

Instituto Superior de Engenharia de Lisboa
Licenciatura em Engenharia Informática e de Computadores
Programação de Sistemas Computacionais
Inverno de 2024/2025
Série de Exercícios

Realize os exercícios seguintes usando a linguagem C. Deve testar devidamente o código desenvolvido, bem como apresentá-lo de forma cuidada, apropriadamente indentado e comentado. Assegure-se de que o compilador não emite qualquer aviso sobre o seu código, com as opções `-Wall -pedantic` activas. Contacte o docente se tiver dúvidas. Encoraja-se a discussão de problemas e soluções com outros colegas.

1. Programe a função `int_to_string` que representa em texto, na base de numeração **base** (considere apenas as bases 2, 8, 10 e 16), o valor inteiro recebido em **value**. O texto é depositado no *array* de caracteres indicado pelo ponteiro **buffer**, no formato de *string* C. O parâmetro **buffer_size** indica a dimensão desse *array*. A função retorna a dimensão da *string* produzida ou zero no caso da dimensão do *array* não ser suficiente. As representações nas bases 2, 8 e 16 devem ser prefixadas com as respectivas notações.

```
size_t int_to_string(unsigned value, int base, char buffer[], size_t buffer_size);
```

2. Programe a função `float_to_string` que representa em texto, o valor real contido em **value**. A representação deve ser na base decimal com uma aproximação de seis casas decimais. O texto é depositado num *array* de caracteres, nas condições definidas no exercício 1. Na programação desta função não deve utilizar operações sobre valores do tipo *float*, deve apenas utilizar operações sobre valores inteiros. A função retorna a dimensão da *string* produzida ou zero no caso da dimensão do *array* não ser suficiente.

```
size_t float_to_string(float value, char buffer[], size_t buffer_size)
```

3. Programe a função `mini_snprintf` segundo a definição da função `sprintf` da biblioteca normalizada da linguagem C, limitada aos especificadores de conversão **c**, **s**, **d**, **x** e **f**, sem adornos. Na programação desta função não deve utilizar a função `sprintf`. Consulte o exemplo da secção 7.3 do livro *The C Programming Language*. O texto produzido é depositado num *array* de caracteres, nas condições definidas no exercício 1. A função retorna a dimensão da *string* produzida ou zero no caso da dimensão do *array* não ser suficiente.

```
size_t mini_snprintf(char *buffer, size_t buffer_size, const char *format, ...);
```

4. Programe a função `time_to_string` que converte a informação de calendário, recebida no parâmetro **tm**, ponteiro para uma *struct* do tipo `struct tm`¹, para uma representação em texto, no formato "**ww, dd-mm-yyyy, hh:mm:ss**". Por exemplo: "**segunda-feira, 16-09-2024, 12:45:23**". O texto produzido é depositado num *array* de caracteres, nas condições definidas no exercício 1. A função retorna a dimensão da *string* produzida ou zero no caso da dimensão do *array* não ser suficiente. (Os caracteres com acento ou o 'ç' ocupam dois bytes na codificação UTF-8.)

```
struct tm {  
    int    tm_sec;  
    int    tm_min;  
    int    tm_hour;  
    int    tm_mday;  
    int    tm_mon;  
    int    tm_year;  
    int    tm_wday;  
    int    tm_yday;  
    int    tm_isdst;  
};
```

```
size_t time_to_string(struct tm *tm, char *buffer, size_t buffer_size);
```

¹ <http://www.cplusplus.com/reference/ctime/tm/>

5. Realize um programa para visualização do conteúdo de ficheiros de dados em formato CSV². O programa deve desenhar um quadriculado com um número de linha e colunas igual ao do ficheiro e escrever em cada quadrícula o valor do respetivo campo.

```
$ ./csv_show <option> <filename>
```

Descrição das opções:

- o <filename>** escrever o resultado num ficheiro com o nome <filename>;
em caso de omissão usar **stdout**;
- a <alignment>** indica o alinhamento do texto nas quadrículas.
Se <alignment> for 'l' alinha à esquerda, se for 'r' alinha à direita;
em caso de omissão alinha à direita;

Deve utilizar a função [getopt](#) no processamento das opções. A utilização desta função permite que as opções possam ser escritas em qualquer posição relativamente aos parâmetros.

O programa deve procurar o ficheiro de entrada na diretoria indicada pela variável de ambiente **CSV_FILE_PATH**. No caso desta variável não estar definida procura na diretoria corrente.

Em situações de erro, o programa deve devolver um valor diferente de zero. As situações de erro são dificuldades no acesso a ficheiros ou argumentos do programa inválidos.

Exemplo:

Considerando o ficheiro **stock.csv**, localizado em **/usr/home/data**, com o seguinte conteúdo:

```
máquina lavar roupa,Indesit,Lisboa,L3
máquina lavar loiça,Bosch,Porto,D4
torradeira,Moulinex,LISBOA,A10
microondas,Samsung,Porto,E4
```

Definir a variável de ambiente:

```
$ export CSV_FILE_PATH=/usr/home/data
```

O seguinte comando:

```
$ csv_show stock.csv -a l
```

produz no terminal o seguinte resultado:

```
+-----+-----+-----+-----+
| máquina lavar roupa | Indesit | Lisboa | L3 |
+-----+-----+-----+-----+
| máquina lavar loiça | Bosch  | Porto  | D4 |
+-----+-----+-----+-----+
| torradeira          | Moulinex | LISBOA | A10 |
+-----+-----+-----+-----+
| microondas          | Samsung | Porto  | E4 |
+-----+-----+-----+-----+
```

² https://en.wikipedia.org/wiki/Comma-separated_values