Date 03/10/2023
Page

Code for Decimal to Binary And Vice-Versa.

Decimal	Binary
0	0
1	1
2	10
3	11
4	100
5	101
⋮	⋮

Base	Quotient	Rem
2	13	
2	6	1
2	3	0
2	1	1
	0	1 ↑

Decimal 13 → Binary 1101

Total no. of unique digit in a number system is known as base of that number system.

Binary
0, 1 → base 2

e.g.

a = 2, b = 5, c = 8

258

Ans = 2 × 10 + b
= 20 + 5 = 25
= 25 × 10 + 8
= 250 + 8 = 258

i.e

258

$2\times10^2 + 5\times10^1 + 8\times10^0$

200 + 50 + 8 = 258.

e.g (11)

3 6 2 4 5 ← num

3

Ans: → $3 \times 10 + 6 = 36$

Ans → $36 \times 10 + 2 = 362$

: $362 \times 10 + 4 = 3624$

$3624 \times 10 + 5 = 36245$

int Ans = 0;
Ans = Ans × 10 + num

★ if $a = 6, b = 9$, make 96.

$b \times 10 + a$

96

6492 → 2946

$2 \times 10^3 + 9 \times 10^2 + 4 \times 10^1 + 6 \times 10^0$

num →

$6 \times 1 + 0 \Rightarrow 6$ ← Ans

$4 \times 10 + 6 \Rightarrow 46$ ← Ans

$9 \times 100 + 46 \Rightarrow 946$ ← Ans

$2 \times 1000 + 946 \Rightarrow 2946$ ← Ans

Ans = 0
Ans = num × 10^i + ans

⇒

	num	Rem
2	13	
	6	1
	3	0
	1	1
	0	1

1101

Ans = 0

Ans = rem × 10^i + ans

$= 1 \times 10^0 + 0 = 1$

$= 0 \times 10^1 + 1 = 01$

$= 1 \times 10^2 + 1 = 101$

$1 \times 10^3 + 101 = 1101$

```
int n =13
int rem, ans=0, mul=1;
while (n>0)
{
    rem= n%2;
    n = n/2;
    ans = rem x mul + ans
    mul = mux10;
}
cout << ans;
```

refer code: ../DecimalToBinary.cpp

Debug/Dry Run

n=13   13>0 True
Ans=0
mul=1

(i) rem=13%2=1
ans=1x1+0=1

(ii) n=6   6>0 True
ans=1
mul=10
rem=6%2=0
ans=0x10+1=1

(iii) n=3   3>0 True
ans=1
mul=100
rem: 3%2=1
ans=1x100+1=101

(iv) n=1   1>0 True
ans=101
mul=1000
rem=1%2=1
ans=1x1000+101
=1101

(v) n=0   0>0 False
n>0
↑
false
↑
loop break

Ans = 1101

Binary To Decimal

101

$\hookrightarrow 1\times 2^2 + 0\times 2^1 + 1\times 2^0$

~~100~~

$4 + 0 + 1 = 5$

1101

$\hookrightarrow 1\times 2^3 + 1\times 2^2 + 0 + 2^1 + 1\times 2^0$

$8 + 4 + 0 + 1 = 13$

$ans = re~~m~~m \times 2^i + ans$

10	1101	rem
10	110	1
10	11	0
10	1	1
10	0	1

rem = num%10 ;

num = num/10 ;

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

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

$1\times 2^2 + 1 = 5$

$1\times 2^3 + 5 = 13$

```
int num = 1101 ;
int  ans = 0, rem, mul = 1;
while ( num > 0 )
{
    rem = num%10 ;
    num = num/10 ;
    ans = rem x mul + ans
    mum *= 2 ;
}
cout << ans ;
```

Refercode: ../BinaryToDecimal.cpp

Dry Run

(i) n = 1101

1101 > 0

True

rem = 1101%10

= 1

num = 110

ans = 1×1 + 0

= 1

mul = 2

(ii) n = 110

110 > 0

True

rem = 110%10

= 0

num = 11

ans = 0×2 + 1

= 1

mul = 4

(iii) n = 11

11 > 0

True

rem = 11%10 = 1

num = 1

ans = 1×4 + 1 = 5

mul = 8

(iv) n = 1

1 > 0

True

rem = 1%10 = 1

~~num = 0~~

ans = 1×8 + 5

= $\boxed{13}$

mul = 16

(v) n = 0

0 > 0

false.

Decimal to octa

8	13		
8	1	5	↑
	0	1	

$(13)_{10} \Rightarrow (15)_8$

octa to Decimal

$(15)_8$

$1 \times 8^1 + 5 \times 8^0$

$8 + 5 = (13)_{10}$

Binary To octa

$(1101)_2 \longrightarrow (\quad)_8$

↳ $(1101)_2 \longrightarrow (\quad)_{10}$

$(\quad)_{10} \longrightarrow (\quad)_8$

octa to Binary

$$(\quad)_8 \longrightarrow (\quad)_2$$

$$\begin{tikzcd}
(\quad)_8 \arrow[r] & (\quad)_{10} \arrow[dl] \\
(\quad)_{10} \arrow[r] & (\quad)_2
\end{tikzcd}$$