

Os exercícios 1 a 3 deverão ser resolvidos em *assembly* IA-32, usando a linguagem C para o código de teste. Não se esqueça de testar devidamente o código desenvolvido, bem como de o apresentar de forma cuidada, apropriadamente indentado e comentado. Não é necessário relatório, mas deverá apresentar um documento com a solução para a questão 4, incluindo a descrição sucinta dos testes de validação das senhas realizados pelo programa em cada fase. Contacte o docente se tiver dúvidas. Encoraja-se a discussão de problemas e soluções com colegas de outros grupos, mas recorda-se que a partilha directa de soluções leva, no mínimo, à anulação das entregas de todos os envolvidos.

1. Implemente em *assembly* IA-32 a função *astrcat*, que será outra sua versão da função *strcat*. Reaproveite os programas de teste desenvolvidos no âmbito da primeira série de exercícios.

```
char * astrcat(char * str1, const char * str2);
```

2. Escreva, em *assembly* IA-32, a função *mulff*, que calcula o produto de dois valores *float*, no formato *binary32* / *single* da norma IEEE 754, utilizando apenas operações inteiras e os registos gerais do processador. O resultado será armazenado no *float* apontado pelo primeiro parâmetro se a função retornar 1 (um). Só é necessário suportar argumentos e resultados com o valor 0.0 ou normalizados com expoente entre -126 (0x01) e +127 (0xFE), inclusive. Caso contrário, a função retornará 0 (zero).

```
int mulff(float *res, float a, float b);
```

Realize um programa de teste em C que lê pares de valores de vírgula flutuante do *standard input* e apresenta no *standard output* o seu produto, calculado com *mulff*, e a diferença desse resultado para o obtido com o produto de *float* próprio da linguagem C.

3. Realize, em *assembly* IA-32, a função *ssearch*, que será uma variante da função *bsearch* da biblioteca C, mas realizando uma pesquisa sequencial sobre um *array* não ordenado.

```
void * ssearch(const void * key, const void * base, size_t size,  
              size_t elem_size, int (* compare)(const void * k, const void * p));
```

Reescreva o programa da alínea 3 da primeira série de exercícios, utilizando a função *ssearch* para realizar as pesquisas das dependências.

4. O programa fornecido em anexo a este enunciado pede 4 senhas de acesso ao utilizador. Pretende-se que descubra quais são as senhas correctas a introduzir, por análise do conteúdo do ficheiro fornecido e da sua execução, tendo sempre em conta que a introdução de uma senha errada tem consequências indesejáveis. Para obter o programa, [escolha apenas o link com o número do seu grupo da sua turma](#).

NOTA: A descoberta das senhas tem dificuldade progressiva. Não desista à primeira contrariedade, mas não deixe de entregar os resultados a tempo só porque lhe falta uma das senhas.

Data limite de entrega: 19 de Abril de 2010

*Bom trabalho!*