Homework 1 Rubric

-20% from points earned if not in Python 3

Style

Code	Description
Style1	Header contains name and uni
Style2	No lines over 79 characters.
Style3	Comments explain logical blocks of code
Style4	Whitespace - between blocks of code, between functions, etc
Style5	Files named according to instructions
Style6	Follows assignment instructions

Time Convert

Code	Description
1A	Uses mod or division and subtraction to calculate times
1B	Asks for user input
1C	3672 => 1 hours(s) 1 min(s) 12 sec(s)
1D	1 => 0 hours(s) 0 min(s) 1 sec(s)
1E	61 => 0 hours(s) 1 min(s) 1 sec(s)

Sum, min, and max

Code	Description
2A	Uses loop to collect input
2B	Terminates on “X”
2C	“3\nX” (\n is a new line) -> prints “sum: 3, min: 3, max: 3”
2D	“3\n-2\n5\n7\n-1\n0\nX” -> prints “sum: 12, min: -2, max: 7”
2E	Handles the case in which “X” is the first and only input

Perfect Numbers

Code	Description
3A	28 True
3B	496 True
3C	100 False
3D	5749 False
3E	Efficiency -- doesn't check every single number from 1 to n
3F	Efficiency - only checks up to sqrt(n) factors

Prime Numbers

Code	Description
4A	Used a loop which did not increment past n
4B	25 False
4C	17 True
4D	0 False, 1 False, 2 True
4E	Efficiency (1) break whenever you see one factor (2) check up to sqrt(n) factors or (less efficiently) for up to n//2

Decimal to Base 2

Code	Description
5A	16 ---> 10000
5B	18472 ---> 100100000101000
5C	Correct loop structure for getting binary digits
5D	Storing binary digits in a string instead of printing them out
5E	Printing out reversed string

Design an Algorithm

Code	Description
6A	More efficient than brute force
6B	Deals with largest possible number 987654321 => “”
6C	Algorithm is correct
6D	Written in legible pseudocode / bulleted English