Decimal Number System:

Base 10 → A single digit can take 10 values → 0 to 9.

Position of each digit in a decimal number represents a different power of 10.

Eg: $342 = 3 \times 100 + 4 \times 10 + 2 \times 1 = 3 \times 10^2 + 4 \times 10^1 + 2 \times 10^0$

$2563 = 2 \times 1000 + 5 \times 100 + 6 \times 10 + 3 \times 1$

$= 2 \times 10^3 + 5 \times 10^2 + 6 \times 10^1 + 3 \times 10^0$

Binary Number System:

- Used in digital electronics and computing.
- Two digits 0 & 1.

$110_{(2)} \rightarrow 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = \underline{6}$

$1011_{(2)} \rightarrow 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = \underline{11}$

Binary - Decimal Conversion:

1. Convert the binary number 1101 to a decimal number.

$$\overset{3\ 2\ 1\ 0}{1101} \rightarrow 1\times2^3 + 1\times2^2 + 0 + 1\times2^0 = \underline{13}$$

2. Convert $10101_{(2)}$ to a decimal number.

$$\overset{4\ 3\ 2\ 1\ 0}{10101} = 1\times2^4 + 1\times2^2 + 1\times2^0 = \underline{21}$$

$$\overset{6\ 5\ 4\ 3\ 2\ 1\ 0}{1011010} = \underline{90}$$

Decimal - Binary Conversion:

2	45
2	22 - 1
2	11 - 0
2	5 - 1
2	2 - 1
2	1 - 0
	0 - 1

$45 = 101101_{(2)}$

2	130
2	65 - 0
2	32 - 1
2	16 - 0
2	8 - 0
2	4 - 0
2	2 - 0
2	1 - 0
	0 - 1

$130 = 10000010_{(2)}$

Addition Of Decimal Numbers:

```
  2  1  0
  3  6  8
  4  5  3
 ---------
  8  2  1
 ---------
```

$8 + 3 = 11$

$11 \div 10 = 1$ =.

$\frac{11}{10} = 1$

Addition Of Binary Numbers:

```
   1 1   1
   1 0 1 0 1
     1 1 0 1
 -------------
   1 0 0 0 1 0
 -------------
```

```
   1 1 0 1 0 1
   1 0 0 1 1 0
 ---------------
 1 0 1 1 0 1 1
 ---------------
```

Bitwise Operators:

Bit → 0 or 1.

OFF → 0

ON → 1

1 set 0 unset

13 + 15

1101 + 1111

```
  1101
  1111
 11100
```

28

And operation (&)

0 & 0 → 0

0 & 1 → 0

1 & 1 → 1

1 & 0 & 1 = 0

1 & 1 & 1 = 1

1 if only all bits are set.

OR (|)

0 | 0 → 0

1 | 1 → 1

1 | 0 → 1

1 if any of the bits are set.

NOT (!)

!0 → 1 !1 → 0

XOR (^)

Exclusive OR

1 ^ 1 → 0

1 ^ 0 → 1

0 ^ 1 → 1

0 ^ 0 → 0

```
  0011
  0011
  0000
```

```
  0001
  0001
  0000
```

x ^ 0 → x

x ^ x →

```
  1011
  0000
  1011
```

[1, 3, 7, 5, 1, 3, 5]

```
  010100
  101101
 1111001  → 57
```

0001 → 1

0011 → 3

0010

Break - 8:12 AM

Binary Representation Of Negative Numbers:

7 6 5 4 3 2 1 0

int → 32

0000 0101

1st step flip → 1111 1010 — (1)

2nd step. 0000 0001

(1) + 1 1111 1011 → 2's complement

1111 1011
0000 0101
0000 0000

0000 0011 → 1111 1100 → 1111 1101 → -3

8 bits signed integer

0000 1010 → 1111 0101

↳ 1111 0110

2^7 → 10000000

8 bits → signed int → $2^{8-1} - 1$ -2^7

32 bits → signed int → $2^{32-1} - 1$ -2^{31}

64 bits → signed int → $2^{63} - 1$ -2^{63}

N " → " " → $2^{N-1} - 1$ -2^{N-1}

N : $-2^{N-1} \rightarrow 2^{N-1} - 1$

Range Of Data Types:

8 bits	→ signed int	→ $2^{8-1}-1$	-2^7
32 bits	→ signed int	→ $2^{32-1}-1$	-2^{31}
64 bits	→ signed int	→ $2^{63}-1$	-2^{63}
N "	→ " "	→ $2^{N-1}-1$	-2^{N-1}

N : $-2^{N-1} \rightarrow 2^{N-1}-1$

Importance Of Constraints:

$0 \leq A[i] \leq 10^5$

$1 \leq A.size \leq 10^6$

} long:

1. TLE or not
2. Data Types to be used.

int → -2^{31} to $2^{31}-1$

Long → 64

int → signed.

$2^{10} \rightarrow 1024$ (CI 10^3)

$2^{30} - (2^{10})^3 = (10^3)^3 = 10^9$