Strings

30 October 2022 02:05

String (Non Primitive)

↳ Collection of chars

'a' + 'b' + 'c' = "abc"

"a string" "Hi"

BODMAS

string + anything = string

. String + anything = string!

1) System.out.println("Hello" + 10); → "Hello 10"
2) System.out.println("Hello"+10+20); → "Hello10" + 20 → "Hello1020"
3) System.out.println(10+"Hello"+10+20); → "10Hello" + 10 + 20 → "10Hello10" + 20 → "10Hello1020"
4) System.out.println(20+10+"Hello"+10+20); 30 + "Hello" + 10 + 20 → "30Hello" + 10 + 20 → "30Hello10" + 20 → "30Hello1020"
5) System.out.println(20+10+"Hello"+(10+20)); → 30 + "Hello" + (10+20) → "30Hello" + (10+20) → "30Hello" + 30 → "30Hello30"

"Hello World"

Binary To Decimal

$(10110110)_2 \longrightarrow (\ ?\ )_{10}$ $\quad (1111)_2 \rightarrow (?)_{10}$

$1\times2^8 + 0\times2^7 + 1\times2^6 + 1\times2^5 + 0\times2^4 + 1\times2^3 + 1\times2^2 + 1\times2^1 + 0\times2^0$

$256 + 64 + 32 + 8 + 4 + 2 = (\ \_\_\_\ )_{10}$

Steps
→ Σ(Remainder 10 * Increasing Powers of 2)
multiplier * = 2

Code

Decimal to Binary

Jis side jana vaha se Remainder

$(14)_{10} \rightarrow (?)_2$

$2^3 2^2 2^1 2^0$
$1110 = 8+4+2 = 14$
$0111 = 4+2+1 = 7$

Long Division

2 | 14	
2 | 7	0×10^0
2 | 3	1×10^1
2 | 1	1×10^2
0	1×10^3

$(14)_{10} \rightarrow (1110)_2$

Steps
Σ(Remainder 2 * Increasing power of 10)
(0-1)

Octal → $(\ )_8$

0-7 ✓

0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	$1\times8^1 + 0\times8^0 = 8$
9	$1\times8^1 + 1\times8^0 = 9$
10	$1\times8^1 + 2\times8^0 = 10$
11	
12	
13	
14	
15	
16	
17	
20	

$(10)_8 \rightarrow (8)_{10}$

$(18)_{10} \rightarrow (?)_8$

8 | 18	2'2
8 | 2	2
0	2

$(22)_8 \rightarrow (?)_{10}$

↳ $2\times8^1 + 2\times8^0 = 16+2 = 18$ ✓

(Hexadecimal)$_{16}$ (0 - F)

0	
1	
2	
3	
4	
5	
6	
7	
8	
9	
A	10
B	11
C	12
D	13
E	14
F	15

$(16)_{10} \rightarrow (10)_{16}$

$1\times16^1 + 0\times16^0$
$= 16 + 0$
$= 16$

10✗

Anybase(x) to Decimal

$(abc)_x \rightarrow (\ ?\ )_{10}$ $\quad a\cdot x^2 + b\cdot x^1 + c\cdot x^0$

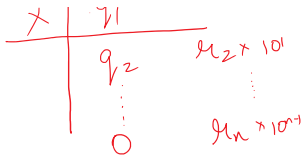
→ Extract Digits (Remainder 10)
→ Increasing power of x
→ Saaka Sum

Decimal to Anybase

$(100)_{10} \rightarrow (?)_x$

$= a_n\times10^{n-1} + a_{n-1}\times10^{n-2} + \dots + a_2\times10^1 + a_1\times10^0$

x | 100	$a_1\times10^0$
x | 91	$a_2\times10^1$
q_2	



$(x_1)_3 \rightarrow (?)_5$

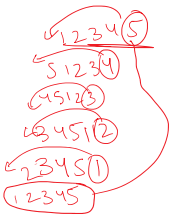
Direct Not Allowed

Anybase to Anybase Conversion

Decimal

Rotate a Number

r=0 12345
r=1 51234
r=2 45123
r=3 34512
r=4 23451
r=5 12345
r=6 51234
r=7 45123



Input n, r

↳ Output → Number after Rotation

1234,5 ⟶ 5,1234

↓

1234 + 5 × 10^{4} → nod of Remaining

= 51234

12345 ⇒ 5

10^4 10^3 10^2 10^1 10^0

5 1234

6 % 5 = 1

7 % 5 = 2

5 % 5 = 0

while(n) {

1 Rotation

↳ Last Digit × 10^{nod} + Remaining Number

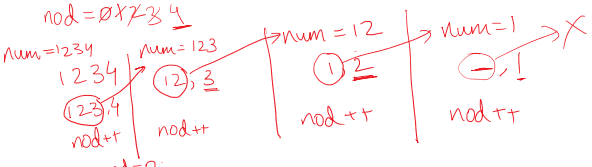
}

12345

rot = rot % nod

5 1234

5123 + 4 × 10^4



nod=0;

while(num!=0) {

num/=10;

nod++;

}

n = 12345

$10^3 = 1000$

part1 = 12, part2 = 345

n = 3

$345 \times 10^2 = 12$

n = 8%5 = 3

nod = 5

part2 = num % 10^3 → n

part1 = num / 10^3 → n

rotated number = part2 × 10^{nod-n} + part1

FAANG number: 2 3 1 4 5 (5 4 3 2 1)

Inverse ← 2 conditions
1) Unique Digits Only
2) 1 → nod

Digit	Position
5	1
4	2
1	3
3	4
2	5

Inverse ← Swap (Digit ⇔ Position)

1 2 4 5 3
5 4 3 2 1

6 5 4 3 2 1
6 4 3 1 5 2

D	P
2	1
5	2
1	3
3	4
4	5
6	6

Math.pow(10, -)

Inverse
10^5 10^4 10^3 10^2 10^1 10^0
6 2 5 4 1 3
6 5 4 3 2 1
↳ 10^1

879x
3 2 1

D	P
9	1
7	2
8	3

Digit Extract ✓
Position ✓
$\Sigma(\text{posi} \times 10^{D_i - 1}) = \text{ans}$

$1 \times 10^{2-1}$ ← D (2 / 1)
↓
P +

$2 \times 10^{5-1}$
$3 \times 10^{1-1} =$

Take 3 input :
Min F : 0
Max F : 100
Step : 20
For each F = 0,20,40,60,80,100 on a scale,
convert them into Celsius
C = 5/9*(F-32)
Output : with 4 spaces "\t"

0	-17
20	-6
40	4
60	15
80	26
100	37

$\frac{5}{9} \times (0-32)$

$\frac{5}{9} \times (40-32) = \frac{5 \times 8}{9} = \frac{40}{9} = 4.$

int / int = int

$\frac{5.0}{2} = 2.5$ ✓

double (8) / int (4) = double ✓

double / double = double ✓

int (4) / double (8) = double ✓