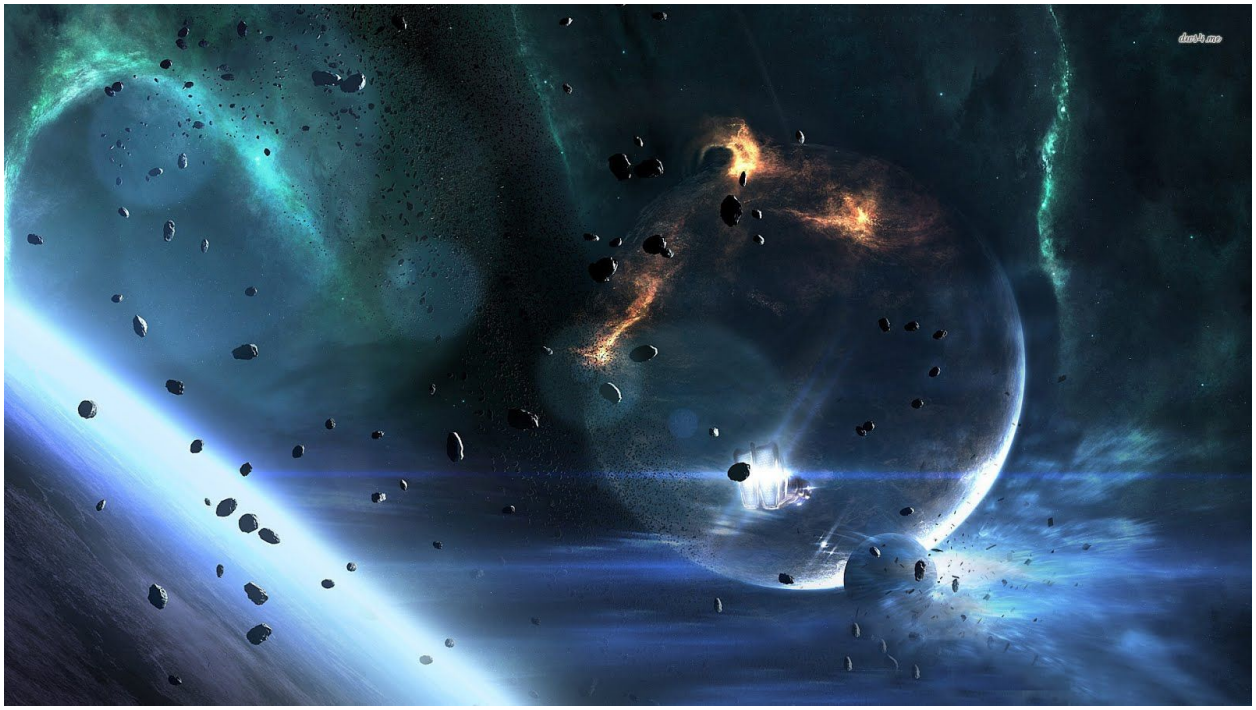
Escape from Eschaton v3

Your home world, Eschaton, is in its final days. Your official duties have finished. It's time to escape the planet's explosion and rejoin your family!



You're fortunate. You've anticipated this day, and saved an old frigate to escape in! Unfortunately your frigate doesn't have the latest navigation systems, so you're going to have to chart the escape course manually.

Problem Description

We're going to describe our course as a function of discrete time *t* which starts at *t=1* (at *t=0*, position, acceleration and velocity are all 0).

Your frigate can fire thrusters to accelerate, coast at your existing speed, or fire thrusters to decelerate at any discrete time *t*. Plotting an escape course means entering an array, indexed from time *t=1*, with instructions for the thrusters:

Fire your thrusters to accelerate...	1
Coast at your existing speed...	0
Fire your thrusters to decelerate...	-1

Remember that velocity is the integral (or in this case, the sum) of acceleration. That means after you fire your thrusters once, your ship will keep moving!

You plot your course by entering JSON into the frigate's computer, for example:

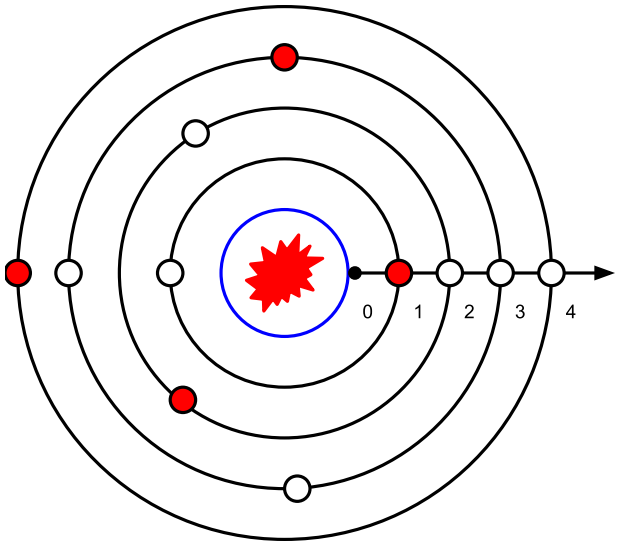
```
[0, 1, 1, -1, 0, 1]
```

The above course would drive your ship as follows:

Time	**Acceleration**	Velocity ($\sum_{0}^{t} a_t$)	Position ($\sum_{0}^{t} v_t$)
t=0	**a=0**	v=0	p=0
t=1	**a=0**	v=0	p=0
t=2	**a=1**	v=1	p=1
t=3	**a=1**	v=2	p=3
t=4	**a=-1**	v=1	p=4
t=5	**a=0**	v=1	p=5
t=6	**a=1**	v=2	p=7

Eschaton has an asteroid field circling it, and the asteroids can easily destroy your ship. The blast radius will also follow your ship at a fixed rate; see Death by Blast, below. Finally, it's possible that you can drive your ship directly into Eschaton's surface itself, in which case you will also be destroyed.

You have a chart of these fields, so you know when the asteroids will be in your path…



In the example chart to the left, there are four concentric asteroids you must pass. Each ring contains exactly one asteroid, moving clockwise as *t* increases.

Eschaton itself is always at position 0. There are no asteroids on Eschaton, but the blast will soon consume it...

The asteroids in this chart are at positions 1-4. Each asteroid has an *offset* which represents where the asteroid is at *t=0*, and a number of orientations the asteroid will successively assume as *t* increases, expressed as *t_per_asteroid_cycle*.

Your ship travels along *offset* 0. Don't worry about hitting asteroids between positions when you're going fast (*v > 1*), they are *between orientations* and therefore cannot be hit.

The above diagram is represented by the following example chart, which is also JSON:


```
{
  "t_per_blast_move": 10,
  "asteroids": [
    {
      "offset": 0,
      "t_per_asteroid_cycle": 2
    },
    {
      "offset": 1,
      "t_per_asteroid_cycle": 3
    },
    {
      "offset": 3,
      "t_per_asteroid_cycle": 4
    },
    {
      "offset": 1,
```

```
      "t_per_asteroid_cycle": 2
    }
  ]
}
```

Death by Hitting Eschaton Itself

You can hit Eschaton itself. In the above example, consider the course:

```
[-1]
```

The idea here is that you throw your ship into reverse on the launch pad.

Test data is available here: https://gist.github.com/0872aafe0bb179d81bed

Our simulator at AltSchool, running on the example chart, says:

```
time elapses: t=0 p=0 v=0
the ship accelerates: a=-1
time elapses: t=1 p=-1 v=-1
the ship hit eschaton itself: p=-1
false
```

Death by Hitting an Asteroid

You can hit an asteroid by trying to occupy the same position it does. In the above example, consider the following course:

```
[0, 1]
```

The idea here is to wait for the asteroid in the first ring to cycle to the other side, then accelerate into its path as comes back around Eschaton.

Test data is available here: https://gist.github.com/0872aafe0bb179d81bed

Our simulator at AltSchool, running on the example chart, says:

```
time elapses: t=0 p=0 v=0
the ship accelerates: a=0
time elapses: t=1 p=0 v=0
the ship accelerates: a=1
```

```
time elapses: t=2 p=1 v=1
the ship hits an asteroid: p=1
false
```

Death by Blast

You can be consumed in the blast. In the above example, consider the following course:

```
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```

The idea here is to wait for the blast time to elapse. The blast is expressed as the amount of time before each successive position can no longer be occupied, expressed as *t_per_blast_move*. In this case we waited on Eschaton's surface until *t=10*. At *t=10*, the blast radius moves to *p=1* and the surface (*p=0*) is no longer safe; at *t=20*, the blast radius moves to *p=2*, etc.

Test data is available here: https://gist.github.com/0872aafe0bb179d81bed

Our simulator at AltSchool, running on the example chart, says:

```
time elapses: t=0 p=0 v=0
the ship accelerates: a=0
time elapses: t=1 p=0 v=0
the ship accelerates: a=0
time elapses: t=2 p=0 v=0
the ship accelerates: a=0
time elapses: t=3 p=0 v=0
the ship accelerates: a=0
time elapses: t=4 p=0 v=0
the ship accelerates: a=0
time elapses: t=5 p=0 v=0
the ship accelerates: a=0
time elapses: t=6 p=0 v=0
the ship accelerates: a=0
time elapses: t=7 p=0 v=0
the ship accelerates: a=0
time elapses: t=8 p=0 v=0
the ship accelerates: a=0
time elapses: t=9 p=0 v=0
the ship accelerates: a=0
time elapses: t=10 p=0 v=0
the ship is consumed by the blast: t=10 p=0 v=0
false
```

Escaping Eschaton

You can escape Eschaton by flying out of the asteroids. In the above example, consider the following course:

```
[1, 1, 1]
```

The idea here is to fly through the gaps in the asteroids and exit the asteroid belt. Once you've cleared the last belt, you're home free!

Test data is available here: https://gist.github.com/0872aafe0bb179d81bed

Our simulator at AltSchool, running on the example chart, says:

```
time elapses: t=0 p=0 v=0
the ship accelerates: a=1
time elapses: t=1 p=1 v=1
the ship accelerates: a=1
time elapses: t=2 p=3 v=2
the ship accelerates: a=1
time elapses: t=3 p=6 v=3
the ship escapes: p=6
true
```

Challenge Submissions

Verify that your course will allow you to escape by using the AltSchool simulator:

http://eschaton.herokuapp.com/instructions/v3

Once verified, please create a **secret** gist containing just the JSON for the plotted course. For example, here's a **secret** gist of the example course tests, above:

https://gist.github.com/0872aafe0bb179d81bed

You can submit a gist anonymously online or use a command line tool to upload it.

Finally, please **email** us:

- The Gist URL.
- A **.zip* or **.tgz* file containing the code you used to solve the problem.

Take a deep breath, it's time to rejoin your family! Here are the official charts of Eschaton:

https://gist.github.com/83dbd1bcd9684cc8757d

Hint: you may want to limit the size of the chart during development.

Extra Credit

Want to really impress us? Submit an *optimal* course, one requiring the least amount of time *t* to escape the asteroid fields.