1<l2)

return ansd;

else if (l1==l2)

{

if (ptr1->data<ptr2->data);

}

}

int main()

{node \*wanser=NULL;

get();

prod->data=0;

prod->next=NULL;

prod->prev=NULL;

wanser=sub(top1,top2);

cout<<"\nsubtraction\n";

while(wanser->next!=NULL)

wanser=wanser->next;

while(wanser!=NULL)

{

cout<<wanser->data;

wanser=wanser->prev;

}

cout<<"\nmultiplication\n";

multipli(top1,top2);

cout<<"\naddition\n";

wanser=add(top1,top2);

while(wanser->next!=NULL)

wanser=wanser->next;

while(wanser!=NULL)

{

cout<<wanser->data;wanser=wanser->prev;

}

return 0;

}

***Sample Input and Output***

