Decimal to Binary :

(1) 10 →

$\frac{10}{2} = 5$, $\frac{5}{2} = 2$, $\frac{2}{2} = 1$, $\frac{1}{2} = 0$

(Remainder) 0 1 0 1

Finish ← Start

$2^3\ 2^2\ 2^1\ 2^0$

10 → 1010

Steps : i) Divide number by 2
ii) Store the remainder.
iii) Repeat for the dividend.

Division	Remainder
10/2	0
5/2	1
2/2	0
1/2	1

Reverse = 1010

g. n = 7 to Binary

Division	Remainder
7/2	1
3/2	1
1/2	1

Reverse = 111

(2) n = 5 → Binary ?

Logic : n & 1 = 0 if n is even.

n & 1 = 1 if n is odd

Thus, a bit is 1 if bit & 1 = 1
a bit is 0 if bit & 1 = 0

Thus, (any_bit) & 1 = (any_bit)

∴ Last bit of 5 = 5 & 1 = 1

Check 5 = 101 ⇒

```
  101
& 001
  ---
  (1)
```

Right shift 5 by 1 ⇒ n = 2 (010)

Second last bit of 5 = 2 & 1 = 0

Right shift 2 by 1 ⇒ n = 1 (001)

First bit of 5 = 1 & 1 = 1

Right shift 1 by 1 ⇒ n = 0 (000)

Khatam !

```cpp
#include<iostream>
#include<math.h>
using namespace std;


int main() {

    int n;
    cin >> n;


    int ans  = 0;
    int i = 0;
    while(n != 0 ) {

        int bit  = n & 1;

        ans = (bit * pow(10, i) ) + ans;

        n = n >> 1;
        i++;

    }

    cout<<" Answer is " << ans << endl;


}
```

Example : n = 6

① ans = 0 , i = 0

② bit = 6 & 1 = 0 (even)

③ ans = 10^0 x 0 + ans = 0 , n = 3 , i = 1

④ bit = 3 & 1 = 1 (odd)

⑤ ans = (10^1 x 1) + ans = 10 , n = 1 , i = 2

⑥ bit = 1 & 1 = 1 (odd)

⑦ ans = $(10^2 \times 1)$ + ans = 100 + 10 = 110 , n = 0

ans = 110

Negative Decimal to Binary: (Homework)

-6 ⟶ Binary ??

We have discussed the logic about the storage & display of negative numbers earlier.

$$-6 \rightarrow 6 \rightarrow 000\ldots00110 \rightarrow \begin{array}{r} 111\ldots11001 \\ +\quad\quad\quad\quad 1 \\ \hline 111\ldots11010 \end{array}$$

Think what is 111....11010 equal to if it is unsigned?

$\underbrace{111\ldots11}_{29\text{ bits}}010 = 4294967290$

$= 2^{32} - 6$

$\therefore -6 \xrightarrow{\text{Binary}}$ Binary of $(2^{32} - 6) \rightarrow \underbrace{111\ldots11}_{29}010$

We can't express this in any data type.

Assuming we have 2 byte (= 16 bit) integers:

(Might need online IDE)

```cpp
#include <iostream>
#include <math.h>
using namespace std;

int main(void)
{
    long long int n;
    cin >> n;
    unsigned long long int i = 0, ans = 0;
    if(n < 0) {
        n = pow(2, 16) + n;
    }
    cout << n << endl;
    while(n) {
        int lastBit = n & 1;
        ans = (pow(10, i) * lastBit) + ans;
        n = n >> 1;
        i++;
        cout << ans << endl;
    }
    cout << ans << endl;
    return 0;
}
```

Binary to Decimal:

$101011 = 2^0 \times 1 + 2^1 \times 1 + 2^2 \times 0 +$

$2^5\ 2^4\ 2^3\ 2^2\ 2^1\ 2^0 \quad 2^3 \times 1 + 2^4 \times 0 + 2^5 \times 1$

$= 1 + 2 + 0 + 8 + 0 + 32$

$= \boxed{43}$

Logic: $n \& 1 = \begin{cases} 0, & \text{don't do anything} \\ 1, & \text{multiply with } 2^i \end{cases}$

$i++, \quad n >> 1.$

We will give int n = 10010; but it is actually a decimal number so last bit = n%10;

```
#include<iostream>
#include<math.h>
using namespace std;


int main() {

    int n;
    cin >> n;

    int i = 0, ans = 0 ;

    while( n != 0) {

        int digit = n % 10;

        if( digit == 1) {
            ans = ans + pow(2, i);
        }

        n = n/10;
        i++;

    }
    cout<< ans << endl;


}
```