

1. Tracing control statements

(a) Suppose that the function `mystery` is defined as

```
def mystery(x, y):  
    while x != 0 and y != 0:  
        if x > y:  
            x -= y  
        else:  
            y -= x  
    return max(x, y)
```

Trace through the execution of the function to find the value of `mystery(77, 42)`.

(b) Suppose that the function `puzzle` is defined as

```
def puzzle(n):  
    s = 0  
    for i in range(1, 2 * n, 2):  
        s += i  
    return s
```

Trace through the execution of the function to find the value of `puzzle(8)`.

2. In both the novel and the movie version of Andy Weir’s 2011 bestseller *The Martian*, Mark Watney (played by Matt Damon in the film) communicates with Earth by recovering the long-dormant Sojourner rover, recharging its batteries, and using its communication system to send video signals to Earth through Sojourner’s camera. Communication from NASA back to Mars, however, is more challenging because the only thing that the Earth-based operators can do is rotate the camera. Mark’s solution is to position 16 flags in a circle around the camera, each of which is labeled with a hexadecimal digit (0-F) and then have the NASA technicians rotate the camera to spell out words in the ASCII subset of Unicode, which is reproduced as Figure 1 below.

For example, the first message that NASA sends (in both the movie and the book) is

48 4F 57 41 4C 49 56 45

If you look up these digit pairs in Figure 1, you will see that it spells out “**HOWALIVE**”.

Another message that appears in the novel (but not in the movie) is:

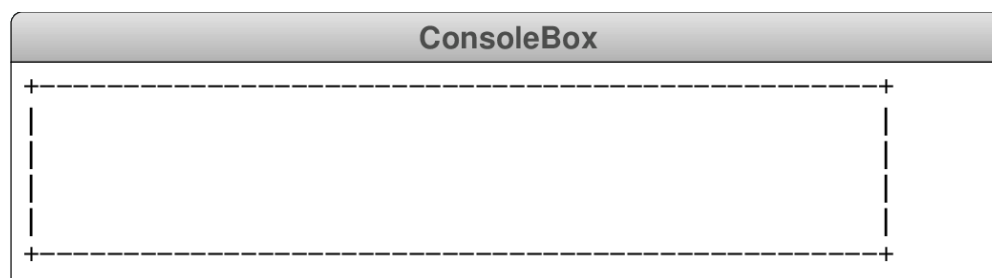
42 52 49 4E 47 53 4A 52 4E 52 4F 55 54

What is NASA trying to tell Mark?

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0x	\0								\b	\t	\n	\v	\f	\r		
1x																
2x	space	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
3x	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4x	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5x	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
6x	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7x	p	q	r	s	t	u	v	w	x	y	z	{		}	~	

Figure 1: The ascii table layed out in a hexadecimal grid. The starting hexadecimal characters can be found in the rows to the left, while the second characters range across the top.

3. Write a function `draw_console_box(width, height)` that draws a box on the console with the specified dimensions. The corners of the box should be represented using a plus sign (+), the top and bottom borders using a minus sign (-), and the left and right borders using a vertical bar (|). For example, calling `draw_console_box(52, 6)` should produce the following diagram:



4. In the Middle Ages, one of the problems that required sophisticated algorithmic thinking was determining the date of Easter, which falls on the first Sunday after the first full moon following the vernal equinox. Given this definition, the calculation involves interacting cycles of the day of the week, the orbit of the moon, and the passage of the sun through the zodiac. Early algorithms for solving this problem date back to the third century and are described in the writings of the eighth-century scholar known as the Venerable Bede. In 1800, the German mathematician Carl Friedrich Gauss published an algorithm for determining the date of Easter that was purely computational in the sense that it relied on arithmetic rather than looking up values in tables. His algorithm appears below.
- Divide the number of the year for which one wishes to calculate Easter by 19, by 4, and by 7, and call the remainders of these divisions a , b , and c , respectively.
 - Divide the value $19a + 24^*$ by 30 and call the remainder d .
 - Divide $2b + 4c + 6d + 5^*$ by 7 and call the remainder e .

*This digit depends on the century in question, and this value works for 2000-2099

- If $d + e$ is less than 10, then Easter is on March $22 + d + e$, otherwise it is on April $d + e - 9$.
- (a) Write a Python function `find_easter_date(year)` that return a string showing the date of Easter in the specified year. For example, calling `find_easter_date(2020)` should return the string `"April 12"`.
- (b) Write a testing function that compares the output of your function to the below provided years and date using `assert` statements. Remember that running this function should result in **no output** if everything is working correctly.

Year	Easter Date
2020	April 12
2021	April 4
2022	April 17
2023	April 9
2024	March 31
2025	April 20