Step 1: Manual GET with Telnet

Q1: What version of HTTP is the server running?

From the response in the screenshots, the server responds with: HTTP/1.1 200 OK

This confirms that the server is running **HTTP/1.1**.

Q2: How is the beginning of the content sent by the server recognized by the client?

The HTTP protocol separates the headers from the body of the response using a blank line (CRLF). This blank line marks the end of the headers and the beginning of the content.

Q3: How does the client know what type of content is returned?

The server specifies the content type in the Content-Type header field. For example: Content-Type: text/css which tells the client how to interpret the returned content.

Step 3: Inspect the Trace

Q1: What is the format of a header line?

Header lines follow the format: Field-Name: Field-Value.

For example, Host: www.washington.edu Connection: keep-alive

Q2: What headers are used to indicate the kind and length of content that is returned in a response?

Content-Type: Specifies the type of content (e.g., text/css).

Content-Length: Specifies the length of the content in bytes.

Step 4: Content Caching

Q1: What is the name of the header the browser sends to let the server work out whether to send fresh content?

The browser uses the If-Modified-Since header to let the server know the timestamp of the last cached copy of the resource.

Q2: Where exactly does the timestamp value carried by the header come from?

The timestamp comes from the Last-Modified header in the server's response. For example:

```
Last-Modified: Mon, 16 Jul 2012 05:28:05 GMT
```

Step 5: Complex Pages

For complex pages, many embedded resources (e.g., CSS, JavaScript, images) will be fetched as part of the page loading process. Each of these requests may involve headers like Referer or If-Modified-Since depending on the caching strategy.