1. Environment Variables
The web server sets various environment variables before it executes a CGI script. These variables contain information about the server, the client request, and other details.
Common environment variables include:
REQUEST_METHOD: The method used to make the request (e.g., GET, POST).
QUERY_STRING: The query string from the URL, provided for GET requests. It contains the parameters and their values passed in the URL.
CONTENT_TYPE: For POST requests, this indicates the media type of the data being sent by the client.
CONTENT_LENGTH: The length of the data (in bytes) being sent by the client in a POST request.
SCRIPT_NAME: The path to the CGI script being executed.
HTTP_USER_AGENT: Information about the user agent (web browser) making the request.
REMOTE_ADDR: The IP address of the client making the request.

①

Input Data
For POST requests, the server passes the body of the request to the CGI script through the script's standard input (stdin). This data can include form data, file uploads, or any other information sent by the client. For GET requests, parameters are passed via the QUERY_STRING environment variable instead of through standard input.

CGI script needs to:
Read the environment variables to understand the context of the request.
If it's a POST request, read from the standard input to get the body of the request.
Parse the query string or input data to extract the parameters and their values.
Perform the required action based on the request (e.g., database queries, data processing) and generate a response.

generate an appropriate HTTP response, which includes:
Sending the Content-Type header to indicate the type of the response (e.g., text/html for HTML responses).
Optionally, other HTTP headers as needed.
The body of the response, which could be HTML, JSON, plain text, or any other format appropriate for your application.

②

```cpp
#include <iostream>
#include <cstdlib> // For getenv() and system()

using namespace std;

int main() {
    // Set the HTTP header
    cout << "Content-type: text/html\n\n";

    // Start of the HTML content
    cout << "<html>\n";
    cout << "<head>\n";
    cout << "<title>Simple C++ CGI Script</title>\n";
    cout << "</head>\n";
    cout << "<body>\n";
    cout << "<h1>Hello from C++ CGI!</h1>\n";

    // Getting the QUERY_STRING environment variable
    char* query_string = getenv("QUERY_STRING");
    if (query_string) {
        cout << "<p>Query String: " << query_string <<
            "</p>\n";
    } else {
        cout << "<p>No query string provided.</p>\n";
    }

    // End of the HTML content
    cout << "</body>\n";
    cout << "</html>\n";

    return 0;
}
```

The CGI script communicates this response back to the web server, which then forwards it to the client.

③