Data Types & Typecasting

6 → 0115

① number System

(i) Binary System → 2 [0 1]
(ii) Dec " → 10 → [0 to 9]
(iii) Octal " → 8 [0 to 7]

$18_{(10)} = 10010_{(2)}$

$15 \div 10 = 5$
$1 \div 10 = 1$
$9 \div 2 = 9$
$3 \div 10 = 0$
$10 \div 2 = 5$
$10 \div 2 = 5$
$1 \div 2 = 0$
$1 \div 2 = 1$

$6 \div 2 = 0$
$8 \div 2 = 1$
$8 \div 2 = 1$

$$\begin{array}{r} 1\,0 \\ +\ 2\,4\,5\,3\,7 \\ 2\,4\,1\,3\,7\ + \\ \hline 5\,1\,0\,0 \end{array}$$

$$\begin{array}{r} 1\,0\,1 \\ 1\,3\,4\,5\,7 \\ 2\,6\,3\,8 \\ \hline 1\,1\,0\,9\,5 \end{array}$$

$$\begin{array}{r} 0\,0\,1 \\ 0\,1\,1\,0\,1 \\ 0\,1\,0\,0\,1 \\ \hline 1\,0\,1\,1\,1\,0 \end{array}$$

$3 \div 2 = 0$
$2 \div 2 = 1$
$3 \div 2 = 1$
$3 \div 2 = 1$
$1 \div 2 = 0$

$3745 = 3 \times 10^3 + 7 \times 10^2 + 4 \times 10^1 + 5 \times 10^0$

N = 37

$100101 = 1 \times 10^5 + 0 \times 10^4 + 0 \times 10^3 + 1 \times 10^2 + 0 \times 10^1 + 1 \times 10^0$

while (n>0) {
sum = n%2
sum = sum + sum * mul
mul = mul * 10
n = n/2
}

2	37	1×10^0
2	18	0×10^1
2	9	1×10^2
2	4	0×10^3
2	2	0×10^4
2	1	1×10^5
	0	

mul = 1×10^6

100101

2	37	1×10^0
2	18	0×10^1
2	9	1×10^2
2	4	0×10^3
2	2	0×10^4
2	1	1×10^5
	0	

100000
100
1
100101

Sum = $10^0 + 10^2 + 10^5$
+ 0 + 100000

Str → decimal

while (n>0) {
sum = n%10
sum = sum * mul
mul = mul * 2 * 2
n = n/10
}

10	101011	1×2^0
10	10101	1×2^1
10	1010	0×2^1
10	101	1×2^3
10	10	0×2^4
10	1	1×2^5
	0	

while (n>0) {
sum = n%10
sum = sum * mul
mul = mul * 2
n = n/10
}

① Data - Types

- Primitive
  - byte → 1 byte = 8 bit
  - Short → 2 byte = 16 bit
  - int → 4 bit 32 bit
  - long → 8 byte 64 bit
  - float → 4 byte 32 bit
  - double → 8 byte 64 bit
  - char → 2 byte
  - boolean → 1 bit
- non-primitive → X

⑨

byte b = 5;
Short s = 5
int i = 5;
long l = 5;

⑪ 000 00 101

101 X

byte b = 5;
Short s = 5
int i = 5;
long l = 5;

Signed

0000
1111

$2^8 = 256$

0000 00000
1 0 00 0000

0 1111111
1 0000000

Signed

0 1111111
1 0000000

-127 to 127 | 255

0 1010101 0
0 1010101

Signed

-01010100
1 0101011
0 0000001
1 0101100

-128 to 127

MSB ← 0 1 1 1 1 1 1 1 → LSB
-ve ← 1

1000 0000
0 111 1111
1 111 1111
1000 0000

byte → 8 bit → -2^7 to $2^7 - 1$ → -128 to 127
Short → 16 bit → -2^{15} to $2^{15} - 1$ → 10^4
int → 32 bit → -2^{31} to $2^{31} - 1$ → 10^9
long → 64 bit → -2^{63} to $2^{63} - 1$ → 10^{18}

```
public static void main(String[] args) {
    // TODO Auto-generated method stub
    byte b = 5;
    short s = 5;
    int i = 5;
    long l = 5;
    System.out.println(b);
    System.out.println(s);
    System.out.println(i);
    System.out.println(l);
}
```

00000101

byte b = (byte) (420);

8 bit

1 0 0 1 0 1 0 0

1 0 1 01100
0 1 0 0 1 0 0 1
111
0 1 0 1 0 1 0 0

-84
Story

1 0 0 1 0 1 0 0

0 1 0 1 0 1 0 0 = (+ve)

byte b = -5
int i = b

00 0000101
111 11010
111 11 011
-5

00000101

24

24

a → 97
b → 98
c → 99
d → 100
e → 101
f

A → 65
B → 66
C → 67
D
E
F

'0' → 48
'1' → 49
'2' → 50
'3'

2
3

-128 to 127

for (byte i = 0; i < 128; i++) {
System.out.println(i);
}

026

int a = 032;
2 × 8 0
2 × 8 1
26

i = 0 × 127
0
1
2
3
1
126
127
-128
-127
-126

i = byte (111)
= -128
-127