Standard ACLs Practice Problems—KEY

Remember, a "0" bit in a wildcard mask means match the corresponding bit in the address, and a "1" bit in a wildcard mask means ignore the corresponding bit in the address.

1. Here are three solutions to this problem:

- access-list 2 permit host 193.5.2.76
- access-list 1 permit 193.5.2.76 0.0.0.0
- access-list 3 permit 193.5.2.76

2. One solution to this problem:

```
access-list 7 deny host 11.5.25.239
access-list 7 permit any
```

3. Since there are 254 possible host addresses on this network, we don't want to specify them individually. Therefore, we will use a wildcard mask. An access list that meets the requirements is:

```
access-list 7 permit 196.25.1.0 0.0.0.255
```

4. One solution to this problem:

```
interface e2
 ip access-group 13 in
access-list 13 deny host 104.2.64.33
access-list 13 deny host 152.5.35.83
access-list 13 permit 185.25.0.0 0.0.255.255
```

Again, we can use the keyword "host", as in "host 104.2.64.33", or we can use the mask "0.0.0.0" following a host address, such as "252.5.35.83 0.0.0.0". Note also that since the first two "denies" are covered by the implicit "deny any" that ends a standard IP access list, we can devise a more efficient solution as follows:

```
interface e2
 ip access-group 13 in
access-list 13 permit 185.25.0.0 0.0.255.255
```

5. Access list 25 has been placed inbound on interface E1. Therefore, any IP traffic from host 101.2.3.40 will be allowed into the router via E1. No IP traffic from any of the 256 Class "C" networks starting with 203.45.0.0 will be allowed into E1, but any other IP traffic will be permitted. Since traffic from host 101.2.3.40 is also permitted by the last line, the first line is superfluous, and the same result could be obtained by using:

```
access-list 25 deny 203.45.0.0 0.0.255.255
access-list 25 permit any
```

6. The most straightforward solution:

```
  interface token-ring 3/1
   ip access-group 66 out
  access-list 66 permit host 1.2.3.98
  access-list 66 permit host 1.2.3.99
```

On the other hand, we could get cute and use a wildcard mask. If we examine the bit patterns for the two host addresses, we notice that they are identical in the first three octets, and identical up to the last bit in the fourth octet, where the two possibilites are our two host addresses. Therefore, we can cover both addresses with one line, and an alternative solution is as follows:

```
  interface token-ring 3/1
   ip access-group 66 out
  access-list 66 permit 1.2.3.98 0.0.0.1
```

7. This configuration places access list 13 inbound on Token Ring interface 7. Accordingly, all IP traffic from host 201.3.4.2 is allowed in on e7, IP traffic from host addresses 203.45.0.0 through 203.45.255.255 is denied access inbound through To7, IP traffic from host addresses 84.7.22.240 through 84.7.22.247 is denied access inbound through To7, and all other IP traffic is permitted inbound through To7. Since 201.3.4.2 is a subset of the last line of access list 13, the first line of access list 13 is superfluous, and the list could be written more concisely as:

```
  access-list 13 deny 203.45.0.0 0.0.255.255
  access-list 13 deny 84.7.22.240 0.0.0.7
  access-list 13 permit any
ACL 84 prevents any traffic from the range 203.45.6.0 to 203.45.6.255
from exiting the interface.
```

8. One efficient solution:

```
  access-list 98 permit 222.111.3.0 0.0.0.255
  access-list 98 permit 222.111.4.0 0.0.3.255
The first line covers network 222.111.3.0/24, and the second line
covers networks 222.111.4.0/24 through 222.111.7.0/24.
```

9. One solution:

```
  interface ethernet 0
   ip access-group 39 out
  interface token-ring 2
   ip access-group 39 in
  access-list 39 permit 10.0.0.0 0.0.255.255
  access-list 39 permit 10.1.0.0 0.0.255.255
  access-list 39 permit 10.2.0.0 0.0.255.255
  access-list 39 permit 10.3.0.0 0.0.255.255
  access-list 39 permit 10.4.0.0 0.0.255.255
  access-list 39 permit 10.5.0.0 0.0.255.255
  access-list 39 permit 10.6.0.0 0.0.255.255
```

```
  access-list 39 permit 10.7.0.0 0.0.255.255
  access-list 39 permit 10.9.0.0 0.0.255.255
  access-list 39 permit 10.10.0.0 0.0.255.255
  access-list 39 permit 10.11.0.0 0.0.255.255
  access-list 39 permit 10.12.0.0 0.0.255.255
  access-list 39 permit 10.13.0.0 0.0.255.255
  access-list 39 permit 10.14.0.0 0.0.255.255
  access-list 39 permit 10.15.0.0 0.0.255.255
```

That solution is kind of long, but it has the advantage of being straightforward. Another solution is:

```
  interface ethernet 0
   ip access-group 49 out
  interface token-ring 2
   ip access-group 49 in
  access-list 49 deny 10.8.0.0 0.0.255.255
  access-list 49 permit 10.0.0.0 0.15.255.255
```

This is concise and relatively easy to understand. Can you see why access lists 39 and 49 give equivalent results?