# HOMEWORK #6:

# *A Fistfull of Robots*

Due Monday, October the 15th, 6:00pm

For this assignment, you will submit the following files:

1. '`myqueue.h`': your declaration of a queue class.
2. '`myqueue.hpp`': your implementation of a queue class.
3. '`robotsim.cpp`': your simulation program.

Remember to put your name and section at the top of all files.
Your simulation program should expect all input to come from '`cin`', and all your output should be to '`cout`'.

## Problem

Every morning, robots line up at the entrance of "Mom's Robot Service & Maintenance" shop to enjoy some of the services offered. The shop offers 4 different services: **Metalworking**, to remove dents and dings; **Painting**, for a shiny new look; **Memory wiping**, for those who know too much, and **Oil Bathing**, a decadent treat. Robots receive services in the order they are lined up. A robot may want to use one or more of these services. However, only one robot can receive a particular service at the same time, and after receiving a service, if a robot has other requests, it must go back to the end of the line.

Your job is to write a program that simulates the shop, and predict when each robot receives a service, when do they leave the shop, and how much time it takes the shop to service all robots.

## Input

The input will consist of a list of the robots lined up. Each robot has a name, a number of services it wants, and the list of services it wants. The end of the list of robots is signaled with a '`.`' character.

## Output

Consider a clock that starts at 0 when the shop opens. Output the time at which each robot receives a service, moves back to the queue, or leaves the shop.

## Simulation Details:

- The services will be denoted by the strings **"**`paint`**"**, **"**`metalwork`**"**, **"**`oilbath`**"** and **"**`memwipe`**"**.
- Painting a robot takes 23 minutes.

- Metalwork takes 14 minutes to finish a robot.
- An oil bath takes 20 minutes to completion.
- A Memory Wipes takes just 8 minutes to flash a robot's mind squeaky clean.
- A robot who gets a Memory Wipe forgets the other services it wanted and leaves the shop.
- Robot names do not contain spaces.

## Sample

Input

```
Bender   1 paint
Calculon 4 metalwork memwipe metalwork paint
r2d2     2 oilbath paint
c3p0     3 memwipe oilbath paint
Hedonismbot 5 oilbath oilbath oilbath paint oilbath
TinyTin  3 metalwork paint metalwork
.
```

Output

```
At time 0 : Bender gets paint
At time 0 : Calculon gets metalwork
At time 0 : r2d2 gets oilbath
At time 0 : c3p0 gets memwipe
At time 8 : c3p0 is done!
At time 14 : Calculon gets in line for memwipe
At time 20 : r2d2 gets in line for paint
At time 20 : Hedonismbot gets oilbath
At time 20 : TinyTin gets metalwork
At time 20 : Calculon gets memwipe
At time 23 : Bender is done!
At time 23 : r2d2 gets paint
At time 28 : Calculon is done!
At time 34 : TinyTin gets in line for paint
At time 40 : Hedonismbot gets in line for oilbath
At time 46 : r2d2 is done!
At time 46 : TinyTin gets paint
At time 46 : Hedonismbot gets oilbath
At time 66 : Hedonismbot gets in line for oilbath
At time 66 : Hedonismbot gets oilbath
At time 69 : TinyTin gets in line for metalwork
```

```
At time 69 : TinyTin gets metalwork
At time 83 : TinyTin is done!
At time 86 : Hedonismbot gets in line for paint
At time 86 : Hedonismbot gets paint
At time 109 : Hedonismbot gets in line for oilbath
At time 109 : Hedonismbot gets oilbath
At time 129 : Hedonismbot is done!
```

## Implementation Guidelines:

- Your queue implementation should be a subclass of the provided "`AbstractQueue`" class.
- Implement and **test** your queue **before** you code the simulation.
- Think about how you are going to represent Robots and Services. A class 'Robot' and a class 'ServiceStation' may prove useful.
- When you grant a robot a service, keep track of when than service will be completed.
- This will prove to be a challenging homework. Start early!!
- A suggestion for your simulation algorithm is provided in the next section:

## Simulation Algorithm Draft:

```
WHILE the queue is not empty OR the services are busy.

    IF the queue is not empty
        Let 'r' be the robot at the front of the queue.
        Let 'm' be the service requested by 'r'

    IF the queue is not empty AND 'm' is available
        Give 'r' service 'm'
        Record time of completion of service 'm'
    ELSE
        Let 'w' be the service that will finish earliest.
        Let 'x' be the robot receiving 'w'
        Advance the clock to the time 'w' finishes
        IF 'x' wants more services
            place 'x' at the back of the queue
        ELSE
            'x' is done!
```