unsigned int used all 32 bits to store the value and the MSB.(=1) will make the value.

An unsigned int does not use the 2's compliment again to display the number.

Thus, 11...10010000. gets printed as it is in decimal

Therefore,

```
unsigned int a=-112;
cout << a << endl;
```

output!
4294967184

Operators

Basic Arithmetic operators!

+, -, x, /, %

Caution ⚠

① int / int = int (floor value of answer)

Examples:

5/2 = 1

3/5 = 0

9/2 = 4

② int / float
float / int } float

double / int
int / double } double

Relational operators:

==, >=, <=, >, <, !=

① Is a=b ?

a==b $\xrightarrow{yes}$ 1

$\xrightarrow{No}$ 0

```
int a = 2;
int b = 3;
bool first = (a==b);
cout << first << endl;
bool second = (a>b);
cout << second << endl;
bool third = (a<b);
cout << third << endl;
bool fourth = (a>=b);
cout << fourth << endl;
bool fifth = (a<=b);
cout << fifth << endl;
bool sixth = (a!=b);
cout << sixth << endl;
```

output:

```
0
0
1
0
1
1
```

Logical operators:

&&, ||, !

(And), (OR), (Not)

① Logical And:

All conditions must be true for the output to be true.

Example: int a=5, b=10, c=15;

cout << ((a>0) && (b!=0) && (c<=15));

output:

1

② Logical OR:

At least 1 conditions must be true for the output to be true:

Example: int a=5, b=10, c=15

cout << ((a>5) || (b<10) || (c>=15));

output:

1

③ Logical Not!

Inverts the logic. True ⇄ False

Non-zero ⇄ zero

Example!

```
int a=10, b=0;
cout << (!a) << endl;
cout << (!b) << endl;
```

output!

```
0
1
```