



Decimal to Binary :

① $10 \rightarrow$

$$\begin{array}{cccc} \frac{10}{2} = 5, & \frac{5}{2} = 2, & \frac{2}{2} = 1, & \frac{1}{2} = 0 \\ \downarrow & \downarrow & \downarrow & \downarrow \\ \text{(Remainder)} & 0 & 1 & 0 & 1 \\ & \text{Finish} & \leftarrow & \text{Start} \end{array}$$

$10 \rightarrow 2^3 2^2 2^1 2^0$
 $10 \rightarrow 1010$

- Steps :
- Divide number by 2
 - Store the remainder.
 - 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 $n = 5 \rightarrow$ Binary ?

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

$n \& 1 = 1$ if n is odd

Thus, a bit is 1 if $\text{bit} \& 1 = 1$
a bit is 0 if $\text{bit} \& 1 = 0$

Thus, $(\text{any_bit}) \& 1 = (\text{any_bit})$

\therefore Last bit of 5 = $5 \& 1 = 1$

Check $5 = 101 \Rightarrow$

$$\begin{array}{r} 101 \\ \& \underline{1} \\ \hline 1 \end{array}$$

Right shift 5 by 1 $\Rightarrow n = 2$ (010)

Second last bit of 5 = $2 \& 1 = 0$

Right shift 2 by 1 $\Rightarrow n = 1$ (001)

First bit of 5 = $1 \& 1 = 1$

Right shift 1 by 1 $\Rightarrow n = 0$ (000)

Khatam!

```
1 #include<iostream>
2 #include<math.h>
3 using namespace std;
4
5
6 int main() {
7
8     int n;
9     cin >> n;
10
11
12     int ans = 0;
13     int i = 0;
14     while(n != 0) {
15
16         int bit = n & 1;
17
18         ans = (bit * pow(10, i)) + ans;
19
20         n = n >> 1;
21         i++;
22     }
23
24
25     cout<<" Answer is " << ans << endl;
26
27
28 }
```

Example : $n = 6$

① $\text{ans} = 0, i = 0$

② $\text{bit} = 6 \& 1 = 0$ (even)

③ $\text{ans} = 10^0 \times 0 + \text{ans} = 0, n = 3, i = 1$

④ $\text{bit} = 3 \& 1 = 1$ (odd)

⑤ $\text{ans} = (10^1 \times 1) + \text{ans} = 10, n = 1, i = 2$

⑥ $\text{bit} = 1 \& 1 = 1$ (odd)

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

$$\text{ans} = 110$$

Negative Decimal to Binary: (Homework)

-6 \rightarrow Binary ??

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

$$\begin{array}{r} -6 \rightarrow 6 \rightarrow 000\dots00110 \rightarrow 111\dots11001 \\ + 1 \\ \hline 111\dots11010 \end{array}$$

Think what is $111\dots11010$ equal to if it is unsigned?

$$\begin{array}{l} \underbrace{111\dots11010}_{29 \text{ bits}} = 4294967290 \\ = 2^{32} - 6 \end{array}$$

$$\therefore -6 \xrightarrow{\text{Binary}} \text{Binary of } (2^{32} - 6) \rightarrow \underbrace{111\dots11010}_{29}$$

We can't express this in any data type.

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

```
1  #include <iostream>
2  #include <math.h>
3  using namespace std;
4
5  int main(void)
6  {
7      long long int n;
8      cin >> n;
9      unsigned long long int i = 0, ans = 0;
10     if(n < 0) {
11         n = pow(2, 16) + n;
12     }
13     cout << n << endl;
14     while(n) {
15         int lastBit = n & 1;
16         ans = (pow(10, i) * lastBit) + ans;
17         n = n >> 1;
18         i++;
19         cout << ans << endl;
20     }
21     cout << ans << endl;
22     return 0;
23 }
```

(Might need)
(online IDE)

Binary to Decimal:

$$\begin{aligned}
 101011 &= 2^0 \times 1 + 2^1 \times 1 + 2^2 \times 0 + \\
 &\quad 2^3 \times 1 + 2^4 \times 0 + 2^5 \times 1 \\
 &= 1 + 2 + 0 + 8 + 0 + 32 \\
 &= \boxed{43}
 \end{aligned}$$

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

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

```

1  #include<iostream>
2  #include<math.h>
3  using namespace std;
4
5
6  int main() {
7
8      int n;
9      cin >> n;
10
11     int i = 0, ans = 0 ;
12
13     while( n != 0 ) {
14
15         int digit = n % 10;
16
17         if( digit == 1 ) {
18             ans = ans + pow(2, i);
19         }
20
21         n = n/10;
22         i++;
23     }
24     cout<< ans << endl;
25
26
27
28 }

```