**Binary operations using doubly linked list..**

#include <iostream>

using namespace std;

struct node{

int bit;

node \*prev,\*next;

};

class binary{

public:

node \*hn,\*ln;

node \*create(node \*head,int bits){

node \*p,\*q;

cout<<endl<<"Enter binary number:";

for(int i=0;i<bits;i++){

p=new node;

cin>>p->bit;

p->next=p->prev=NULL;

if(head==NULL){

head=p;

head->next=head->prev=NULL;

}else{

q=head;

while(q->next!=NULL){

q=q->next;

}

q->next=p;

p->prev=q;

p->next=NULL;

}

}

hn=head;

ln=p;

return head;

}

void ones(){

node \*p=NULL;

int bits;

cout<<endl<<"Enter count of bits:";

cin>>bits;

p=create(p,bits);

p=hn;

cout<<endl<<"Ones complement:";

while(p!=NULL){

if(p->bit==0){

cout<<"1";

p=p->next;

}else if(p->bit==1){

cout<<"0";

p=p->next;

}

}

}

void twos(){

node \*p=NULL;

int bits;

cout<<endl<<"Enter count of bits:";

cin>>bits;

p=create(p,bits);

p=ln;

while(p!=NULL){

if(p->bit==1){

p->bit=1;

p=p->prev;

while(p!=NULL){

if(p->bit==0){

p->bit=1;

p=p->prev;

}else if(p->bit==1){

p->bit=0;

p=p->prev;

}

}

break;

}else if(p->bit==0){

p->bit=0;

p=p->prev;

}

}

p=hn;

cout<<endl<<"Twos complement:";

while(p!=NULL){

cout<<p->bit;

p=p->next;

}

}

void addition(){

node \*h1,\*h2;

h1=NULL;

h2=NULL;

int bits=0,carry=0,A[20],p=0;

cout<<endl<<"Enter count of bits:";

cin>>bits;

h1=create(h1,bits);

h1=ln;

h2=create(h2,bits);

h2=ln;

while(h1!=NULL&&h2!=NULL)

{

if(h1->bit==0&&h2->bit==0)

{

A[p]=carry;

p++;

carry=0;

}

else if(h1->bit==1&&h2->bit==1)

{

A[p]=carry;

p++;

carry=1;

}

else if(h1->bit!=h2->bit)

{

if(carry==1)

{

A[p]=0;

p++;

carry=1;

}

else

{

A[p]=1;

p++;

}

}

h1=h1->prev;

h2=h2->prev;

}

if(carry==1){

A[p]=carry;

p++;

}

cout<<endl<<"Addition is:";

for(int i=p-1;i>=0;i--){

cout<<A[i]<<"\t";

}

}

};

int main()

{

binary b1;

int ch;

do{

cout<<endl<<"1.ones complement...\n2.Twos complement...\n3.Addition...";

cout<<endl<<"Enter your choice:";

cin>>ch;

switch(ch){

case 1:

b1.ones();

break;

case 2:

b1.twos();

break;

case 3:

b1.addition();

break;

}

}while(ch!=5);

return 0;

}