Laboratorio 3: Algunas soluciones

Ejercicio 1 - Parte A

Función table_from_file

```
void table_from_file(WeatherTable a, const char *filepath) {
   FILE *file = NULL;

   file = fopen(filepath, "r");
   if (file == NULL) {
       fprintf(stderr, "File does not exist.\n");
       exit(EXIT_FAILURE);
   }

   unsigned int k_year = 0u;
   unsigned int k_month = 0u;
   unsigned int k_day = 0u;
   while (!feof(file)) {
       int res = fscanf(file, " %u %u %u ", &k_year, &k_month, &k_day);
       if (res != 3) {
           fprintf(stderr, "Invalid table.\n");
           exit(EXIT_FAILURE);
       }

       Weather weather = weather_from_file(file);

       a[k_year-FST_YEAR][k_month-1][k_day-1] = weather;
   }

   fclose(file);
}
```

Función weather_from_file

```
Weather weather_from_file(FILE* file)
{
   Weather weather;
```

```
    int res = fscanf(file,
                     " %d %d %d %u %u %u ",
                     &weather._average_temp,
                     &weather._max_temp,
                     &weather._min_temp,
                     &weather._pressure,
                     &weather._moisture,
                     &weather._rainfall);
    if (res != 6) {
        fprintf(stderr, "Invalid table.\n");
        exit(EXIT_FAILURE);
    }

    return weather;
}
```

Ejercicio 1 - Parte B

Función min_temp_min

```
int min_temp_min(WeatherTable table) {
    int result;

    result = table[0][0][0]._min_temp;
    for (unsigned int year = 0u; year < YEARS; ++year) {
        for (month_t month = january; month <= december; ++month) {
            for (unsigned int day = 0u; day < DAYS; ++day) {
                if (table[year][month][day]._min_temp < result) {
                    result = table[year][month][day]._min_temp;
                }
            }
        }
    }

    return result;
}
```

Función max_temp_max

```
void max_temp_max(WeatherTable table, int output[YEARS]) {
    for (unsigned int year = 0; year < YEARS; year++) {
        // mayor temperatura del año `year`
        int max_temp_year = table[year][0][0]._max_temp;
        for (month_t month = january; month <= december; ++month) {
            for (unsigned int day = 0u; day < DAYS; ++day) {
                if (table[year][month][day]._max_temp > max_temp_year) {
                    max_temp_year = table[year][month][day]._max_temp;
                }
            }
        }

        output[year] = max_temp_year;
    }
}
```

Función month_max_rainfall

```
void month_max_rainfall(WeatherTable table, month_t output[YEARS]) {
    for (unsigned int year = 0; year < YEARS; year++) {
        month_t month_max = january;
        unsigned int max_rainfall = 0;
        for (month_t month = january; month <= december; ++month) {
            // calcular suma de lluvias para el mes `month`
            unsigned int month_rainfall = 0;
            for (unsigned int day = 0u; day < DAYS; ++day) {
                month_rainfall += table[year][month][day]._rainfall;
            }

            // ver si es el máximo para este año
            if (month_rainfall > max_rainfall) {
                // guardar nuevo máximo y su mes
                max_rainfall = month_rainfall;
                month_max = month;
            }
        }

        output[year] = month_max;
    }
}
```