CSC209H Worksheet: Select

Consider the man page of an imaginary system call `office_hours`. Type `ps_set` is a defined type, similar to type `fd_set` that we use with `select`.

```
OFFICE_HOURS(2)            BSD System Calls Manual           OFFICE_HOURS(2)

NAME
     PS_CLR, PS_ISSET, PS_SET, PS_ZERO, office_hours

SYNOPSIS

     void PS_CLR(int ps, ps_set *psset);

     int PS_ISSET(int ps, ps_set *psset);

     void PS_SET(int ps, ps_set *psset);

     void PS_ZERO(ps_set *psset);

     int officehours(ps_set *prof, struct timeval window);

DESCRIPTION
     office_hours() examines the schedules for the professors in the
professor set prof to see which have office hours scheduled within the
given window from the current time.  office_hours() replaces the
professor set with the subset of professors who have office hours
in the given window.

RETURN VALUE
     office_hours() returns the number of professors from the ps_set
who have office hours in the upcoming window amount of time, or -1 if
an error occurs. If office_hours() returns with an error, the
descriptor sets will be unmodified and the global variable errno will
be set to indicate the error.
```

Suppose that (like file descriptors), professors are represented by small integers and there are professors defined as follows:

```
#define MICHELLE 1
#define ANDREW 2
#define JEN 3
#define ALAN 4
...   (there are more)
```

You are only interested in whether there are office hours held by Jen or Michelle in the next 5 hours. Finish the program on the other side of this page, so that it calls `office_hours` and then prints either the message `"Jen has office hours"` or the message `"Michelle has office hours"` or both messages as appropriate.
Demonstrate that you know how to properly check for errors on a system call by writing the code to give the conventional behaviour if `officehours` fails.

CSC209H Worksheet: Select

```
int main() {

    // set up the second argument to office_hours
    // (this is done for you and you do not need to set any other fields)
    struct timeval window;
    window.interval = 5 * 60;

    // set up first argument to office_hours









    // call office_hours (use window as the second parameter)









    // print the appropriate message(s)
```