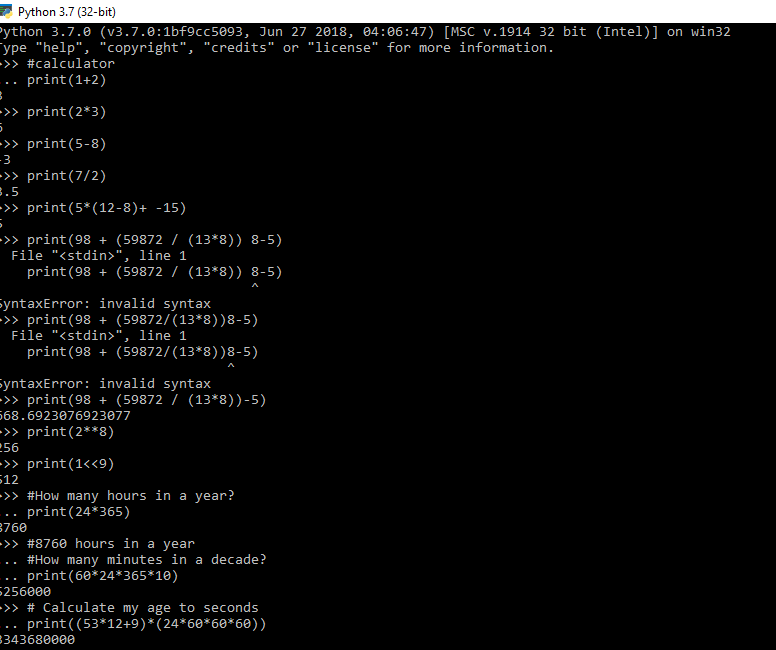
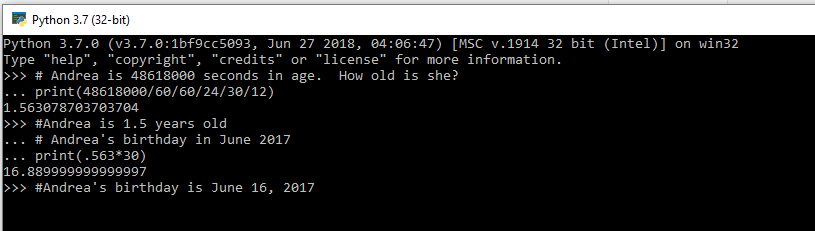
Frances Endencia July 18, 2018

1MillionWomentotech

Homework week1 day 1





Frances Endencia July 18, 2018

Q: How many days does it take a 32 bit system to time out if there is an integer overflow?

The latest time that can be represented in Unix's [signed 32-bit integer](https://en.wikipedia.org/wiki/Integer_(computer_science)) time format is 03:14:07 [UTC](https://en.wikipedia.org/wiki/Coordinated_Universal_Time) on Tuesday, 19 January 2038 (231-1 = 2,147,483,647 seconds after 1 January 1970).[[1]](https://en.wikipedia.org/wiki/Year_2038_problem#cite_note-spinellis-1) Times beyond that will wrap around and be stored internally as a negative number, which these systems will interpret as having occurred on 13 December 1901 rather than 19 January 2038. This is caused by [integer overflow](https://en.wikipedia.org/wiki/Integer_overflow). The counter runs out of usable digit bits, flips the sign bit instead, and reports a maximally negative number (continuing to count *up*, toward zero). Resulting erroneous calculations on such systems are likely to cause problems for users and other relying parties.

Programs that work with future dates will begin to run into problems sooner; for example a program that works with dates 20 years in the future would have to have been fixed no later than 19 January 2018.

Q: How about 64 bit? The integer overflow does not occur in a 64 bit because it is longer.