

Programação Imperativa

LEI (1º ano) – Ano Lectivo 2007/08

1ª a fase de avaliação prática

2 de Abril de 2008

Apresentação e Normas

Esta é a primeira componente de avaliação contínua da disciplina de Programação Imperativa do 1º ano de LEI. Consiste numa prova prática com a duração de 1h30, na qual os grupos (formados por 3/2 alunos) têm de resolver 3 problemas que testam os princípios básicos da programação em C, nomeadamente a sintaxe, a implementação de funções e a utilização de *arrays* unidimensionais.

O programa em C que resolve cada problema deve ser submetido para avaliação automática, ao sistema Mooshak acessível no URL:

<http://mooshak.di.uminho.pt/~mooshak>

O sistema Mooshak classifica a submissão de **errada** (*compiler error, presentation error* ou *wrong answer*) ou de **correcta** (*accepted*). Um programa é considerado correcto se passar nos 6 testes a que é submetido. Para isso, o programa tem de produzir um resultado exactamente igual ao previsto, com base nas seguintes regras:

1. O programas deve ler do *standard-input* (sem qualquer pergunta inicial) e escrever no *standard-output*.
2. Os *inputs* usados para testar os programas submetidos obedecem às seguintes normas:
 - Não há espaços no fim de cada linha e todas as linhas terminam com mudança de linha (*newline*).
 - Não se usa espaçamento múltiplo, a menos que o enunciado do problema respectivo o especifique explicitamente.
3. Os *outputs* devem obedecer às mesmas normas do ponto anterior.

Sendo esta a primeira utilização do sistema Mooshak pela maior parte dos alunos de PI, os enunciados dos dois primeiros problemas a resolver contêm implementações da função `main` que levam a cabo todas as operações de leitura e escrita de dados necessárias à sua resolução. Nestes casos, a resolução dos problemas consiste apenas na implementação de uma função.

Um problema errado pode ser re-submetido para nova avaliação.

Para evitar qualquer surpresa na compilação automática do programa, aquando da sua submissão ao sistema de avaliação, informa-se que o Mooshak compila os programas C usando o seguinte comando (em que `$file` é o nome do ficheiro contendo o programa submetido):

```
/usr/bin/gcc -Wall -lm -O2 $file
```

Os alunos devem registar-se no sistema Mooshak, no *contest* correspondente a esta primeira prova, designado por PI0708_1, com o número de grupo que corresponde à inscrição feita no sistema de submissão da página de PI (SSeTPs).

Por cada submissão, o sistema Mooshak regista apenas o **login do grupo** (regista portanto o número do grupo e não o nome/número de cada um dos seus elementos) e o **endereço IP** do computador do qual foi submetida a resolução do problema. O ficheiro contendo o código C de resposta a cada questão deve ter um nome formado de acordo com a seguinte regra:

probXn.c

onde X é a letra (A, B ou C) que identifica cada problema e n é o número do grupo.

A classificação final (na escala de 0 a 20) é obtida somando os pontos atribuídos a cada problema correcto. Nesta prova, o problema A vale 7 pontos, o problema B vale 7 pontos, e o problema C vale 6 pontos.

Problema A

Pretende-se desenvolver uma função que determine se um dado número é primo, isto é, se um dado número inteiro n é divisível apenas por 1 e por n ele próprio.

Complete o seguinte programa com uma implementação da função `eprimo` tendo em conta que deve retornar o valor inteiro 1 se o número que receber como parâmetro for primo, e 0 caso contrário.

```
#include <stdio.h>

int main()
{
    int x, r;

    scanf("%d", &x);
    r = eprimo(x);
    printf("%d\n", r);

    return 0;
}
```

Problema B

Pretende-se desenvolver uma função que calcule o mínimo múltiplo comum (mmc) a dois números inteiros, isto é, dado dois números inteiros x e y , encontrar o menor número inteiro m tal que $m = k_1x$ e $m = k_2y$, com k_1 e k_2 inteiros positivos.

Complete o seguinte programa com uma implementação da função `mmc` tendo em conta que deve retornar o valor inteiro do mínimo múltiplo comum entre os dois números inteiros que recebe como parâmetros.

```

#include <stdio.h>

int main() {
    int x, y, r;

    scanf("%d", &x);
    scanf("%d", &y);
    r = mmc(x, y);
    printf("%d\n", r);

    return 0;
}

```

Problema C

Contexto

A cifra linear é um algoritmo que transforma um texto limpo, representado como uma sequência de letras num determinado alfabeto, num criptograma que codifica o mesmo texto limpo, mas de forma aparentemente ilegível.

A transformação introduzida depende de uma chave k : um número inteiro que pode tomar valores de 1 a L , em que L é o número de letras no alfabeto. Neste enunciado vamos adoptar o alfabeto inglês com $L = 26$ letras

ABCDEFGHIJKLMN**OP**QRSTUVWXYZ

adoptando sempre uma representação em LETRAS MAIÚSCULAS.

Dado um texto limpo e uma chave k , a cifra linear funciona da seguinte forma, para cada letra:

- Encontrar a posição x da letra no alfabeto.
- Calcular o valor de $y = k + x$.
- Caso, $y > L$, então $y = y - L$, por forma a y tomar um valor na gama de 1 a L .
- Substituir a letra original pela letra que corresponde à posição y no alfabeto.

A transformação é invertível conhecendo o criptograma e o mesmo valor de k que foi utilizado para cifrar o texto original.

Problema

Dada uma sequência de no máximo 100 letras maiúsculas codificadas com uma cifra linear e com a chave¹ $k = 3$, recuperar o texto limpo original.

¹Diz-se que a cifra linear era utilizada por Júlio César com este valor específico da chave e, por este motivo, é comum denominar-se esta configuração particular de Cifra de César.

Input

O *input* contém apenas uma linha, terminada com *newline*, contendo a sequência de letras maiúsculas a decifrar. Por exemplo:

DEFGHIJKLMNOPQRSTUVWXYZABC

Output

O output deverá ter o mesmo formato do *input*. Por exemplo:

ABCDEFGHIJKLMNOPQRSTUVWXYZ