

**Programação em Sistemas Computacionais**

Primeira série de exercícios – Inverno de 2013/2014

---

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. Contacte o docente se tiver dúvidas. Não é necessário relatório. Encoraja-se a discussão de problemas e soluções com outros colegas, mas recorda-se que a partilha directa de soluções leva, no mínimo, à anulação das entregas de todos os envolvidos.

1. Admitindo uma arquitectura que utiliza representação de complemento para 2 para os valores inteiros com sinal, escreva a função `max_value`, que retorna o maior valor positivo que pode ser armazenado no número de *bytes* indicado pelo parâmetro `nbytes`, retornando zero se o valor não for representável num `unsigned long`. O parâmetro `sign` indica se os valores a armazenar podem ser negativos. Valorizam-se implementações que tenham em conta o número de *bits* por *byte* indicado por `CHAR_BIT`.

```
unsigned long max_value(bool sign, size_t nbytes);
```

2. A função `strtrim` remove caracteres brancos do início e do fim da *string* `src`, deixando o resultado em `dst` e retornando o número de caracteres removidos. Consideram-se caracteres brancos aqueles para os quais `isblank` não retorna 0. Desenvolva a função `strtrim` sem recorrer a outras funções da biblioteca *standard* do C, salvo `isblank`. Valorizam-se as implementações que minimizem o número total de operações de escrita por chamada.

```
size_t strtrim(char * dst, const char * src);
```

3. Construa a função `s2f`, que converte um `short int` num `float` utilizando apenas operações aritméticas e lógicas sobre inteiros.

```
typedef struct floatbits {  
    unsigned significand : 23; unsigned exponent : 8; unsigned sign : 1;  
} floatbits_t;  
  
float s2f(short int s) { floatbits_t f; /* ... */ return *(float*)&f; }
```

4. A estrutura `StudentNode` contém informação sobre um aluno e contém ainda um campo (`next`) que permite a ligação em lista de várias instâncias. A função `buildStudentSections` 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 do aluno. As listas são indicadas pelo *array* `sections`, em que cada índice do *array* corresponde directamente ao valor da turma (`section`) e contém um ponteiro para o primeiro nó da respectiva lista. Implemente a função indicada.

```
typedef struct stnode {  
    unsigned number; char * name; unsigned section; struct stnode * next;  
} StudentNode;  
  
void buildStudentSections(StudentNode * sections[],  
                          StudentNode students[], size_t num_students);
```

Data limite de entrega: 14 de Outubro de 2013

*Bom trabalho!*