



Workshop: Advent of Code – day 2

Adám Brudzewsky
Richard Park
Rodrigo Girão Serrão



Day 1

- ✧ Array-oriented techniques
- ✧ Bingo and Strings

Day 2

- ✧ Reading and parsing data from files
- ✧ Mathematical insights for array-oriented programming



Problems

- 15.2 Wrapping Presents
- 15.17 Filling Containers
- 16.3 Triangles

Topics

- Reading and parsing data from files
- Mathematical insights for array-oriented programming



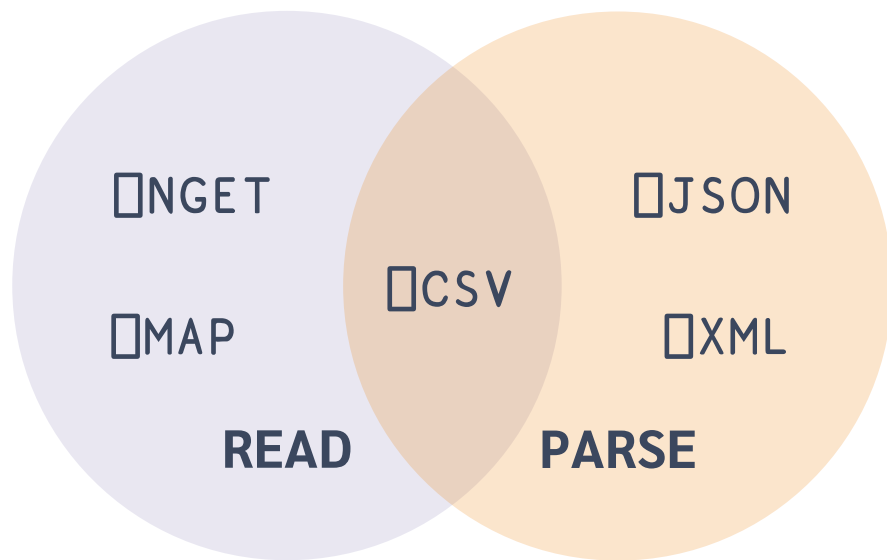
Reading and Parsing Input

System Functions



Reading and Parsing Input

System Functions



UNGET

- Vector of characters:

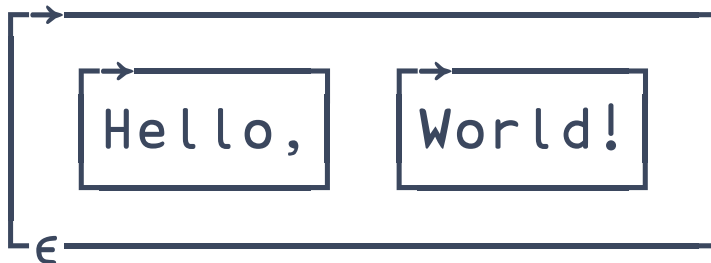
`>UNGET filename`



Line separator is
always `UNCS 10`

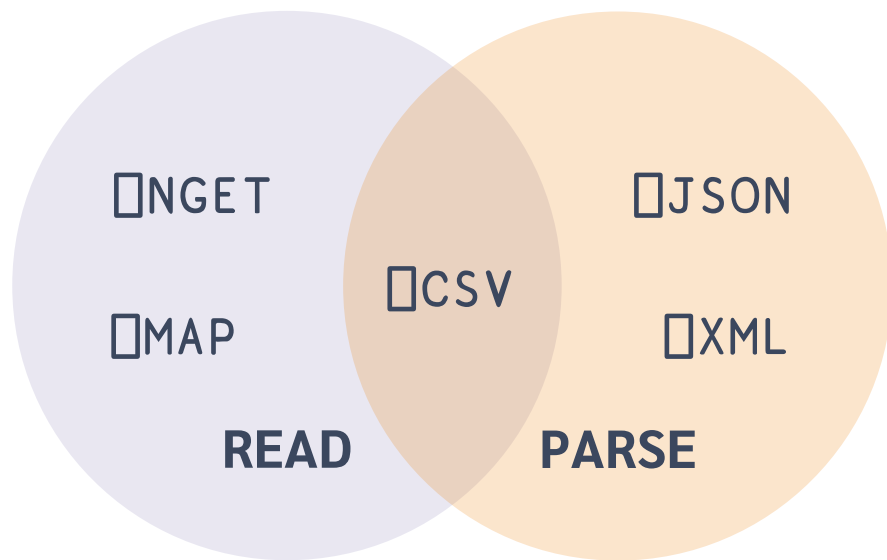
- Vector of character vectors:

`>UNGET filename 1`



Reading and Parsing Input

System Functions



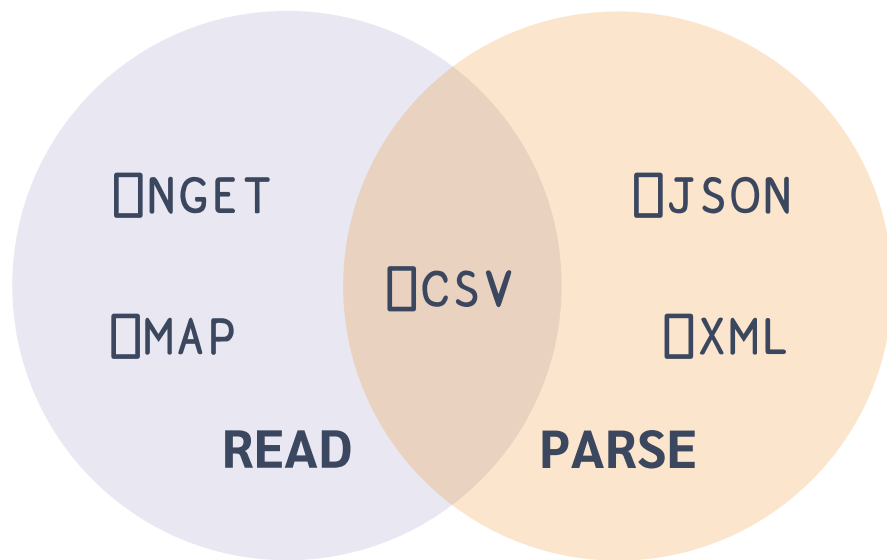
MAP

- Vector of characters: `80 ^1 MAP filename`
 - n-row matrix: `80 n ^1 MAP filename`
 - n-column matrix: `80 ^1 n MAP filename`
 - n-column m-row array: `80 ^1 n m MAP filename`
- etc.



Reading and Parsing Input

System Functions



CSV

From array:

```
CSV ('1,2,3' '4,5,6') @ 4
```

→			
↓	1	2	3
	4	5	6
~			

From file:

```
CSV 'foo.csv' @ 4
```

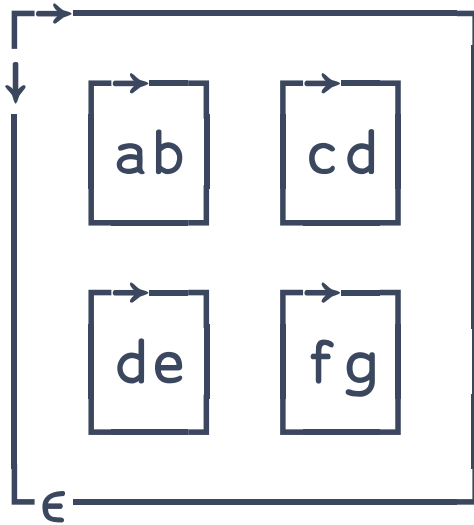
→			
↓	1	2	3
	4	5	6
~			

2 to error on
non-numbers



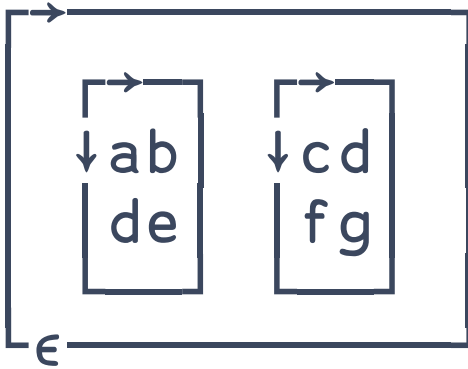
□ CSV

□ CSV = 'ab,cd' 'de,fg'



□ CSV

□ CSV □ 'Invert' 1 < 'ab,cd' 'de,fg'



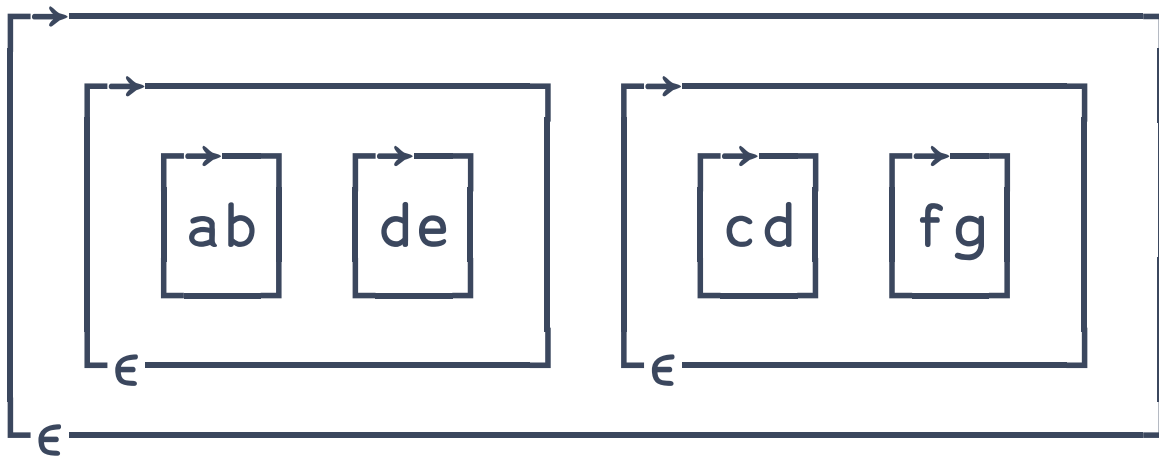
□ \Leftrightarrow □ OPT
is in Classic



CSV

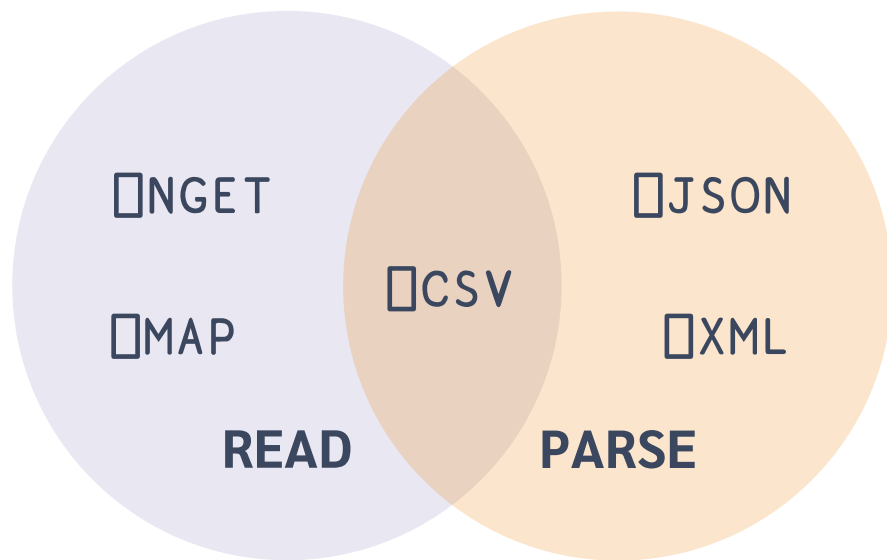
For more techniques, watch is.gd/dyalog_csv

CSV: Invert '2' < 'ab,cd' 'de,fg'



Reading and Parsing Input

System Functions



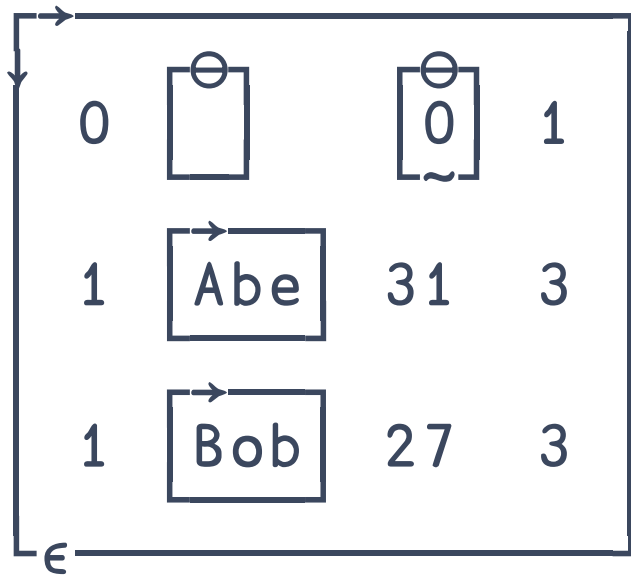
JSON

```
r ← JSON ' {"Abe":31, "Bob":27} '  
]names r  
r.Abe r.Bob  
r.Abe  
31
```



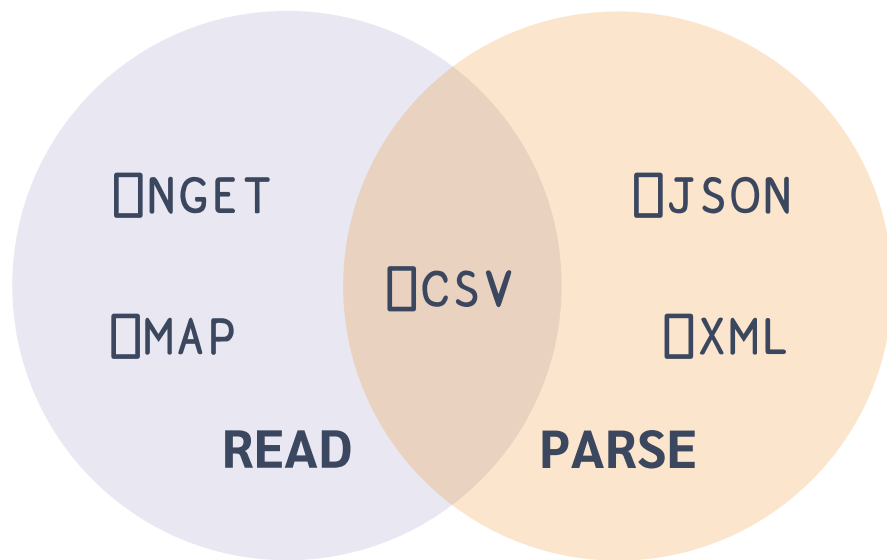
JSON

JSON: 'M' ⊢ '{ "Abe" : 31, "Bob" : 27 } '



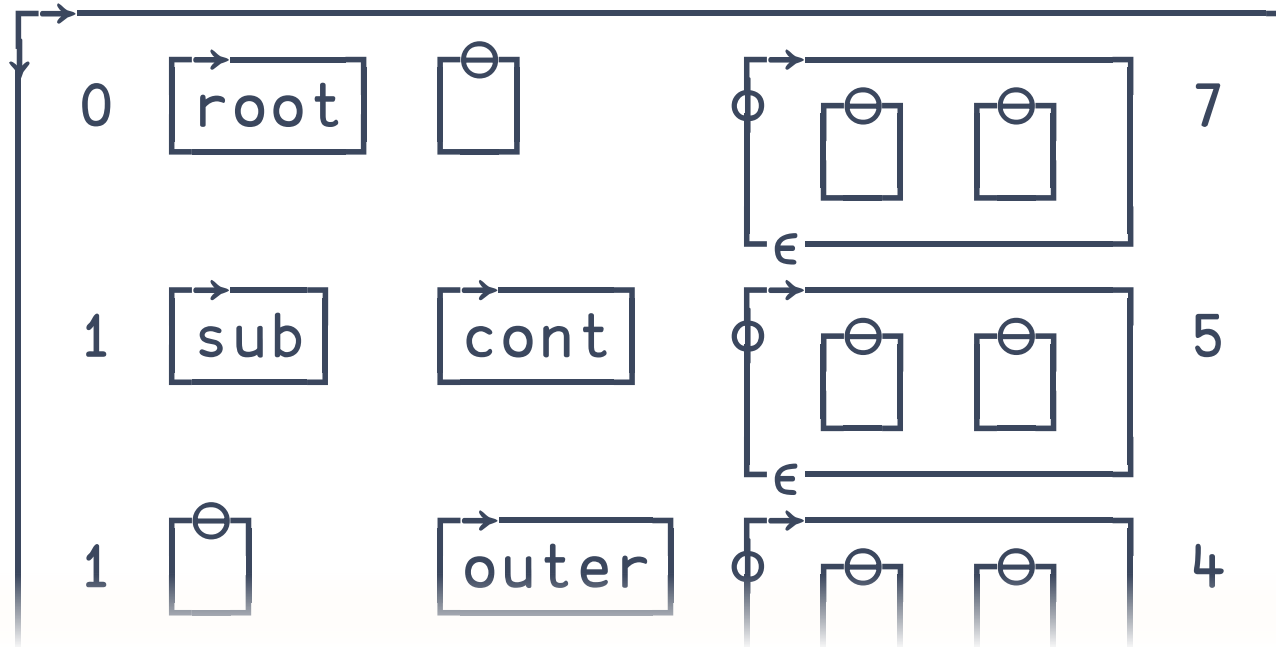
Reading and Parsing Input

System Functions



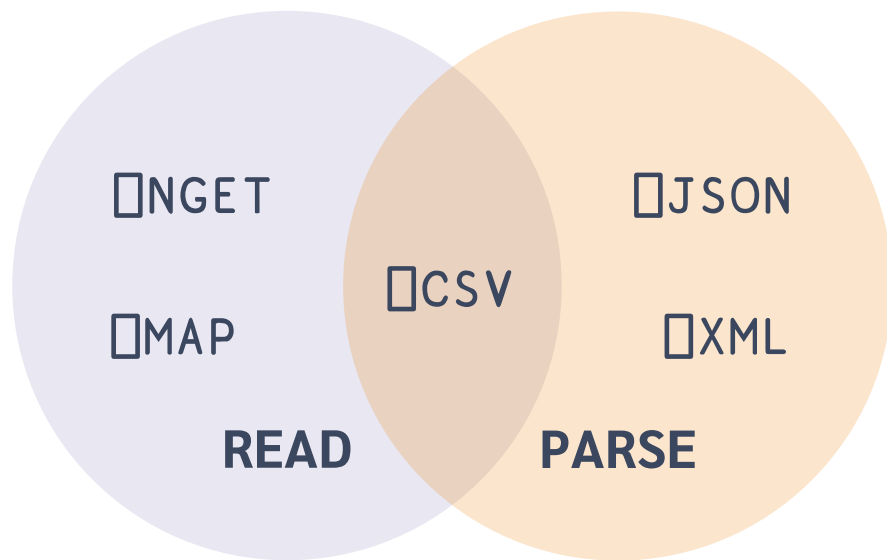
□ XML

□ XML ' <root>_{cont}outer</root> '



Reading and Parsing Input

System Functions



Reading and Parsing Input

"x"-delimited tuples to numeric matrix (15.2)

$\uparrow \{ \neg \text{JSON} '[', \omega, ']' \} \cdot 'x' \neg \text{R}', ' \Rightarrow \neg \text{NGET path } 1$

$\uparrow 'x' (0 \neg \text{JSON} \cdot \neq \subseteq \vdash) \cdot \Rightarrow \neg \text{NGET path } 1$

$\neg \text{CSV} \neg \text{'Separator' } 'x' \vdash \text{path } \theta \ 2$



Reading and Parsing Input

Lines of same length to character matrix (15.5, 16.6)

$\uparrow \Rightarrow \text{NGET path } 1$

$\Rightarrow \text{CSV} \oplus \text{'Invert' } 1 \subset \text{path}$

$\{0 \text{ } ^{-1} \downarrow \omega \rho \sim n, \sim n \div \sim \neq \omega\} \Rightarrow \text{NGET path}$

$0 \text{ } ^{-1} \downarrow 80 \text{ } ^{-1} n \text{ } \square \text{MAP path}$

n is line length + 1



Reading and Parsing Input

Fixed-width fields to numeric matrix (16.3)

```
↑ ' '(0 JSON''≠⊆⊢)''⇒NGET path 1
```

```
CSV⊢'Widths'(5 5 5)⊢path ' θ 2
```

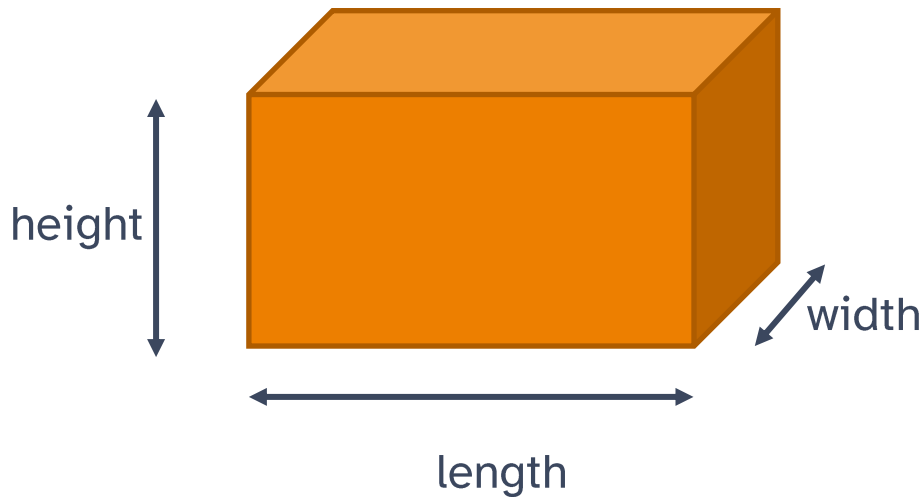


Mathematical Insights

- 15.2 Wrapping Presents
- 15.17 Filling Containers
- 16.3 Triangles



Wrapping Presents



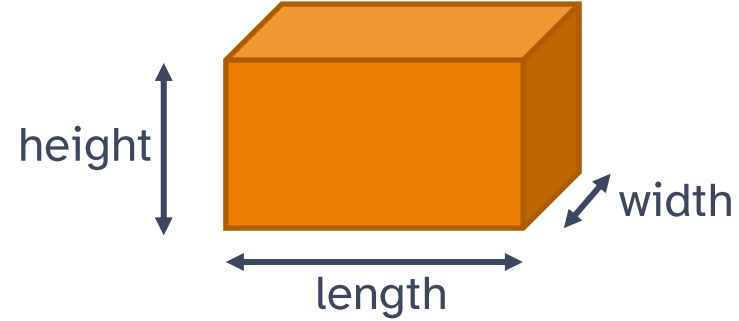
Wrapping Presents

Area of paper required:

Surface area of whole box

+

Area of smallest face

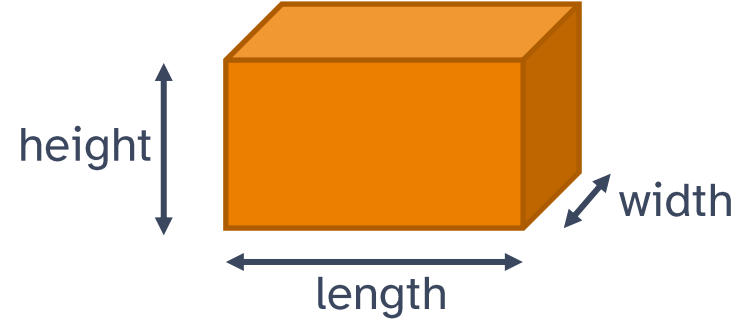


Wrapping Presents

$$(2 \times l \times w) + (2 \times w \times h) + (2 \times h \times l)$$

+

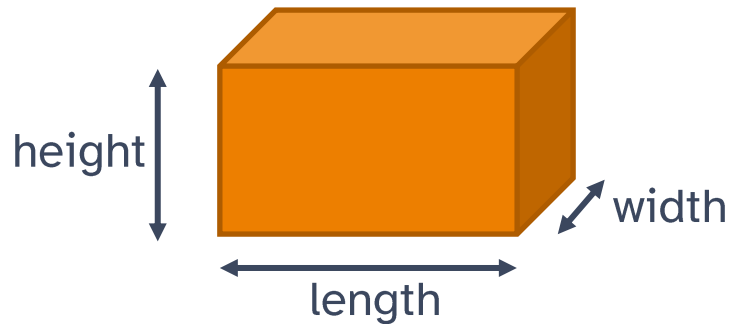
Area of smallest face



Wrapping Presents

Area of smallest face

- $\{\omega[\Delta\omega]\}$
 - $\times/2\uparrow$
 - $\times/\omega[1\ 2]$
- Multiply all 3 sides and divide by largest
 $(\times/\div[\ /) \ l \ w \ h$
- Take smallest of products of sides
 $l / l \ w \ h \times \ w \ h \ l$

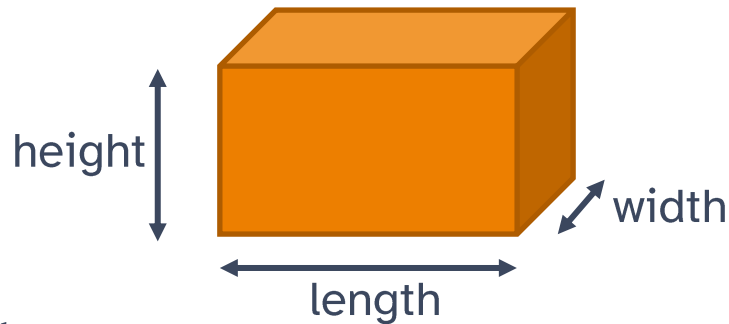


Wrapping Presents

Area of smallest face

Exercise: Using a similar insight, compute
the perimeter of the smallest face

Multiply all 3 sides and divide by largest
($\times / \div \lceil / \rceil$) l w h



Filling Containers

```
CanFillRecursive ← {  
   $\alpha \leq 0 : \alpha = 0$   
   $0 \in \rho\omega : 0$   
   $+ / (\alpha - \omega) \nabla'' (i \neq \omega) \downarrow'' \subset \omega$   
}
```



Filling Containers

Combinations are related to Boolean vectors

When doing base conversion, $\Box I O \leftarrow 0$

Exercise: Use τ (or \perp) to find all Boolean numbers 0 to ω



Filling Containers

Filtered arithmetic

Exercise: What is another way to write:

$$+/\alpha/\omega$$

for Boolean vector α and numeric vector ω ?



Filling Containers

Filtered arithmetic

Exercise: How can we filter numeric vector ω using each row of Boolean matrix α ?

α : 3 5p1 0 0 1

1 0 0 1 1

0 0 1 1 0

0 1 1 0 0

ω : 3 4 5 6 7

\leftarrow : A result need not be nested

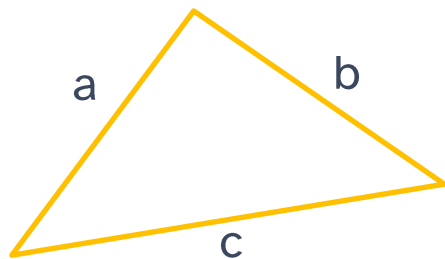
3	6	7	5	6	4	5
---	---	---	---	---	---	---



Triangles

Three numbers may define a triangle if

$$a + b \geq c$$

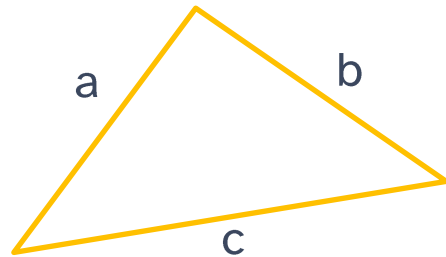


Triangles

Three numbers may define a triangle if

$$a + b \geq c$$

This requires finding out which two sides are shortest.



Can we find a calculation which doesn't depend on conditions?

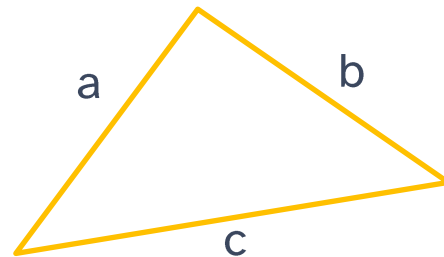


Triangles

For part 1, each row of 3 numbers defines the sides of a triangle.

For part 2, each vertical group of 3 defines the sides of a triangle.

How does the following code achieve this redefinition?

$$(\rho\omega)\rho,\Phi\omega$$




We're here for you!

General support

Forums

Chat room

Adám Brudzewsky

Richard Park

Rodrigo Girão Serrão

support@dyalog.com

forums.dyalog.com

apl.chat

adam@dyalog.com

rpark@dyalog.com

rodrigo@dyalog.com

