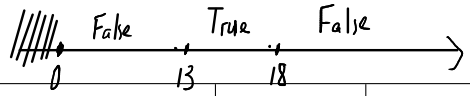
CSCA08H Worksheet: Choosing Test Cases

For each of the following functions, choose a set of test cases. Test only with valid input and avoid duplicate tests. The tables may contain more or fewer rows than necessary.

1.
```
def is_teenager(age: int) -> bool:
    """Return True iff age is between 13 and 18 inclusive.

    Precondition: age >= 0
    """
```



Test Case Description	age	Expected Result
boundary: minimum valid input	0	F
boundary: smallest returns True	13	T
boundary: biggest returns True	18	T
less than 13, returns False	5	F
between 13 and 18, returns True	15	T
larger than 18, returns False	25	F
	12	F
	14	T
	17	T
	19	F

2.
```
def all_fluffy(word: str) -> bool:
    """Return True iff every character in word is fluffy. Fluffy characters are those that
    appear in the word 'fluffy'. If word is empty, return True.
    """
```

Test Case Description	word	Expected Result
empty	""	True
1 char: fluffly	"f"	True
1 char: not fluffy	"x"	False
longer, fluffy	"flufffl|y"	True
longer, all non-fluffy	"xxzop"	False
longer, only last char non-fluffy	"fufx"	False
longer, only first char non-fluffy	"zf|y"	False
longer, a mix of fluffy and non-fluffy chars	"fzlyff"	False

extra: consider each of 'f', 'l', 'u', 'y' separately
whitespace
capitalization

3. `def same_abs(int1: int, int2: int) -> bool:`
   `"""Return True iff int1 and int2 have the same absolute value.`
   `"""`

positive vs negative vs zero
same sign vs diff. signs

Test Case Description	int1	int2	Expected Result
both pos, same abs	5	5	T
both pos, diff abs	3	4	F
both neg, same abs	-3	-3	T
both neg, diff abs	-4	-7	F
pos and neg, same abs	4	-4	T
pos and neg, diff abs	5	-4	F
all of the above, but also considering zeros!	0	0	
	0	2	
	-2	0	
first pos, second neg vs first neg second pos.	5	-5	
	-2	2	

4. `def most_popular(company_to_placements: dict[str, list[int]]) -> list[str]:`
   `"""Return the company (or companies) with the most placements in the race`
   `according to company_to_placements.`

   `Precondition: company_to_placements is not empty.`
   `"""`

1 company or
> 1 company

Test Case Description	company_to_placements	Expected Result
1 company	{'a': [1, 2, 3]}	['a']
2 companies, winner	{'a': [3, 6], 'b': [1, 2, 4]}	['b']
	{'a': [3, 6, 5], 'b': [1, 2, 4]}	['a', 'b'] OR ['b', 'a']
many companies, winner	{'a': [1], 'b': [3, 5, 6], 'c': [2, 4], 'd': [7, 8]}	['b']
many companies, some ties	{'a': [1], 'b': [3, 5, 6], 'c': [2, 4], 'd': [7, 8, 9]}	['b', 'd'] or ['d', 'b']
many companies, all tied	{'a': [1, 6], 'b': [3, 5], 'c': [2, 4], 'd': [7, 8]}	list with 'a','b','c','d' in any order
a company with - no placements, - one pacement - many placements		

winner: company with most placements