```
print("""
    🚨   SPOILER ALERT   🚨
    🚨  SOLUTIONS AHEAD 🚨
""")
```

```python
# Day one

### Part one

with open('inputs/dayone.txt') as f:
    calorie_input = f.read().split('\n\n')

elves = []
for calories in calorie_input:
    elves.append(sum([int(calorie) for calorie in calories.split('\n')]))

print(max(elves))

#### Part two

print(sum(sorted(elves, reverse=True)[:3]))
```

```python
# Day two

### part one

with open('inputs/daytwo.txt') as f:
    moves = f.read().split('\n')
    round_inputs = [move.split() for move in moves]


def calculate_points(their_move, your_move):
    their_move_points = point_mapper[their_move]
    your_move_points = point_mapper[your_move]

    action = [their_move, your_move]

    if their_move == your_move:
        their_result = your_result = 'draw'
    elif action in your_winning_actions:
        your_result, their_result = 'win', 'lose'
    elif action not in your_winning_actions:
        their_result, your_result = 'win', 'lose'

    their_round_points = point_mapper[their_result]
    your_round_points = point_mapper[your_result]

    return (
      their_move_points + their_round_points,
      your_move_points + your_round_points
    )
```

```python
# Day two

### part one continued...

point_mapper = {
    'rock': 1,
    'paper': 2,
    'scissors': 3,
    'lose': 0,
    'win': 6,
    'draw': 3
}

move_mapper = {
    'A': 'rock',
    'B': 'paper',
    'C': 'scissors',
    'X': 'rock',
    'Y': 'paper',
    'Z': 'scissors',
}

your_winning_actions = [
    ['rock', 'paper'],
    ['paper', 'scissors'],
    ['scissors', 'rock']
]

their_total_points = 0
your_total_points = 0

for round_ in round_inputs:
    their_move = move_mapper[round_[0]]
    your_move = move_mapper[round_[1]]

    their_round_points, your_round_points = calculate_points(
      their_move, your_move
    )

    your_total_points += your_round_points
    their_total_points += their_round_points

print(your_total_points)
```

```python
# Day two

### part two

round_decider = {
    'scissors': {
        'X': 'paper',
        'Y': 'scissors',
        'Z': 'rock'
    },
    'rock': {
        'X': 'scissors',
        'Y': 'rock',
        'Z': 'paper'
    },
    'paper': {
        'X': 'rock',
        'Y': 'paper',
        'Z': 'scissors'
    }
}

their_total_points = 0
your_total_points = 0
for round_ in round_inputs:
    their_move = move_mapper[round_[0]]
    your_move = round_decider[move_mapper[round_[0]]][round_[1]]

    their_round_points, your_round_points = calculate_points(
      their_move, your_move
    )

    your_total_points += your_round_points
    their_total_points += their_round_points

print(your_total_points)
```

```python
# Day three

### part one

with open('./inputs/daythree.txt', 'r') as f:
    input_items = f.read().split('\n')

points = {
    **{chr(i):n+1 for n,i in enumerate(range(ord('a'),ord('z')+1))},
    **{chr(i):n+27 for n,i in enumerate(range(ord('A'),ord('Z')+1))}
}

total_points = []
for n,item in enumerate(input_items):
    comp_one = item[:int(len(item)/2)]
    comp_two = item[int(len(item)/2):]
    overlap = set(comp_one).intersection(comp_two)
    if overlap:
        priority = overlap.pop()
        total_points.append(points[priority])

print(sum(total_points))

### part two

def chunks(xs, n):
    n = max(1, n)
    return (xs[i:i+n] for i in range(0, len(xs), n))

total_points = []
for groups in list(chunks(input_items, 3)):
    group_badge = set(
        groups[0]
    ).intersection(
        set(groups[1])
    ).intersection(groups[2])

    total_points.append(points[group_badge.pop()])
print(sum(total_points))
```

```python
# Day 4

### part one

with open('./inputs/dayfour.txt', 'r') as f:
    input_items = [i.split(',') for i in f.read().split('\n')]


def apply_conditionals(conditionals1, conditionals2, operator="all"):
    if eval(operator)(conditionals1) or eval(operator)(conditionals2):
        return 1
    else:
        return 0


def main(input_items, operator="any"):
    counter = 0
    for n,assignments in enumerate(input_items):
        assignment1 = [int(i) for i in assignments[0].split('-')]
        assignment2 = [int(i) for i in assignments[1].split('-')]

        sections1 = list(range(assignment1[0], assignment1[1]+1))

        sections2 = list(range(assignment2[0], assignment2[1]+1))

        returned_counter = apply_conditionals(
            [sections2[0] in sections1, sections2[-1] in sections1],
            [sections1[0] in sections2, sections1[-1] in sections2],
            operator=operator
        )

        counter += returned_counter

    return counter


print(main(input_items, "all"))

# part two: change to any
print(main(input_items, "any"))
```