Program 3

Due Date: Thursday, November 30, 2005 **Points**: 100

This program asks you to build a basic engine for a firewall. Call your program "fw".

Your program is to read in a set of rules describing how an incoming packet is to be transformed and routed. Once the rules are read in, your program will read in data representing the packets, and write out the transformed packets.

Input

Your program must be named *fw*, and must accept the following command-line syntax:

```
fw rules_file packet_file
```

where *rules_file* is the name of the file containing the rules, and *packet_file* is the name of the file containing the data representing the packets.

Each line in the *rules_file* has the following form:

```
source IP : source port : destination IP : destination port : action
```

where *action* is one of the following:

drop	discard the packet; no further rules are executed
accept	send the packet on; no further rules are executed
src_IP *new_IP*	change the source IP of the packet to *new_IP*; go to the next rule
src_port *new_port*	change the source port of the packet to *new_port*; go to the next rule
dest_IP *new_IP*	change the destination IP of the packet to *new_IP*; go to the next rule
dest_port *new_port*	change the destination port of the packet to *new_port*; go to the next rule

If the first four fields in the line match the appropriate parts of the packet, *action* occurs. If either the *source IP* or *destination IP* are the character "*", then that matches any value of the field.

Each line in the *packet_file* has the following form:

```
source IP : source port : destination IP : destination port : body
```

where *body* is any sequence of characters up to, but not including, the end of the line.

Output

Your program should copy the contents of the *packet_file* to the standard output, modifying it as indicated by the rules in the *rules_file*. If the line in the *packet_file* does not match any line in the *rules_file*, accept the packet by copying it to the standard output. Otherwise, act as the appropriate line(s) in the *rules_file* indicate. All processing is to be done in order of the rules (that is, apply the first rule; if the packet is neither accepted nor dropped, apply the second line; and so forth.)

Example

The *rules_file* contains the following:

```
129.234.6.8 : 25 : 15.4.7.8 : 25 : dest_IP 10.3.4.5
* : 25 : 10.3.4.5 : 25 : accept
* : * : 15.4.7.8 : 80 : dest_IP 10.3.4.8
* : * : 10.3.4.8 : 80 : dest_port 8080
* : * : 10.3.4.8 : 8080 : accept
* : * : * : * : drop
```

The *packet_file* contains the following:

```
129.234.6.8 : 25 : 15.4.7.8 : 25 : here is a letter
```

```
130.234.6.8 : 25 : 15.4.7.8 : 25 : here is another letter
178.23.56.77 : 3456 : 15.4.7.8 : 80 : here is a web message
```

The output should be:

```
129.234.6.8 : 25 : 10.3.4.5 : 25 : here is a letter
178.23.56.77 : 3456 : 10.3.4.8 : 8080 : here is a web message
```

Note that the second packet does not match any line except the last, and so is dropped.

Extra Credit

1. (*Easy; 15 points*) If the *packet_file* argument is "—", read the *packet_file* from the standard input.
2. (*Harder; 25 points*) Create a log file by printing the results of applying each line of the *rules_file* to each line of the *packet_file*. For each line, indicate whether there is a match; and if there is a match, what the packet looked like after the *action* took place..