OX BARE

CMPSC OXC8 ALLEGHENY COLLEGE OX7E7
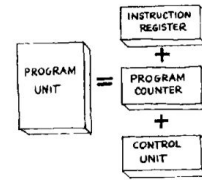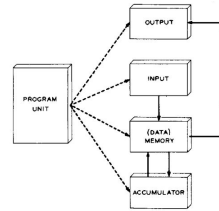
C8 METAL

Activity

For your designated table:

● look up the machine assigned to you, identify what you take to be its historical significance and list 3 facts that you find curious or interesting
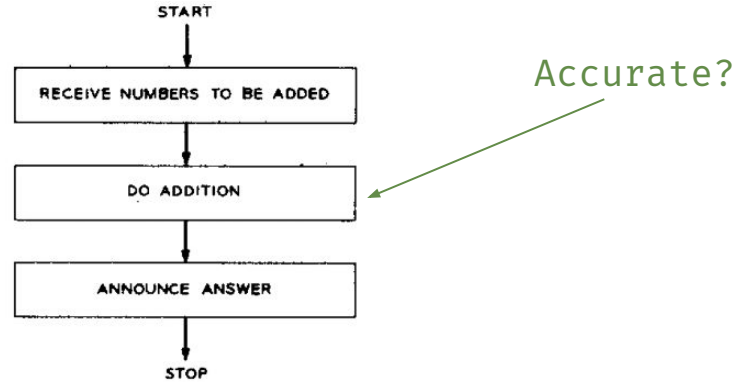
Atansoff-Berry	Altair 8800	MANIAC
DEC PDP-8	Manchester Mark I	Apple][

Activity

Atansoff-Berry	Altair 8800	MANIAC
DEC PDP-8	Manchester Mark I	Apple][

Add me some numbers

START

RECEIVE NUMBERS TO BE ADDED

DO ADDITION

Accurate?

ANNOUNCE ANSWER

STOP

Add me some numbers

START   (TURN ON THE COMPUTER)

READ FIRST NUMBER   (TAKE FIRST NUMBER FROM INPUT AND STORE IT IN THE MEMORY)

READ SECOND NUMBER   (TAKE SECOND NUMBER FROM INPUT AND STORE IT IN THE MEMORY)

PUT FIRST NUMBER IN ACCUMULATOR   (CLEAR THE ACCUMULATOR OF ANY PREVIOUS CONTENTS AND TRANSFER FIRST NUMBER TO IT FROM THE MEMORY)

ADD SECOND NUMBER TO FIRST   (TRANSFER SECOND NUMBER FROM MEMORY TO ACCUMULATOR AND ADD IT TO THE FIRST NUMBER)

STORE SUM IN MEMORY   (TRANSFER SUM FROM THE ACCUMULATOR TO THE MEMORY)

PRINT SUM   (TRANSFER THE SUM FROM THE MEMORY TO THE OUTPUT AND PRINT IT)

STOP   (TURN OFF THE COMPUTER)

Accurate?

Instruction set architecture (ISA)

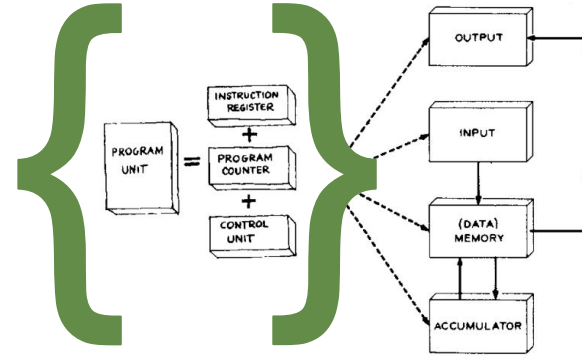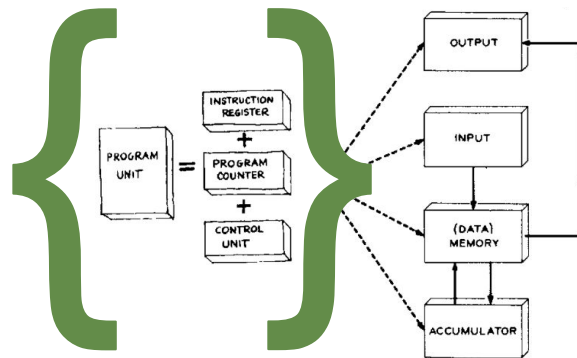Set of architecture-based on processor architecture (short definition).

Input

Clear accumulator and add

Add ← Adds number to current accumulator

Output

Halt

Instruction set architecture (ISA)

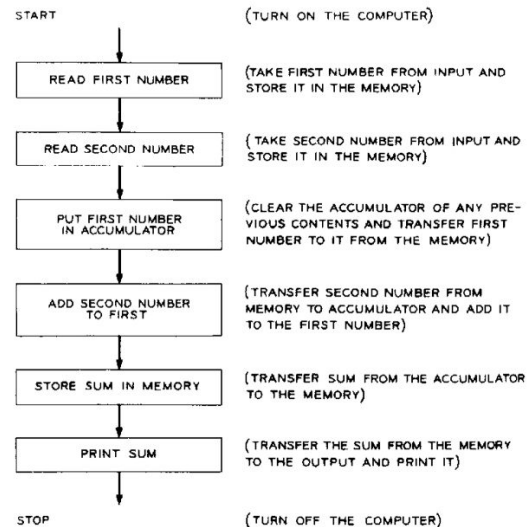What's the program to add 2+2 and 4+4, given this ISA?

Input

Clear accumulator and add

Add ←——————— Adds number to current accumulator

Output

Halt

Instruction set architecture (ISA)

What's the program to add 2+2 and 4+4, given this ISA?

Input

Clear accumulator and add

Add ← Adds number to current accumulator

Output

Store ← Store number from accumulator

Halt

START (TURN ON THE COMPUTER)

READ FIRST NUMBER (TAKE FIRST NUMBER FROM INPUT AND STORE IT IN THE MEMORY)

READ SECOND NUMBER (TAKE SECOND NUMBER FROM INPUT AND STORE IT IN THE MEMORY)

PUT FIRST NUMBER IN ACCUMULATOR (CLEAR THE ACCUMULATOR OF ANY PREVIOUS CONTENTS AND TRANSFER FIRST NUMBER TO IT FROM THE MEMORY)

ADD SECOND NUMBER TO FIRST (TRANSFER SECOND NUMBER FROM MEMORY TO ACCUMULATOR AND ADD IT TO THE FIRST NUMBER)

STORE SUM IN MEMORY (TRANSFER SUM FROM THE ACCUMULATOR TO THE MEMORY)

PRINT SUM (TRANSFER THE SUM FROM THE MEMORY TO THE OUTPUT AND PRINT IT)

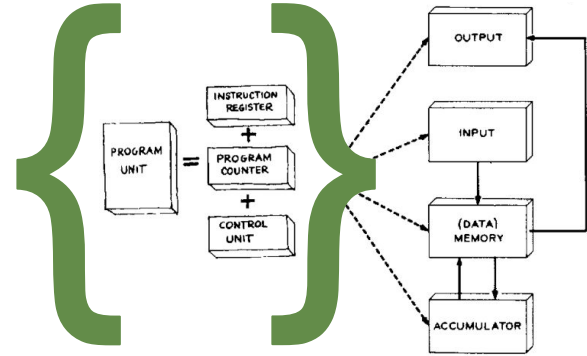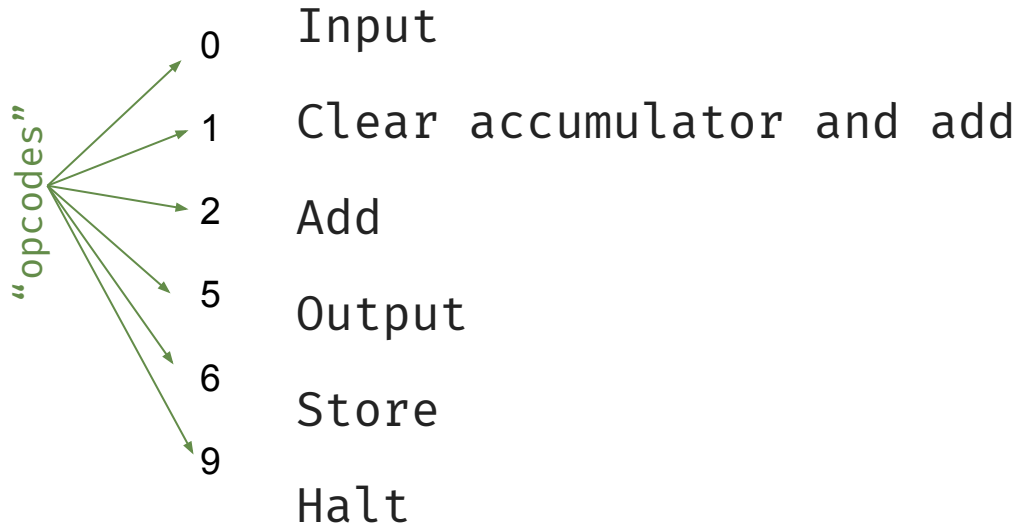STOP (TURN OFF THE COMPUTER)

A complication: registers

New rules, same problems:

- Memory can't be written to except by input
- You cannot output from the accumulator
- Operations have to use "registers" to perform actions
  - numbers have to have an open, unoccupied space to go (accumulator can be stored; spaces cannot be reused)
  - Instructions call the "register" where the number lives to use it
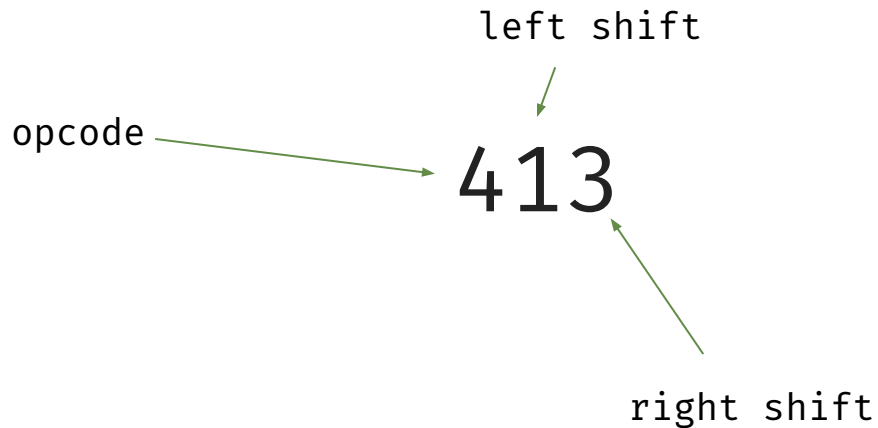- Can only use the amount of registers given to you on paper
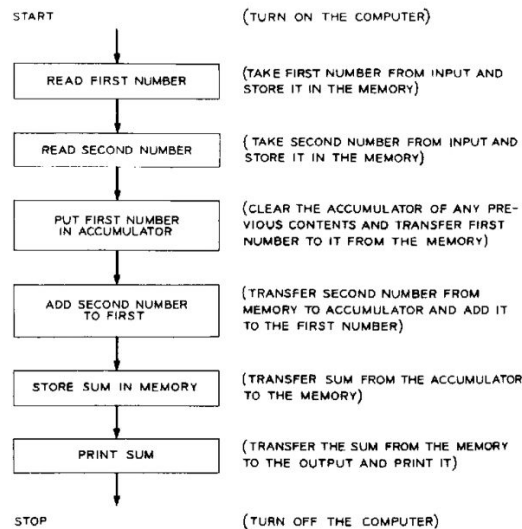
Instruction set architecture (ISA)

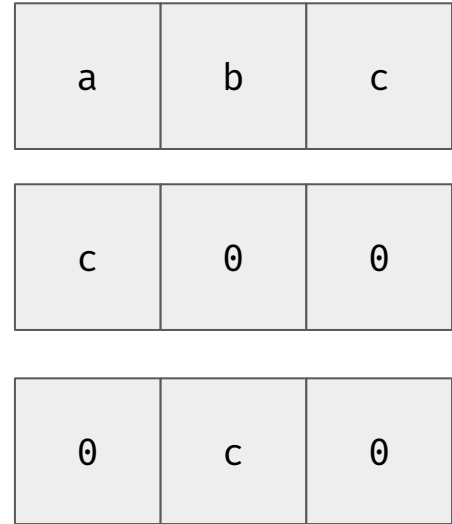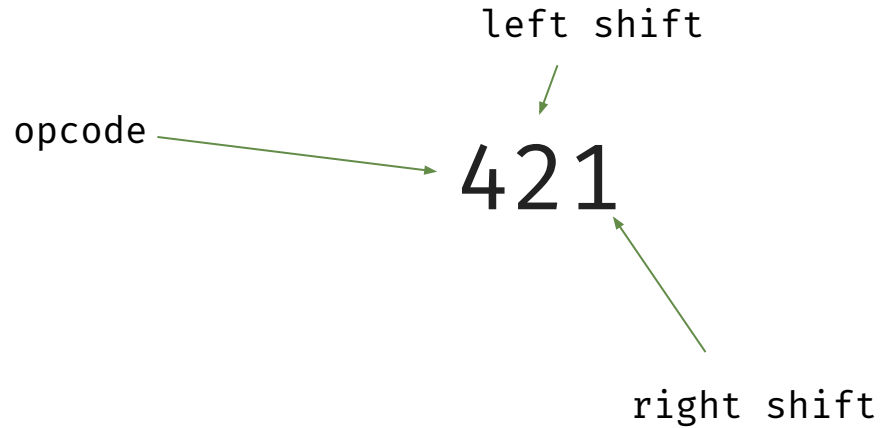Now, imagine that you can't use the full words; we can only use numerical data. How would you represent this ISA?
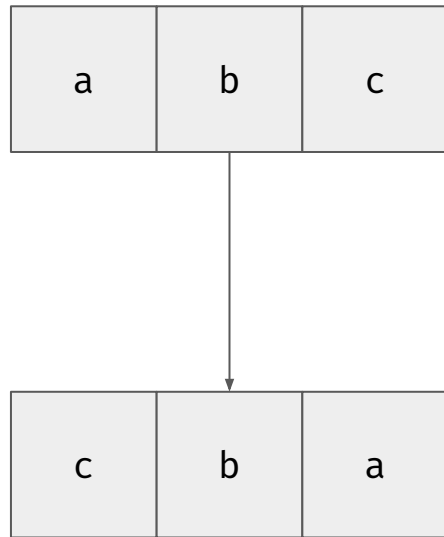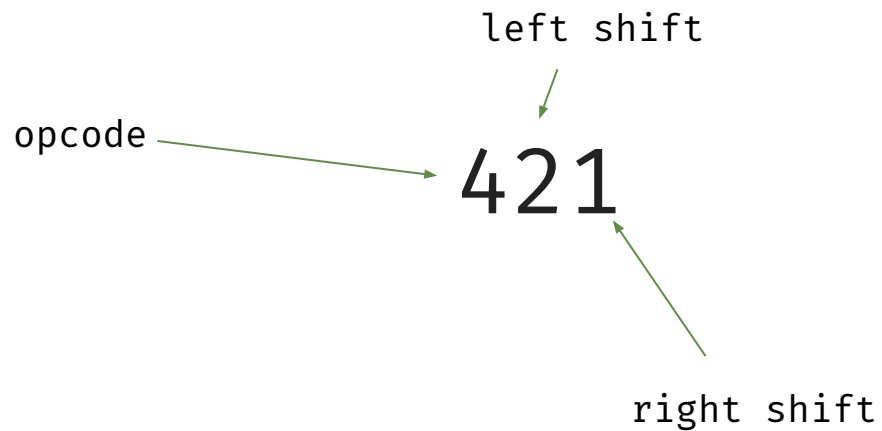
"opcodes"

0	Input
1	Clear accumulator and add
2	Add
5	Output
6	Store
9	Halt

The SFT instruction

"Shifts" the Accumulator

left shift

opcode

413

right shift

The SFT instruction

left shift

opcode → 421

right shift

a	b	c

c	0	0

0	c	0

The SFT instruction

left shift

opcode

421

right shift

a	b	c

c	b	a

Magic 9 Decoder
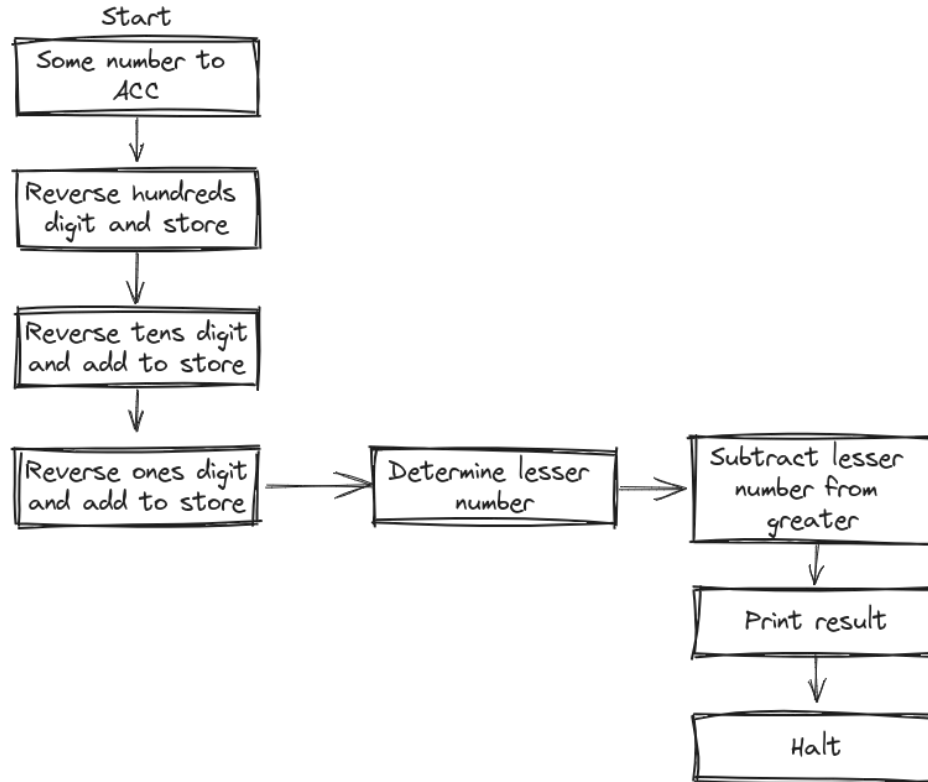
Subroutines
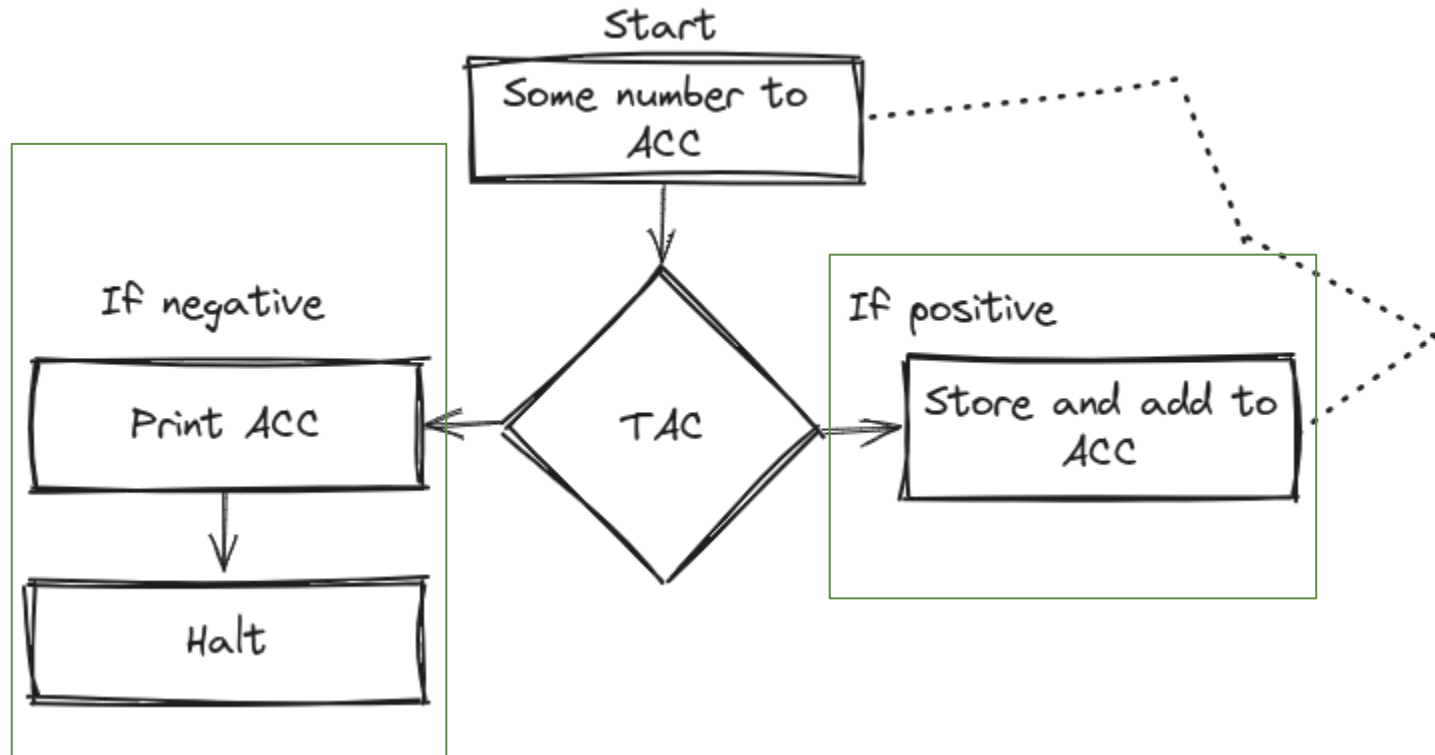
High dimensional adder

What happens if we add

in the CARDIAC?

→

413 4

,123

HOLE IN THE WORLD NEVER DO THIS
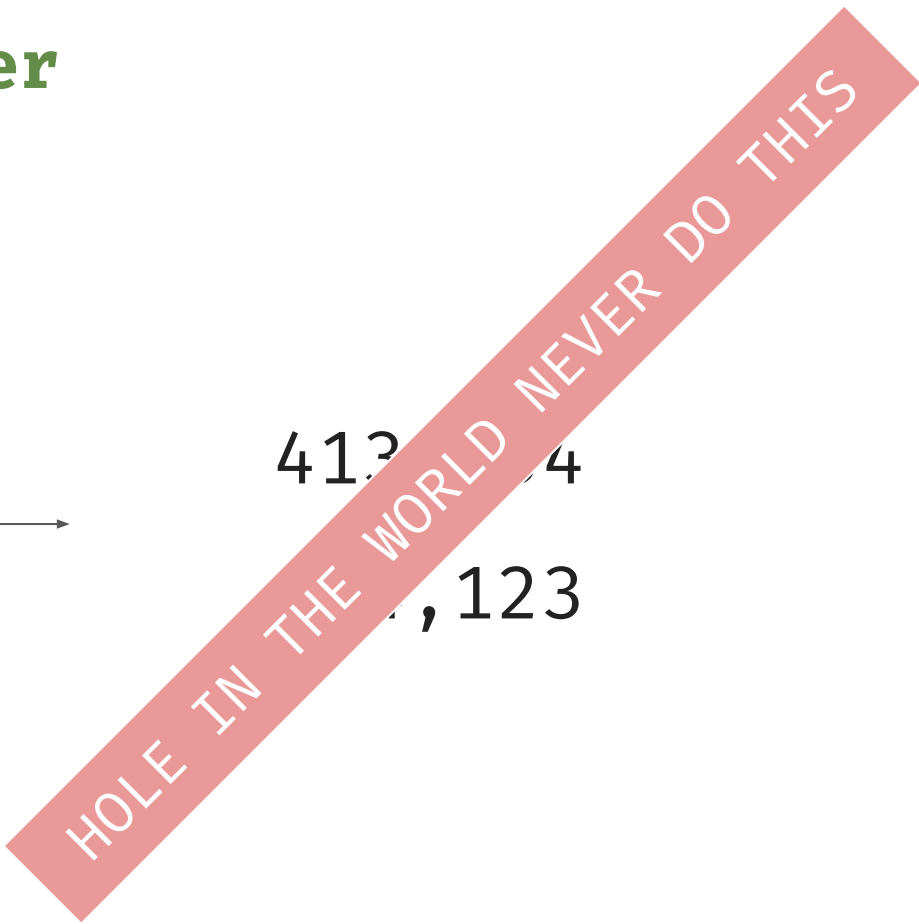
BUT WHAT IF WE DID IT ANYWAY

Least "significant" digits

413,234

Most "significant" digits
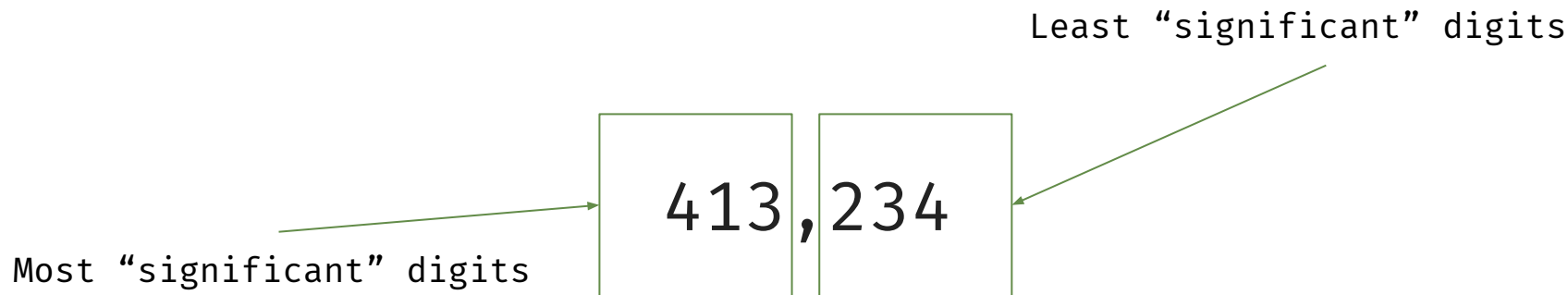
(Sorry, 234. You don't matter as much, apparently.)