

Instituto Superior de Engenharia de Lisboa  
Licenciatura em Engenharia Informática e de Computadores  
**Programação de Sistemas Computacionais**  
Inverno de 2021/2022  
Série de Exercícios 1

---

Realize os exercícios seguintes usando a linguagem C. Não se esqueça de testar devidamente o código desenvolvido, bem como de o apresentar de forma cuidada, apropriadamente indentado e comentado. Assegure-se de que o compilador não emite qualquer aviso sobre o seu código, mesmo com a opção `-Wall` activa. Contacte o docente se tiver dúvidas. Não é necessário relatório. Encoraja-se a discussão de problemas e soluções com outros colegas, mas a partilha directa de soluções leva, no mínimo, à anulação das entregas de todos os envolvidos.

1. Implemente, em *linguagem C*, a função **check\_for\_pattern** que retorna um valor booleano (0 false, 1 true) que indica se existe em **value**, alguma sequência de bits igual à sequência entre o *bit* mais significativo com o valor um e o bit menos significativo com o valor um de **pattern**. (Exemplos: `pattern = 5` e `value = 0xa` retorna true; `pattern = 0xc` e `value = 3` retorna true; `pattern = 0xb` e `value = 0x2d` retorna true )

```
int check_for_pattern(int value, int pattern);
```

2. Escreva a função **get\_max\_index** que procura, no *array* recebido em **array**, o maior valor positivo. O parâmetro **array\_size** indica a dimensão do *array*. A função **get\_max\_index** produz dois resultados: o maior valor encontrado, que devolve como valor da função e a posição do *array* onde foi encontrado, que devolve através do ponteiro **index\_ptr**.

```
int get_max_index(int array[], size_t array_size, int *index_ptr);
```

3. Implemente, em linguagem C, a função **string\_find\_substitute** que procura em **string** as ocorrências da *substring* **string1** e substitui-as pela *substring* **string2**. Admita que a memória de suporte a **string** é suficiente para suportar o aumento da dimensão de **string**, que ocorre quando a dimensão de **string2** é superior à dimensão de **string1**.

Na programação desta função deve utilizar funções da biblioteca, tais como **strncmp**, **strlen**, **memmove**, **memcpy** ou outras.

```
void string_find_substitute(char *string, char *string1, char *string2);
```

4. Escreva a função **round\_to\_int**, que devolve o valor do tipo **int**, resultante do arredondamento do valor do tipo **float** que recebe em parâmetro.

Na implementação interna só podem ser utilizadas operações aritméticas e lógicas sobre inteiros. Qualquer operação de vírgula flutuante invalida a resolução do exercício.

```
int round_to_int(float value);
```

5. Realize o programa utilitário **file\_dump**, que apresenta o conteúdo de um ficheiro no *standard output*. O ficheiro pode ser indicado por argumento do programa ou recebido via *standard input*. O programa deve comportar-se com funcionalidade idêntica à do utilitário **hexdump** quando invocado com a opção **-C** (experimentar: `$ hexdump -C filename`).

Data recomendada de entrega: 6 de novembro de 2021

ISEL, 10 de outubro de 2021