



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 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 `long int min_max_value(int nbytes)` com o seguinte comportamento: a função aceita valores positivos ou negativos para o parâmetro `nbytes`, considerando o seu valor absoluto como a dimensão em *bytes* de um tipo numérico. A função retorna o maior valor (se `nbytes` positivo) ou o menor valor (se `nbytes` negativo) capazes de serem representados na dimensão indicada (valor absoluto de `nbytes`).

Escreva um programa de teste da função `min_max_value` que apresenta o menor e maior valor que pode ser armazenado em cada tipo numérico. Valoriza-se a portabilidade do programa.

2. Implemente a função `char * xstrtok(char * str, const char * delimiters)`, que será a sua implementação da função `strtok` da biblioteca *standard* da linguagem C, sem recorrer a outras funções dessa biblioteca.

Escreva um programa de teste que, para cada linha lida do *standard input*, apresenta no *standard output* os *tokens* da linha que tenham sido especificados nos argumentos do programa. Na saída, cada *token* será precedido por "`nn:`", em que `nn` é o número do *token* na linha de entrada. Leia as linhas do *standard input* com `gets`, admitindo que cada uma não tem mais do que 255 caracteres. O programa termina com uma linha sem *tokens*. No programa de teste poderá recorrer a funções da biblioteca *standard* da linguagem C.

3. Considere as seguintes definições:

```
typedef struct serie {
    char name[4];           /* Nome da série */
    time_t due_date;       /* Prazo de entrega da série */
} Serie;
typedef struct submitted_serie {
    Serie * due_serie;     /* Ponteiro para informação da série */
    time_t submitted_date; /* Data de entrega da série */
} SubmittedSerie;
typedef struct grupo {
    char number;           /* Número do grupo */
    SubmittedSerie sserie[MAX_SERIES]; /* Séries submetidas */
} Grupo;
```

Escreva um programa que leia, via *standard input*, a lista dos grupos de uma turma de entre as turmas de PSC e PICC, disponíveis no ficheiro de texto anexo a este enunciado, considerando que não existem mais do que 20 grupos. As primeiras quatro linhas do ficheiro de texto indicam por ordem crescente as datas de entrega das séries. Os grupos são seleccionados função da turma indicada no parâmetro passado na linha de comando. O programa deverá listar os grupos de acordo com (pelo menos) os seguintes critérios, dependendo dos parâmetros passados igualmente na linha de comando:

- Os grupos que entregaram uma determinada série com uma antecedência mínima de  $N$  dias relativamente à data limite;
- Os grupos que entregaram todas as séries com uma antecedência mínima de  $N$  dias relativamente à data limite;
- Os grupos que não entregaram pelo menos uma série.

#### NOTAS:

1. O tipo `time_t` está definido no ficheiro `/usr/include/time.h`, de forma equivalente a `typedef long int time_t`, representando os valores do tipo o número de segundos de uma data contados a partir de 1/1/1970.
2. O ficheiro de texto anexo só será disponibilizado quando os grupos das turmas de PSC e PICC estiverem definidos.

Data limite de entrega: 27 de Março de 2010

*Bom trabalho!*