```
#include "gps.h"

// Function that takes a file and field
// Fields of file must be separated by commas
// Prints contents of file with field highlighted in yellow
uint32_t gps(char str[], char field[])
{
    FILE *fp;   // Pointer to file
    uint32_t count; // Count to get to field
    int32_t num;    // Field # as an int
    char c;     // For each character in file

    // Opens file and check for an error
    fp = fopen(str, "r");
    if (fp == NULL) {
        printf("Was not able to open file %s\n", str);
        return 0;
    }

    // Converts field to int and check that it is a valid number
    num = atoi(field);
    if (num <= 0) {
        printf("Enter in valid number above 0!\n");
        return 0;
    }

    // Finds field using commas that separate fields
    // Then prints contents of file but with correct field colored
    // Resets counter every newline and stops at end of file
    count = 1;
    for (c = getc(fp); !feof(fp); c = getc(fp)) {
        if (c == '\n')
                        count = 1;
        else if (c == ',')
                        count++;

        if (count == num) {
            if (c != ',')
            {
                printf("\033[0;33m");
                printf("%c", c);
            } else {
                printf("\033[0m");
                printf("%c", c);
            }
        } else {
            printf("\033[0m");
            printf("%c", c);
        }
    }

    // Closes file
    fclose(fp);

    return 0;
}
```