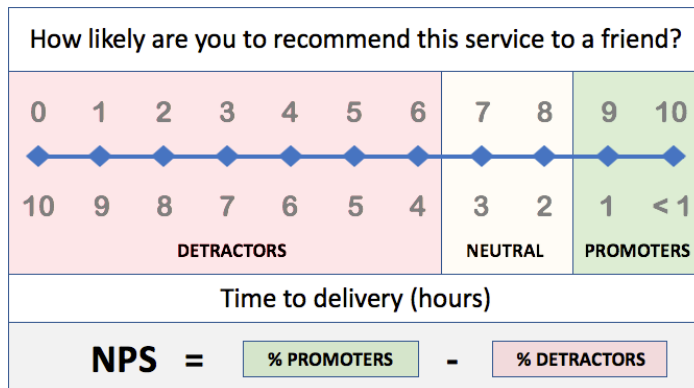


# Drone Delivery Challenge

## Overview

Implement an algorithm to schedule drone-carried deliveries to customers in a small town. The town is organized in a perfect grid, with a warehouse and drone-launch facility at the center. All deliveries originate at the warehouse and are carried by a drone to a customer location. A drone's "ground speed" is exactly one horizontal or vertical grid block per minute.

Your homework assignment is to design and write a drone launch scheduling program that maximizes the net promoter score (NPS) amongst drone-delivery customers. **Net promoter score** is defined as the percentage of promoters minus the percentage of detractors. See the diagram below to understand the relationship between delivery speed and customer feedback responses. The town owns one drone, which is allowed to operate from 6 a.m. until 10 p.m.



## Input Description

You will be given a file that contains one line of input for each order. That line includes an order identifier, followed by a space, the customer's grid coordinate, another space, and finally the order timestamp. The order identifier will have the format: **WM####**. The customer coordinate will have a North/South direction indicator (N or S) and a East/West indicator (E or W) -- for example: **N5E10**. The timestamp will have the format: **HH:MM:SS**. The orders in the input file will be sorted by order timestamp. See the **Example Input File Rows** section for an example of the input rows.

## Output Description

The program should produce a file that contains one output line for each customer order. That line should indicate the order number and the drone's departure time. The last line in the output file should contain the estimated NPS score. There is no requirement to deliver orders in sequence in which they were received.

## Requirements

- Implement your solution using a programming language that you are comfortable with. We work in Java, but we are more interested in understanding how you think than in language specifics.
- The solution and tests should build and execute entirely via the command line. The command for executing the program should accept the complete path to the input file as an argument and should return the full path to the output file.
- A README file should be included in your submission that documents your assumptions and includes instructions for building the solution and executing the tests.

Your solution will be reviewed by engineers that you will be working with if you join the Walmart Labs team. We are interested in seeing your software design, coding, and testing skills.

## Example Data

### Example Input File Rows:

```
WM001 N11W5 05:11:50
WM002 S3E2 05:11:55
WM003 N7E50 05:31:50
WM004 N11E5 06:11:50
...
```

### Example Output File Rows:

```
WM002 06:00:00
WM001 06:07:13
WM004 06:31:24
WM003 06:55:35
...
NPS 87
```