Welcome ☺

Agenda : Modular arithmetic
%
2 problems.

% basics

$n \% a \rightarrow$ remainder when n is divided by a

r = divident − (greatest multiple of div ≤ divident)

$10 \% 4 \rightarrow 2 \Rightarrow 10 - (4*2) = 2$

$13 \% 5 \rightarrow 3 \Rightarrow 13 - (5*2) = 3$

Dividend = div * quo + remainder

Remainder = divident − $\underbrace{\text{div} * \text{quo}}$

greatest multiple of divisor <= divident

Quiz

$150 \% 11 = 150 - (11*13) = 150 - 143 = 7$

$100 \% 7 = 100 - (7*14) = 100 - 98 = 2$

greatest mul of 7 ≤ −40

$-40 \% 7 = -40 - (-42) = -40 + 42 = 2$

$-60 \% 9 = -60 - (-63) = -60 + 63 = 3$

Python users   Java / C++ / C / NET / JS   // doubt session

$-40 \% 7$ $2 \xleftarrow{+7} -5$

$-60 \% 9$ $3 \xleftarrow{+9} -6$

```
if( a < 0) {
   // a % p
   a % p + p
}
```

Why % $\Rightarrow$ Limits our input data to required range

→ Hashing (in future classes)

→ consistent hashing (LLD_HLD)

$$\left.\begin{matrix}-\infty \\ \\ \infty\end{matrix}\right\} \% 10 = [0, 9]$$

$$\left.\begin{matrix}-\infty \\ \\ \infty\end{matrix}\right\} \% p = [0, p-1]$$

Modular arithmetic (* + - /)

① $(a + b) \% p \rightarrow 0 \rightarrow p-1$

a	b	p
8	6	10

$(8+6) \% 10$

$14 \% 10 = 4$

$(a \% p + b \% p) \% p$

$0 \rightarrow p-1$ $0 \rightarrow p-1$

$p-1 + p-1 = 2p-2 \rightarrow$ out of range. $\rightarrow$ ∴

$$\boxed{(a+b) \% p = (a \% p + b \% p) \% p}$$

P.2

$$\boxed{(a * b) \% p = (a \% p * b \% p) \% p}$$

$[0 \to p-1]$ $[0 \to p-1]$

$(p-1) * (p-1) \simeq p^2$

$p^2 \% p \Rightarrow 0 \to p-1$

a	b	p
8	6	10

P-3 $(a-b) \% p$
P-4 $(a/b) \% p$ } → advance batch, inverse modulo

---

① $(a \% p) \% p = a \% p$

$[0 \to p-1) \% p \Rightarrow [0 \to p-1]$

✓ eg: $(6 \% 10) \% 10$
$= 6 \% 10 = 6$

② $(a \% p * b) \% p = (a * b) \% p$

$x = a \% p$
$y = b$

$(x * y) \% p \Rightarrow (x \% p * y \% p) \% p$

$((a \% p) \% p * b \% p) \% p$

$(a \% p * b \% p) \% p$

$\Rightarrow (a * b) \% p$

dry run
a = 6
b = 4
p = 4

$(6 \% 4 * 4) \% 4$

$(2 * 4) \% 4$

$(2 \% 4 * 4 \% 4) \% 4$

$(2 * 0) \% 4 \Rightarrow 0$

Divisibility Rules

$\%3 \Rightarrow$ sum of digits should be divisible by 3

$\%9 \Rightarrow$ sum of digits should be divisible by 9

$\%4 \Rightarrow$ last two digit should be divisible by 4

$\%8 \Rightarrow$ last 3 digits should be divisible by 8

$\%6$, $\%7$ TODO / H.W.

Proof $\rightarrow \%3$

$$(2475)\%3 \rightarrow (2\times10^3 + 4\times10^2 + 7\times10^1 + 5\times10^0)\%3$$

$$\rightarrow \left((2\times10^3)\%3 + (4\times10^2)\%3 + (7\times10^1)\%3 + (5\times10^0)\%3\right)\%3$$

$$\rightarrow \left((2\%3 * 1)\%3 + 4\%3 + 7\%3 + 5\%3\right)\%3$$

$$\rightarrow \underbrace{(2+4+7+5)}_{\text{sum of all digits.}}\%3$$

// obs

$10^1 \% 3 = 1$

$10^2 \% 3 = 1$

$10^3 \% 3 = 1$

$\vdots$

$10^n \% 3 = 1$

$10^1 \% 9 = 1$

$10^2 \% 9 = 1$

Proof $\%4$

$$(2475)\%4 = (2400 + 75)\%4$$

$$= (2400\%4 + 75\%4)\%4$$

$$= (0 + 75\%4)\%4 = 75\%4$$

obs

$10^1 \% 4 = 2$

$100 \% 4 = 0$

if $100 \% 4 == 0$ then any multiple of 100 will have zero remainder.

$10^3 \% 4 = 0$

Proof $\%8$

$10^2 \% 8 \neq 0$

$10^3 \% 8 = 0$

any multiple of 1000 is also divisible by 8

Q1 Given a, n, p calculate $a^n \% p$, without inbuilt func^n.

constraints $1 \leq a \leq 10^9$ $1 \leq p \leq 10^9$

10^{19}

$1 \leq n \leq 10^5$

eg: $a = 3$ $n = 4$ $p = 7$

$\rightarrow (3^4) \% 7 = 81 \% 7 = 4$

Pseudo

```
int ans = 1
for( int i = 1 ; i ≤ n ; i++)
{
    ans = ans * a
}
return ans % p
```

X wrong
overflow in multiplica^n step

⇓

```
~~int~~ long ans = 1
for( int i = 1 ; i ≤ n ; i++)
{
    ans = (ans * a) % p
}
return ans ~~% p~~
```

$\underbrace{ans}_{[0,p-1]} * \underbrace{a}_{[1,10^9]}$

not req.

$[p-1] * 10^9$

$10^9 * 10^9 = 10^{18}$

// Dry run

$$ans = (ans * a) \% p$$

ans	i
$\underline{1}$	1
$a \% p$	2
$a^2 \% p$	3

$ans = \underline{a \% p}$ no overflow

$ans = (a \% p * a) \% p$

$= [(a \% p) \% p * a \% p] \% p$

$= (a \% p * a \% p) \% p$

$\Rightarrow \underline{a^2 \% p}$

$ans = (a^2 \% p * a) \% p$

$= (a^2 \% p * a \% p) \% p$

$= (a^2 * a) \% p = a^3 \% p$

Q 2 Google Given a number in array format calculate arr % p

↳ each arr[i] represents a single digit of number.

Constraints

$1 \le N \le 10^5$

$0 \le arr[i] \le 9$

$1 \le p \le 10^9$

eg: $N = 5$ arr[5] : | 6 | 2 | 3 | 4 | 3 |

$p = 49$

$62343 \% 49$

Approaches

Idea 1 — not possible

Convert arr[] → number, then mod p

$N = 2$ max $99 = 10^2 - 1$

$N = 3$ $999 = 10^3 - 1$

⋮

$N = 10^5$ 10^{10^5} ~~10^5~~

⇓

storing in int / long is impossible

Idea 2 Calculating divisibility rule for any P, it won't work.

Hint Split number digit by digit.

$n = 5$

$p = 43$ arr[5] : | 6 | 2 | 3 | 4 | 3 |

$$[6 \times 10^4 + 2 \times 10^3 + 3 \times 10^2 + 4 \times 10^1 + 3 \times 10^0] \% p$$

| 6 | 2 | 3 | 4 | 3 |

$$[6 \times 10^4 + 2 \times 10^3 + 3 \times 10^2 + 4 \times 10^1 + 3 \times 10^0] \% p$$

$10^0 \% p$

$3 * t = 1$

$t = (t * 10) \% p$

$4 * t = 10^1 \% p$

$t = (t * 10) \% p$

$3 * t = 10^2 \% p$

$t = (t * 10) \% p$

$2 * t = 10^3 \% p$

$t = (t * 10) \% p$

$6 * 10^4 \% p$

$\Rightarrow$ add all the values

Pseudocode

```
arrmod ( int arr[], int p )
{
    int n = arr.length();
    long ~~int~~ sum = 0
    long ~~int~~ t = 1
    for ( int i = n-1 ; i >= 0 ; i-- )
    {
        sum = (sum + arr[i] * t) % p
        t = (t * 10) % p
    }
    return sum;
}
```

sum → [0, p-1], arr[i] → [0, 9], t → [0, p-1]

$p-1 + 9*(p-1)$

$\simeq 10p$

max $p \to 10^9$

$10p \to [0, 10^{10}]$

overflow.

$(t*10)$ → [0, p-1] → $\dfrac{10 * p}{}$

⇓

$[0, 10^{10}]$

overflow

Doubt

Java / C++ . . . .

$a/b \to$ integer division

$-40 \% 7 \to \quad -40/7 = -5$

$-40 - 7*(-5) = -40 + 35 = -5$

$-60 \% 9 \Rightarrow -60 - 9 * \overset{-60/9}{(-6)} = -60+54 = -6$

Python

a/b $\quad 6/4 = 1.5 \quad \text{floor}(1.5) = 1$

$$-40 \% 7 = -40 - 7 * (\text{floor}(-40/7))$$

$-5.6\ldots$

$$= -40 - 7 * (-6)$$

$$= -40 + 42 = 2$$