

[Advent of Code](#)
[\[About\]](#)
[\[Events\]](#)
[\[Shop\]](#)
[\[Settings\]](#)
[\[Log Out\]](#)
[notthehoople \(AoC++\) 14*](#)
[/*2018*/](#)
[\[Calendar\]](#)
[\[AoC++\]](#)
[\[Sponsors\]](#)
[\[Leaderboard\]](#)
[\[Stats\]](#)

--- Day 15: Beverage Bandits ---

Having perfected their hot chocolate, the Elves have a new problem: the **Goblins** that live in these caves will do anything to steal it. Looks like they're here for a fight.

You scan the area, generating a map of the walls (**#**), open cavern (**.**), and starting position of every Goblin (**G**) and Elf (**E**) (your puzzle input).

Combat proceeds in rounds; in each round, each unit that is still alive takes a turn, resolving all of its actions before the next unit's turn begins. On each unit's turn, it tries to move into range of an enemy (if it isn't already) and then attack (if it is in range).

All units are very disciplined and always follow very strict combat rules. Units never move or attack diagonally, as doing so would be dishonorable. When multiple choices are equally valid, ties are broken in reading order: top-to-bottom, then left-to-right. For instance, the order in which units take their turns within a round is the reading order of their starting positions in that round, regardless of the type of unit or whether other units have moved after the round started. For example:

These units:	would take their turns in this order:
#####	#####
#.G.E.#	#.1.2.#
#E.G.E#	#3.4.5#
#.G.E.#	#.6.7.#
#####	#####

Our **sponsors** help make Advent of Code possible:

Formlabs - 3D printing with lasers. Software, hardware, and more. Pew pew!

Each unit begins its turn by identifying all possible targets (enemy units). If no targets remain, combat ends.

Then, the unit identifies all of the open squares (□) that are in range of each target; these are the squares which are adjacent (immediately up, down, left, or right) to any target and which aren't already occupied by a wall or another unit. Alternatively, the unit might already be in range of a target. If the unit is not already in range of a target, and there are no open squares which are in range of a target, the unit ends its turn.

If the unit is already in range of a target, it does not move, but continues its turn with an attack. Otherwise, since it is not in range of a target, it moves.

To move, the unit first considers the squares that are in range and determines which of those squares it could reach in the fewest steps. A step is a single movement to any adjacent (immediately up, down, left, or right) open (□) square. Units cannot move into walls or other units. The unit does this while considering the current positions of units and does not do any prediction about where units will be later. If the unit cannot reach (find an open path to) any of the squares that are in range, it ends its turn. If multiple squares are in range and tied for being reachable in the fewest steps, the square which is first in reading order is chosen. For example:

Targets:	In range:	Reachable:	Nearest:	Chosen:
#####	#####	#####	#####	#####
#E..G.#	#E.?G?#	#E.@G.#	#E.!G.#	#E.+G.#
#...#.# -->	#.?.#?# -->	#.@.#.# -->	#.!.#.# -->	#...#.#
#.G.#G#	?G?#G#	@G@#G#	!G.#G#	.G.#G#
#####	#####	#####	#####	#####

In the above scenario, the Elf has three targets (the three Goblins):

- Each of the Goblins has open, adjacent squares which are in range (marked with a `?` on the map).
- Of those squares, four are reachable (marked `@`); the other two (on the right) would require moving through a wall or unit to reach.
- Three of these reachable squares are nearest, requiring the fewest steps (only `2`) to reach (marked `!`).
- Of those, the square which is first in reading order is chosen (`+`).

The unit then takes a single step toward the chosen square along the shortest path to that square. If multiple steps would put the unit equally closer to its destination, the unit chooses the step which is first in reading order. (This requires knowing when there is more than one shortest path so that you can consider the first step of each such path.) For example:

In range:	Nearest:	Chosen:	Distance:	Step:
#####	#####	#####	#####	#####
#.E...#	#.E...#	#.E...#	#4E212#	#..E..#
#...?.# -->	#...!.# -->	#...+.# -->	#32101# -->	#.....#
#..?G?#	#..!G.#	#...G.#	#432G2#	#...G.#
#####	#####	#####	#####	#####

The Elf sees three squares in range of a target (`?`), two of which are nearest (`!`), and so the first in reading order is chosen (`+`). Under "Distance", each open square is marked with its distance from the destination square; the two squares to which the Elf could move on this turn (down and to the right) are both equally good moves and would leave the Elf `2` steps from being in range of the Goblin. Because the step which is first in reading order is chosen, the Elf moves right one square.

Here's a larger example of movement:

```
Initially:
#####
#G..G..G#
#.....#
```

```
" . . . . . "
```

```
# . . . . . #
```

```
# G . . E . . G #
```

```
# . . . . . #
```

```
# . . . . . #
```

```
# G . . G . . G #
```

```
#####
```

After 1 round:

```
#####
```

```
# . G . . . G . #
```

```
# . . . G . . #
```

```
# . . . E . . G #
```

```
# . G . . . . #
```

```
# . . . . . #
```

```
# G . . G . . G #
```

```
# . . . . . #
```

```
#####
```

After 2 rounds:

```
#####
```

```
# . . G . G . . #
```

```
# . . . G . . #
```

```
# . G . E . G . #
```

```
# . . . . . #
```

```
# G . . G . . G #
```

```
# . . . . . #
```

```
# . . . . . #
```

```
#####
```

After 3 rounds:

```
#####
```

```
# . . . . . #
```

```
# . . GGG . . #
```

```
# . . GEG . . #
```

```
u ^ ^ u
```

```
#G..G...#
#.....G#
#.....#
#.....#
#####
```

Once the Goblins and Elf reach the positions above, they all are either in range of a target or cannot find any square in range of a target, and so none of the units can move until a unit dies.

After moving (or if the unit began its turn in range of a target), the unit attacks.

To attack, the unit first determines all of the targets that are in range of it by being immediately adjacent to it. If there are no such targets, the unit ends its turn. Otherwise, the adjacent target with the fewest hit points is selected; in a tie, the adjacent target with the fewest hit points which is first in reading order is selected.

The unit deals damage equal to its attack power to the selected target, reducing its hit points by that amount. If this reduces its hit points to 0 or fewer, the selected target dies: its square becomes . and it takes no further turns.

Each unit, either Goblin or Elf, has 3 attack power and starts with 200 hit points.

For example, suppose the only Elf is about to attack:

	HP:		HP:
G....	9	G....	9
..G..	4	..G..	4
..EG.	2	-->	..E..
..G..	2	..G..	2
...G.	1	...G.	1

The "HP" column shows the hit points of the Goblin to the left in the corresponding row. The Elf is in range of three targets: the Goblin above it (with `4` hit points), the Goblin to its right (with `2` hit points), and the Goblin below it (also with `2` hit points). Because three targets are in range, the ones with the lowest hit points are selected: the two Goblins with `2` hit points each (one to the right of the Elf and one below the Elf). Of those, the Goblin first in reading order (the one to the right of the Elf) is selected. The selected Goblin's hit points (`2`) are reduced by the Elf's attack power (`3`), reducing its hit points to `-1`, killing it.

After attacking, the unit's turn ends. Regardless of how the unit's turn ends, the next unit in the round takes its turn. If all units have taken turns in this round, the round ends, and a new round begins.

The Elves look quite outnumbered. You need to determine the outcome of the battle: the number of full rounds that were completed (not counting the round in which combat ends) multiplied by the sum of the hit points of all remaining units at the moment combat ends. (Combat only ends when a unit finds no targets during its turn.)

Below is an entire sample combat. Next to each map, each row's units' hit points are listed from left to right.

Initially:

```
#####
#.G...#   G(200)
#...EG#   E(200), G(200)
#.#.#G#   G(200)
#..G#E#   G(200), E(200)
#.....#
#####
```

After 1 round:

```
#####
#..G..#   G(200)
#...EG#   E(197), G(197)
```

```
#...EG#    E(197), G(197)
#.#G#G#    G(200), G(197)
#...#E#    E(197)
#.....#
#####
```

After 2 rounds:

```
#####
#...G.#    G(200)
#..GEG#    G(200), E(188), G(194)
#.#.#G#    G(194)
#...#E#    E(194)
#.....#
#####
```

Combat ensues; eventually, the top Elf dies:

After 23 rounds:

```
#####
#...G.#    G(200)
#..G.G#    G(200), G(131)
#.#.#G#    G(131)
#...#E#    E(131)
#.....#
#####
```

After 24 rounds:

```
#####
#..G..#    G(200)
#...G.#    G(131)
#.#G#G#    G(200), G(128)
#...#E#    E(128)
#.....#
#####
```

After 25 rounds:

```
#####  
#.G...#   G(200)  
#..G..#   G(131)  
#.#.#G#   G(125)  
#..G#E#   G(200), E(125)  
#.....#  
#####
```

After 26 rounds:

```
#####  
#G....#   G(200)  
#.G...#   G(131)  
#.#.#G#   G(122)  
#...#E#   E(122)  
#..G..#   G(200)  
#####
```

After 27 rounds:

```
#####  
#G....#   G(200)  
#.G...#   G(131)  
#.#.#G#   G(119)  
#...#E#   E(119)  
#...G.#   G(200)  
#####
```

After 28 rounds:

```
#####  
#G....#   G(200)  
#.G...#   G(131)  
#.#.#G#   G(116)  
#...#E#   E(113)  
#...G.#   G(200)  
#####
```


More combat ensues; eventually, the bottom Elf dies:

After 47 rounds:

```
#####
#G....#   G(200)
#.G...#   G(131)
#.#.#G#   G(59)
#...#.#
#....G#   G(200)
#####
```

Before the 48th round can finish, the top-left Goblin finds that there are no targets remaining, and so combat ends. So, the number of full rounds that were completed is 47, and the sum of the hit points of all remaining units is $200+131+59+200 = 590$. From these, the outcome of the battle is $47 * 590 = 27730$.

Here are a few example summarized combats:

```
#####      #####
#G..#E#      #...#E#   E(200)
#E#E.E#      #E#...#   E(197)
#G.##.#  --> #.E##.#   E(185)
#...#E#      #E..#E#   E(200), E(200)
#...E.#      #.....#
#####      #####
```

Combat ends after 37 full rounds
 Elves win with 982 total hit points left
 Outcome: $37 * 982 = 36334$

```
#####
#E..EG#      #.E.E.#   E(164), E(197)
#.#G.E#      #.#E..#   E(200)
#E.##E#  --> #E.##.#   E(98)
#G..#.#      #.E.#.#   E(200)
#..E#.#      #...#.#
#####
```

Combat ends after 46 full rounds
 Elves win with 859 total hit points left
 Outcome: $46 * 859 = 39514$

```
#####
#E.G#.#      #G.G#.#   G(200), G(98)
#.#G..#      #.#G..#   G(200)
#G.#.G#  --> #..#..#
#G..#.#      #...#G#   G(95)
#...E.#      #...G.#   G(200)
#####
```

Combat ends after 35 full rounds
 Goblins win with 793 total hit points left
 Outcome: $35 * 793 = 27755$

```
#####
#.E...#      #.....#
#.#..G#      #.#G..#   G(200)
#..###.#  --> #.###.#
#E#G#G#      #.#.#.#
#...#G#      #G.G#G#   G(98), G(38), G(200)
#####
```

Combat ends after 54 full rounds
 Goblins win with 536 total hit points left
 Outcome: $54 * 536 = 28944$

```
#####  
#G.....#      #.G.....#      G(137)  
#.E.#...#      #G.G#...#      G(200), G(200)  
#..##..G#      #.G##...#      G(200)  
#...##..# --> #...##..#  
#...#...#      #.G.#...#      G(200)  
#.G...G.#      #.....#  
#.....G.#      #.....#  
#####
```

Combat ends after 20 full rounds
Goblins win with 937 total hit points left
Outcome: 20 * 937 = 18740

What is the outcome of the combat described in your puzzle input?

To begin, [get your puzzle input](#).

Answer: [\[Submit\]](#)

You can also [\[Share\]](#) this puzzle.