

Instituto Superior de Engenharia de Lisboa
Licenciatura em Engenharia Informática e de Computadores
Programação em Sistemas Computacionais
Verão de 2016/2017

Série de Exercícios #1 - Individual

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. Escreva a função `findsaw` que procura, no inteiro `value`, a mais longa sequência de *bits* consecutivos com valores alternados. A função retorna apenas o índice do primeiro *bit* dessa sequência, sendo 0 o índice do *bit* de menor peso. No caso de haver mais do que uma resposta possível, deve ser retornada a que corresponder ao menor índice. Por exemplo, a chamada `findsaw(0xEAEA476BL)` deve retornar 23.

```
unsigned int findsaw(int value);
```

2. Implemente a função `strtrim`, que não existe implementada na biblioteca *standard* da linguagem C, sem recorrer a outras funções dessa biblioteca. A função elimina os caracteres brancos (espaços, *tabs* ou mudanças de linha) existentes no início e no fim da string passada como argumento, retornando o endereço do primeiro carácter da *string* resultante. A implementação deve procurar realizar o menor número possível de escritas na memória.

```
char *strtrim(char *str);
```

Escreva um programa de teste que, para cada linha lida do *standard input*, apresenta no *standard output* o resultado do respectivo processamento pela função `strtrim`. Cada linha apresentada no *standard output* deverá ser precedida por "`nnn: len1 -` " e sucedida por " `- len2`", sendo `nnn` o número da linha na entrada e `len1` e `len2` o número de caracteres antes e depois da linha ser processada por `strtrim`, respectivamente. Leia as linhas do *standard input* com a função *standard* `fgets`, usando `stdin` como terceiro argumento. O programa deve terminar quando ocorrer a primeira linha vazia no *standard input*. No programa de teste poderá recorrer a funções da biblioteca *standard* da linguagem C.

3. Escreva a função `float_with_int_cmp`, que compara o valor da parte inteira de `fvalue` com o valor inteiro `ivalue`. A função devolve um valor positivo se a parte inteira de `fvalue` for maior que `ivalue`, um valor negativo se a parte inteira de `fvalue` for menor que `ivalue` ou zero se forem iguais. Na implementação da função só podem ser utilizadas operações aritméticas e lógicas sobre inteiros. Qualquer operação de vírgula flutuante invalida o exercício.

```
int float_with_int_cmp(float fvalue, int ivalue);
```

4. Cada instância do tipo **StudentNode** contém informação sobre um aluno e contém ainda um campo (**next**) que permite a ligação de várias instâncias deste tipo em lista simplesmente ligada. A função **groupStudentByClass** recebe um *array* de instâncias de **StudentNode**, que não estão inicialmente ligadas em listas, e adiciona cada uma das instâncias à lista correspondente à turma (**class**) do aluno. As listas são indicadas pelo *array* **classes**, em que cada índice do *array* corresponde directamente ao número da turma e cada posição contém um ponteiro para o primeiro nó da respectiva lista. Implemente a função **groupStudentByClass**.

```
typedef struct student_node {
    unsigned number;
    const char *name;
    unsigned class;
    struct student_node *next;
} StudentNode;
```

```
void groupStudentByClass(StudentNode *classes[], StudentNode students[],
                        size_t num_students);
```

5. Escreva o programa utilitário **with**, que copia para o *standard output* as linhas do *standard input* que contiverem pelo menos uma das *strings* passadas como argumento ao programa na linha de comando.

Data limite de entrega: 9 de Abril de 2017

ISEL, 17 de Março de 2017