

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 colegas de outros grupos, mas recorde-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 que pode ser armazenado no número de *bytes* indicado como parâmetro. A função apenas assume que podem ser armazenados valores negativos se o número de *bytes* indicado for negativo.

```
long int max_value(int bytes);
```

Escreva um programa de teste da função *max\_value* que apresenta o maior valor que pode ser armazenado em cada tipo inteiro, excepto *unsigned long int*. Valoriza-se a portabilidade do programa.

2. Implemente *xstrcat*, que será a sua versão da função *strcat* da biblioteca *standard* da linguagem C, sem recorrer a outras funções dessa biblioteca.

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

Escreva um programa de teste que apresenta no *standard output* as linhas lidas do *standard input* juntas duas a duas. Na saída, cada linha será ser precedida por “*nnn*: ”, em que *nnn* é o número sequencial da linha escrita. Leia as linhas do *standard input* com *gets*, admitindo que cada uma não tem mais do que 255 caracteres.

3. Considere as seguintes definições:

```
typedef struct info_uc {  
    char uc[5];          /* Abreviatura da unidade curricular: POO, ALGA, ... */  
    char tipo;           /* Tipo: B-obrigatória, P-opcional, C-complementar */  
    unsigned char sem;   /* Bitmap dos semestres em que funciona */  
    struct info_uc *depF[3]; /* Dependências fortes: NULL se não existe */  
    struct info_uc *depf[3]; /* Dependências fracas: NULL se não existe */  
} InfoUC;
```

Escreva um programa que leia, via *standard input*, a informação sobre as unidades curriculares de um curso, disponível no [ficheiro de texto anexo](#) a este enunciado. Considere que não existem mais do que 50 unidades curriculares. O programa deverá, apresentar no *standard output* uma listagem das unidades curriculares filtrada com um dos seguintes critérios (e outros à sua escolha), de acordo com uma opção indicada na linha de comando:

- As unidades curriculares de um determinado semestre;
- As unidades curriculares obrigatórias;
- As unidades curriculares que dependem de uma determinada unidade curricular.

Data limite de entrega: 22 de Março de 2010

*Bom trabalho!*