```
#include <stdio.h>
#include <stdlib.h>
#include <curl/curl.h>
#include <cJSON.h>

// Function to handle CURL write callback
size_t write_callback(char *data, size_t size, size_t nmemb, char *buffer) {
    size_t total_size = size * nmemb;
    buffer = realloc(buffer, total_size + 1);
    if(buffer == NULL) {
        printf("Error: Unable to allocate memory.\n");
        return 0;
    }
    strncat(buffer, data, total_size);
    return total_size;
}

int main() {
    CURL *curl;
    CURLcode res;

    char *weather_api_url = "API_URL_HERE";
    char *api_key = "YOUR_API_KEY_HERE";

    char *buffer = malloc(4096 * sizeof(char));
    if(buffer == NULL) {
        printf("Error: Unable to allocate memory.\n");
        return 1;
    }
    buffer[0] = '\0';
```

```
curl = curl_easy_init();
if(curl) {
    char request_url[512];
    sprintf(request_url, "%s?apikey=%s", weather_api_url, api_key);

    curl_easy_setopt(curl, CURLOPT_URL, request_url);
    curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, write_callback);
    curl_easy_setopt(curl, CURLOPT_WRITEDATA, buffer);

    res = curl_easy_perform(curl);
    if(res != CURLE_OK) {
        printf("Error: %s\n", curl_easy_strerror(res));
        return 1;
    }

    curl_easy_cleanup(curl);

    // Parse JSON response
    cJSON *json = cJSON_Parse(buffer);
    if(json == NULL) {
        printf("Error: Failed to parse JSON.\n");
        free(buffer);
        return 1;
    }

    // Extract desired weather data from JSON
    cJSON *temperature = cJSON_GetObjectItem(json, "temperature");
    cJSON *humidity = cJSON_GetObjectItem(json, "humidity");
    cJSON *description = cJSON_GetObjectItem(json, "description");

    // Display weather data
```

```
        printf("Temperature: %.2fÂ°C\n", temperature->valuedouble);
        printf("Humidity: %.2f%%\n", humidity->valuedouble);
        printf("Description: %s\n", description->valuestring);

        // Clean up
        cJSON_Delete(json);
        free(buffer);
    }

    return 0;
}
```