Algorithm

=>
$$110_2 = (1*2^2) + (1*2^1) + (0*2^0)$$

=> $110_2 = 4 + 2 + 0$
=> $110_2 = 6$

- 1. Multiply each digit separately from left to right by 2^0, 2^1, 2^2... respectively.
- Add all the results coming from step
 1.
- Equivalent decimal number would be the result obtained in step 2.

Code

```
binaryToDecimal(int n)
int num = n;
int dec value = 0;
int base = 1;
int temp = num;
while (temp) {
    int last digit = temp % 10;
    temp = temp / 10;
    dec value += last digit * base;
    base = base * 2;
return dec value;
```

Dry Run

```
num = 110 dec_value = 0
base = 1 temp = 110
```

```
last_dig = 0 last_dig = 1 last_dig = 1
temp = 11 temp = 0
dec_val = 0 dec_val = 2 dec_val = 6
base = 2 base = 4 base = 8
```

Input: 110 Output: 6

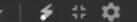
Code

```
int binaryToDecimal(int n)
   int num = n;
   int dec value = 0;
   int base = 1;
   int temp = num;
    while (temp) {
        int last digit = temp % 10;
        temp = temp / 10;
        dec value += last digit * base;
        base = base * 2;
    return dec value;
```

Dry Run

```
num = 110 dec_value = 0
base = 1 temp = 110
```

Input: 110 Output: 6





Thank you for watching!

Please leave us your comments.



Waiting for docs.google.com...

Google Slides