The Skewie Challenge

What’s a skewie?

No idea, it’s a random name I made up and has no meaning whatsoever.

The problem

- **Input:**
  - numbers x_1 to x_n in [1, 10] and n in [5, 8]
  - number y in [1, 100]
- **The goal: use the numbers `$x_1$ ... $x_n$` to create a formula that results in `y`**
- **Example:**
  - [1, 5, 3, 6, 8] and 23
  - Solution: “5 * 3 + 8 / 1 ^ 6”

Operators and precedence

Operator	Symbol
Power	^
Multiply and Divide	*, /
Add & Subtract	+, -

Note: the power operator is right associative, all other operators operators are left associative

a^b^c = a^(b^c)
a-b-c = (a-b) – c

Note: the result of 0^0 is 1

The rules

- **Allowed output characters:**

  [0-9] | “ “ | “+” | “-” | “*” | “/” | “^” | “(“ | “)”

- **Every x_i must be used exactly once! No other numbers are allowed.**
- **The order of numbers and operators does not matter.**
- **No unary operators!**

The rules

- **Every operation must result in an integer**
  - Bad: (1 / 3) * 6
  - Good: (6 / 3) * 1
- **x^y**
  - if |x| > 1 then |y| <= 6
  - if |y| > 1 then |x| <= 100

Practical

- **Language does not matter but it must compile/run on Linux**
- **Send in your source code before Friday, 3 Jan 2020**
- **Each puzzle will be sent as a CSV to stdin**

  1,5,3,6,8,23

- **Print a line with the solution to stdout**

  5 * 3 + 8 / 1 ^ 6

- **Answer must be printed within 1 second (may change)**
- **If there is no solution, print an empty line**

Practical

- 1000 random puzzles will be generated
- Every program will receive the same puzzles and will run consecutively, there's no point in trying to steal CPU or memory from other algorithms.
- Time is measured between sending the puzzle and receiving the solution

Scoring

- **The algorithm with the smallest average time gets 1 point.**
- **Every solution may also receive awards. For every award type, the algorithm with the most awards of that type gets a point.**
- **The algorithm with the most points wins. When it’s a draw, I’ll judge the cleanliness of the code.**

Awards

Award	Definition	Example
PICASSO	Operators and parentheses are symmetric	(3 – 1) * 5 * (8 - 3)
SUPREME POWER	At least 50% of operators are a power operator	2 ^ 3 * 5 - 1 ^ 6
HOLD THE LINE	Do not alter the order of the numbers	
HEADACHE	Use every operator only once	
BINARY FUN	Use a power of 2 (with an exponent > 2)	(3-1) ^ 5 + 6 – 3

Awards

Award	Definition	Example
BIG TIME	Have an intermediary value that's more than twice the result	(4 * 6) / 8 * 3 + 2
COUNTDOWN	Use the numbers in descending order	(8 - 5) * 4 + 2 – 1
ASCENSION	Use the numbers in ascending order	1 + 2 * 4 + 5 * 8
FROZEN	Have an intermediary value below -10	(4 - 6) * 6 + 8 * 2
NINJA	Beat another algorithm by less than 5ms (even if you're not first)	

For those who need hardware specs

- Architecture: x86_64
- CPU op-mode(s): 32-bit, 64-bit
- Byte Order: Little Endian
- Address sizes: 43 bits physical, 48 bits virtual
- Model name: AMD Ryzen 5 2600X Six-Core Processor
- CPU max MHz: 3600.0000
- L1d cache: 192 KiB
- L1i cache: 384 KiB
- L2 cache: 3 MiB
- L3 cache: 16 MiB
- Memory: ~15 GiB

Have fun