```
// example 2.3.a//
clc
clear
format('v',18);
//bb=input ('enter the first number (in
decimal):')//
// aaa=input('enter the second number
(negative):') ;
aaa =-118
bb =32;
aa = -1* aaa ;
a =0;
q =0;
while (aa >0)
    //finding the binary equivalents//
x= modulo (aa ,2) ;
a= a + (10^ q)*x;
aa=aa /2;
aa= floor (aa);
q=q+1;
end
r =0;
b =0;
while (bb >0)
x= modulo (bb ,2) ;
b= b + (10^ r)*x;
bb=bb /2;
bb= floor (bb);
r=r+1;
end
m=b
for i =1:16
a1(i)= modulo (a ,10) ;
a=a /10;
a= round (a);
p1(i) =0;
b1(i)= modulo (b ,10) ;
b=b /10;
```

```
b= round (b);
end
p1 (1) =1;
for i =1:16
    //finding the 2's compliment of second
number//
a1(i)= bitcmp (a1(i) ,1);
end
car (1) =0;
for i =1:16
c1(i)= car (i)+a1(i)+ p1(i);
if c1(i)== 2 then
car (i +1) = 1;
c1(i) =0;
elseif c1(i)==3 then
car (i +1)= 1;
c1(i) =1;
else
car (i +1) =0;
end ;
end ;
re =0;
for i =1:16
re=re +( c1(i) *(10^(i -1) ))
end ;
printf ( ' The binary representation of first
number is ' );
disp (m);
printf ('The 2''s compliment of second number is
' );
disp (re);
a1=c1;
ar (1) =0;
for i =1:8
c1(i)=ar(i)+a1(i)+ b1(i);
// addin both the nmbers ( binary addition )
if c1(i)== 2 then
    // lower byte
```

```
ar(i+1)= 1;
c1(i) =0;
elseif c1(i)==3 then
ar(i+1)= 1;
c1(i) =1;
else
ar(i+1) =0;
end
end
c1 (9)=ar (9)
re =0;
format('v',18);
for i =1:8
re=re +( c1(i) *(10^(i -1) ))
end
printf ( ' The sum of lower bytes of two binary numbers is %d' ,re );
printf ( ' with a carry is %d' ,ar (9));
for i =9:16
c1(i)=ar(i)+a1(i)+ b1(i);
// upper byte//
if c1(i)== 2 then
ar(i+1)= 1;
c1(i) =0;
elseif c1(i)==3 then
ar(i+1)= 1;
c1(i) =1;
else
ar(i+1) =0;
end
end
c1 (17) =ar (17) ;
format ('v',25);
ree =0;
for i =9:16
ree = ree +( c1(i) *(10^(i -9) ));
end
for i =9:16
```

```
re=re +( c1(i) *(10^(i -1) ))
end
printf ( ' The sum of upper bytes of the given
numbers is %d' ,ree);
printf ( ' with a carry is %d ' ,ar (17) );
//displaying results//
printf (' The total sum is ' );
disp (re);
printf ( ' with a carry %d ' ,ar (17) );
disp('when we were using 8 bit we were getting
error as number crosses its limit. While in 16
bit we get our result in 2''s complement form
which comes out to be -150')
```