



Programação 2017/2018

Mestrado em Engenharia Electrotécnica e de
Computadores (MEEC)

WarmingUp



Projecto Final de Programação

1 Introdução

Com o projecto de programação designado de *WarmingUp* pretende-se avaliar uma das maiores ameaças dos nossos tempos que é o aquecimento global (e percebermos se é um mito ou não!). O aquecimento global é o processo de aumento da temperatura média da atmosfera da Terra (e dos oceanos) causado por uma série de atividades humanas, especialmente a queima de combustíveis fósseis e a eliminação de florestas. O aumento das temperaturas globais desencadeia alterações importantes, tais como elevação dos níveis dos mares e o degelo das calotas polares e glaciares nas montanhas. Também provocam alterações climáticas importantes, como uma maior ocorrência de enchentes e secas graves, maiores e mais intensas tempestades, desertificação, etc. Em Portugal, a existência de secas graves com impacto na produção agrícola e a redução da quantidade de água potável para consumo humano são fenómenos atribuídos ao aquecimento global.

O objectivo deste projecto é descobrir padrões de aquecimento (ou arrefecimento) global a partir de um conjunto de dados reais, que foram obtidos de <http://berkeleyearth.org/data/>, tais como:

- Qual foi o aumento da temperatura média na terra?

- Qual foi o aumento da temperatura média num dado país?
- Qual foi o aumento da temperatura média numa dada cidade?
- Quais são os países (ou cidades) mais quentes num dado ano?
- Quais são os países (ou cidades) mais frios num dado ano?
- Quais são os países (ou cidades) com temperaturas mais extremas num dado ano?
- Quais são as cidades mais quentes (e frias) em cada país?

2 Dados de Entrada

Para realizar os objetivos deste projecto vão ser disponibilizados dados reais adquiridos em várias estações meteorológicas e colecionados pelo Laboratório Nacional Lawrence Berkeley. Os primeiros dados foram obtidos por técnicos usando termômetros de mercúrio e por isso com elevado erro; actualmente são utilizados termómetros electrónicos com uma elevada precisão. O estudo da temperatura da superfície da terra de Berkeley combina 1,6 bilhões de relatórios de temperatura de 16 arquivos preexistentes. Para este projecto, foram escolhidos um conjunto de dados interessantes (por exemplo, por país).

Os dados encontram-se disponíveis no formato *comma-separated values* (CSV) que é um formato muito habitual para a partilha de dados em forma de tabela. Cada linha do ficheiro é um item da tabela e cada item possui um ou mais campos, separados por vírgula. Para a realização deste projecto vão ser disponibilizados dois ficheiros, um com informação agrupada por país e outro com informação agrupada por cidade. Os ficheiros contêm a seguinte informação:

- Data da medida de temperatura
- Temperatura média nos últimos 30 dias, i.e. por mês
- Incerteza na medida da temperatura média
- País
- Cidade (apenas no ficheiro *tempcities.csv*)
- Latitude (apenas no ficheiro *tempcities.csv*)
- Longitude (apenas no ficheiro *tempcities.csv*)

Os seguintes ficheiros estão disponíveis:

1. *tempcountries.csv: Global Land Temperatures By Country* – Ficheiro com as medidas por país
2. *tempcities.csv: Global Land Temperatures By Major City* – Ficheiro com as medidas por cidade

Note-se também que os ficheiros são obtidos de fontes reais e podem ter linhas em que certos dados estão ausentes. O vosso programa deve detectar esses casos, e efectuar um tratamento adequado.

3 Funcionamento

O programa *WarmingUp* deve ter dois modos de funcionamento: o modo textual (usando sempre a consola) e o modo gráfico (usando a biblioteca SDL2) e. No modo gráfico deve ser possível visualizar o aumento de temperatura global no planeta e no modo textual deve ser possível calcular um conjunto de estatísticas sobre o aquecimento (ou arrefecimento) global.

O programa deve receber como parâmetros (que correspondem a argumentos da função *main*) dois ficheiros com os formatos acima indicados. Além disso, também é necessário receber como parâmetro uma opção `-g` ou `-t` para indicar se o programa vai funcionar em modo textual ou em modo gráfico. Os parâmetros de entrada podem aparecer por qualquer ordem e o argumento seguinte à opção `-f1` deve ser o ficheiro com as medidas por país e o argumento seguinte à opção `-f2` o ficheiro com as medidas por cidade. Por exemplo, para iniciar o programa podem ser escritos um dos seguintes comandos:

```
./WarmingUp -g -f1 tempcountries.csv -f2 tempcities.csv  
./WarmingUp -f1 tempcountries.csv -f2 tempcities.csv -t  
./WarmingUp -f2 tempcities.csv -f1 tempcountries.csv -t
```

3.1 Modo Textual

No modo textual deve ser apresentado o seguinte menu *principal* para o utilizador escolher uma das opções:

1. Filtragem de dados
2. Histórico de temperaturas
3. Análise da temperatura por ano
4. Análise da temperatura global

Cada uma destas opções é descrita de seguida.

Filtragem de dados

Esta opção pergunta ao utilizador se as listagens ou estatísticas a imprimir (opções seguintes) serão sobre todos os dados carregados dos ficheiros ou apenas sobre os dados que cumpram um ou mais critérios:

1. Intervalo de tempo: O utilizador escolhe um mês e ano. Todos os dados com mês e ano inferior devem ser eliminados.
2. Mês: O utilizador escolhe o mês inicial e o mês final (1 – Janeiro, 12 – Dezembro). Todos os dados que não incluam os meses selecionados devem ser eliminados.

Estes critérios devem ser escolhidos através de um menu:

1. Limpa critérios
2. Escolhe intervalo para análise
3. Escolhe mês para análise

Note-se que quando o utilizador estabelecer um ou mais critérios, todas as análises (opções 2-4 do menu principal) devem ser realizadas apenas com os dados que cumprem todos os critérios. Esta filtragem de dados deve ser aplicada sobre os dados carregados de ambos os ficheiros.

Histórico de temperaturas

Neste modo devem ser impressas as temperaturas mínimas, máximas e médias que cumprem os critérios definidos na seleção de dados e usando um período especificado pelo utilizador. Este período pode

representar um ou mais anos, sobre o qual vão calcular o valor máximo, médio e mínimo; por exemplo, caso se escolha 10 anos, é necessário mostrar as temperaturas máxima, média e mínima de 1900-1910, 1910-1920, ... obtendo assim um gráfico com o período de 10 anos (período de amostragem).

Assim, devem perguntar ao utilizador qual é o período de tempo T (em anos) e apresentar o seguinte menu:

4. Global: aplica-se a todos os países
5. Por país: aplica-se a um único país
6. Por cidade: aplica-se a uma única cidade

Caso seja seleccionada a opção 2 ou 3, devem perguntar o país ou a cidade sobre o qual o utilizador pretende obter informação. Para as opções de 1-2 devem usar os dados do ficheiro tempcountries.csv e para a opção 3 os dados do ficheiro tempcities.csv.

Em relação aos resultados, devem mostrar uma página de 20 entradas e para avançar deve ser carregado em a e para sair em q , caso haja necessidade de mostrar mais do que uma página. Não pode haver mais do que um triplete (mínima, máxima e média) de temperaturas por linha. Cada linha deverá representar um período T e devem colocar o ano inicial e ano final (do intervalo) por linha, logo no início na linha.

Análise da temperatura por ano

Neste modo deve ser perguntado ao utilizador qual o ano que se pretende analisar. De seguida, deve ser mostrado o seguinte menu:

1. Análise por país
2. Análise por cidade

No caso do utilizador escolher a opção 1 (análise por país: ficheiro tempcountries.csv), a seguinte informação deve ser calculada e impressa (utilize três colunas distintas) para o ano introduzido:

- Os N países mais quentes nesse ano.
- Os N países mais frios nesse ano.
- Os N países com as temperaturas mais extremas nesse ano.

No caso do utilizador escolher a opção 2 (análise por cidade: tempcities.csv), a seguinte informação deve ser calculada e impressa (utilize três colunas distintas) para o ano introduzido:

- As N cidades mais quentes nesse ano.
- As N cidades mais frias nesse ano.
- As N cidades com as temperaturas mais extremas nesse ano.

O valor de N deve ser perguntado ao utilizador (e poderá variar de 1 a 20) e deve calcular a temperatura média ao longo desse ano (todos os meses) para descobrir os países (ou cidades) mais quentes e mais frios. Para obter os países (ou cidades) com as temperaturas mais extremas, devem calcular a maior diferença de temperatura que corresponde à diferença da temperatura do mês com a temperatura mais baixa, com o mês com a temperatura mais alta.

Análise da temperatura global

Neste modo deve ser calculado o aquecimento (ou arrefecimento) global utilizando o conceito de *moving average* (MA) para eliminar flutuações de temperaturas dos dados. A MA representa o cálculo da média nas últimas M medidas e é calculado por cada medida registada. Para se conhecer o valor de M, pergunta-se ao utilizador quantos meses pretende utilizar no cálculo do MA, incluindo-se todas as medidas durante esse período. Assim, todos os dados das temperaturas devem ser recalculados utilizando o conceito de *moving average*.

De seguida, e utilizando estes novos dados, a seguinte informação deve ser impressa:

1. Aumento da temperatura global do planeta: Calculado como o aumento da temperatura em 1860, 1910, 1960, 1990 e 2013. O aumento da temperatura é calculado como a diferença entre o ano com a menor temperatura média e o ano com a maior temperatura média incluindo todos os anos anteriores. Realize a média da temperatura de todos os países presentes nos dados.
2. Aumento da temperatura global de um país: O mesmo que o ponto 1, mas apenas incluindo os dados de um só país.
3. Aumento da temperatura global de uma cidade: O mesmo que o ponto 1, mas apenas incluindo os dados de um só cidade.

Naturalmente, devem perguntar ao utilizador qual o país e a cidade para os pontos 2 e 3, respetivamente. No ponto 3 devem utilizar os dados do ficheiro tempcities.csv enquanto para os pontos 1-2 os dados do ficheiro tempcountries.csv.

3.2 Modo Gráfico

O objetivo do modo gráfico (SDL2) é mostrar os dados obtidos do ficheiro tempcities.csv no mapa do planeta terra, e assim mostrar a informação graficamente. No modo gráfico, vai ser ilustrada a evolução da temperatura nas cidades, o que corresponde ao Histórico de Temperaturas (opção *Por Cidade*) mas para todas as cidades simultaneamente. A interface gráfica deve obedecer ao seguinte cenário:

- Carregar um mapa do planeta terra, como uma imagem (por exemplo PNG).
- Mostrar um círculo preenchido com uma cor, por cada cidade, para representar a evolução da temperatura. O círculo tem de estar situado no local da cidade e a cor deve refletir a temperatura da cidade que foi obtida do ficheiro (azul – frio, vermelho – quente).
- Devem efectuar uma simulação temporal, isto é, a cor de cada círculo deve variar à medida que o tempo vai passando, por exemplo cada 2 anos corresponde a 30ms, o que significa que por cada 30ms passados, uma nova cor é atribuído ao círculo.

Opcionalmente podem realizar as seguintes melhorias ao modo gráfico: sempre que se seleccionar uma cidade com o rato mostra-se o valor da temperatura desta no lado direito da aplicação e zoom mediante interação com o utilizador. Também podem utilizar o modo transparência para melhorar o aspecto deste modo.

Podem utilizar as funções de mostrar imagens, desenhar círculos e mostrar texto do projecto intermédio.

4 Estruturas de Dados

No desenvolvimento deste jogo, é obrigatório obedecerem às seguintes regras:

- Todos os dados dos ficheiros devem ser representados através de estruturas de dados.
- Devem usar uma lista (escolham o tipo mais adequado) para representar os dados dos ficheiros.
- Devem usar alocação dinâmica de memória para reservar a memória necessária, tanto para as listyas como para vectores (que eventualmente necessitem de usar).
- A seleção de dados deve ser implementada apagando nós da lista, de forma a poupar memória e tornar o processo de percorrer a lista mais rápido.
- A lista que representam os dados por país (tempcountries.csv) deve estar ordenada por data.
- Opcionalmente, a lista dos dados por cidade, pode estar organizada num vetor de listas ou numa lista de listas para facilitar a extração da informação necessária.

Também é importante salientar que o programa desenvolvido deverá ser estruturado em múltiplos ficheiros que permitam uma organização adequada do código.

5 Desenvolvimento do Programa

Também é fundamental que os alunos cumpram as regras que se seguem no desenvolvimento do jogo.

5.1 Desenvolvimento Faseado

O desenvolvimento deste projecto deverá ser feito de uma forma faseada, devendo os alunos garantir que todas as funcionalidades codificadas até ao momento estão a funcionar corretamente.

É preferível um programa que implementa poucas funcionalidades, mas que funcionam corretamente, do que um programa totalmente desenvolvido mas que faz muito pouco.

Assim sugerem-se os seguintes passos, pela ordem apresentada, para realização do projecto. Note-se que esta lista não é exaustiva. Desta forma, é necessário realizar as seguintes tarefas:

1. Criação de uma estrutura de dados que possa ser utilizada para guardar os dados de ambos os ficheiros.
2. Leitura parâmetros do *main* e dos ficheiros necessários ao funcionamento do programa.
3. Construção dos menus e leitura dos dados introduzidos pelos utilizadores.
4. Leitura do ficheiro tempcountries.csv, inserindo os dados de uma forma ordenada numa lista
5. Implementação das opções 1-2 do modo de listagem de temperaturas.
6. Implementação da filtragem de dados, removendo elementos da lista de temperaturas.
7. Leitura do ficheiro tempcities.csv, inserindo os dados numa lista ou num vector de listas (ou lista de listas).
8. Implementação da opção 3 do modo de listagem de temperaturas.
9. Implementação do modo da análise de temperatura por ano.
10. Implementação do modo de análise de temperatura global.

11. Implementação do modo gráfico.
12. Eventuais melhorias ao modo gráfico da aplicação.

No final de cada passo devem verificar a execução correcta do programa. Os alunos deverão garantir a robustez da aplicação verificando todos os casos de erro (por exemplo quando um parâmetro de entrada não seja válido).

5.2 Documentação

O código produzido pelos alunos deverá ser comentado. Os comentários presentes no código deverão explicitar e explicar o funcionamento da aplicação assim como as decisões tomadas. As seguintes regras devem ser cumpridas:

- O código deve ser comentado sempre que realize alguma operação não óbvia.
- Os comentários devem ser claros, gramaticalmente corretos e usando frases simples.
- A declaração de todas as variáveis e constantes deve ser acompanhada de um comentário com uma breve descrição de para que servem.
- Cada bloco de código (seleção ou repetição) deve ser precedido de um breve comentário explicativo.
- Todos os programas devem ter um comentário inicial que identifique, no mínimo, o trabalho, os seus autores, o fim a que se destina e a data de realização.

5.3 Indentação

Um ponto fundamental na organização de escrita de código é a indentação, isto é, organização hierárquica das linhas de código, de acordo com âmbito onde elas se encontram. A indentação deixa o código fonte do programa mais organizado, mais legível, fácil de entender e de modificar, sendo uma parte essencial do trabalho.

5.4 Estrutura do Código

Todos os programas em C devem possuir a mesma estrutura genérica, composta pelas seguintes secções:

- Bloco de comentários.
- Diretivas `#include`.
- Constantes globais e variáveis globais (caso sejam necessárias).
- Declaração de funções.
- Função `main()`.
- Definição de funções.

Como regra geral deve considerar que as funções devem caber num único ecrã, isto é, devem ter no máximo cerca de 30 linhas. Também deve cumprir as seguintes regras:

- Inicialize sempre as variáveis na sua declaração.
- Teste a validade dos parâmetros recebidos por uma função.
- Declare constantes e evite usar números no corpo das funções.
- Evite repetições de código, use funções, ciclos, etc.
- Evite o uso de variáveis globais.

- Não use **goto**.
- Escreva código simples e claro que um colega seu possa perceber!

5.5 Compilação

O compilador a usar na execução do projecto é o gcc em ambiente Linux. **Os projectos que não compilem, i.e. que tenham erros de sintaxe, não serão avaliados.** A existência de avisos durante a fase de compilação poderá ser indício da existência de problemas no código. Estes deverão ser eliminados corretamente ou ignorados com cuidado extremo.

5.6 Decisões do Projecto

Como em qualquer projecto de informática, o funcionamento do programa não está totalmente definido no enunciado, existindo algumas ambiguidades e omissões. Para resolver essas omissões os alunos deverão tomar algumas decisões aquando do desenvolvimento do projecto. Estas decisões devem ser fundamentadas, sem nunca ir contra o definido no enunciado.

5.7 Biblioteca SDL

Durante o desenvolvimento deste projecto devera ser usada a biblioteca SDL2. A aplicação deverá ser compilada usando as bibliotecas SDL2, SDL2_image, SDL2_ttf. Mais informação disponível aqui:

- <http://wiki.libsdl.org/APIByCategory>
- <https://wiki.libsdl.org/FrontPage>

6 Submissão

Os alunos deverão submeter o código desenvolvido através do sistema FENIX até 24 de Maio. Devem entregar o código correspondente ao programa desenvolvido, nomeadamente os ficheiros de texto com extensão **.c** e **.h** e outros ficheiros necessários ao funcionamento do programa (por exemplo, fontes e imagens). Não utilize um processador de texto (e.g. Word) para formatar o seu programa.

7 Plágio

Os trabalhos serão objeto de um sistema de deteção de plágio. Os alunos podem conversar entre si para discutir possíveis soluções para algum problema que tenham mas não podem partilhar código fonte. Nesta entrega final, o projeto deve ser realizado em grupos de dois alunos! Se uma cópia for detetada todos os alunos envolvidos na cópia serão penalizados.

8 Avaliação

A avaliação do projeto terá em conta diversos parâmetros:

- Funcionalidades implementadas
- Uso correto de estruturas de dados
- Criação de listas, filas ou pilhas de forma adequada aos objetivos do projeto
- Eficiência e qualidade da implementação das funcionalidades do projeto

- Qualidade do aspecto gráfico
- Qualidade do código produzido
- Estruturação da aplicação
- Comentários e legibilidade do código
- Tratamento de erros