Paradigmas da Programação II
LESI + LMCC (1^o ano)

2º Trabalho Prático

Ano Lectivo de 01/02

O trabalho deve ser realizado em grupo,
e deve ser entregue acompanhado dum relatório, por submissão electrónica,
até 6ªfeira, 05 de Abril.

1 Objectivos e Organização

Este trabalho prático tem como principais **objectivos**:

- continuação da exploração do computador em ambiente linux, e de algumas ferramentas de apoio à programação
- codificação, na linguagem imperativa C, de estruturas de dados e algoritmos mais complexos concebidos e execução e teste e dos programas escritos;
- documentação dos projectos de programação realizados usando o processador LaTeX.

Para o efeito, esta folha contém seis enunciados, dos quais deverá resolver um.
O **relatório** a elaborar, deve ser claro e sucinto e, além do respectivo enunciado e da descrição do problema, deverá conter exemplos de utilização e o código do programa. Como foi dito, o relatório terá de ser escrito em LaTeX sugerindo-se que adoptem o esqueleto proposto que pode ser descarregado a partir da página WWW principal da disciplina, no URL:

`http://www.di.uminho.pt/~prh/curpp201.html`

Não se esqueça de indicar na capa do relatório, e como título, o nome do problema que escolheu; no campo *autores* deve indicar o nome completo e o número de cada elemento activo do grupo.

2 Enunciados

2.1 Cifra por substituição arbitrária

Considere-se uma cifra (palavra chave) constituída pelas letras "S A P O". Para a cifragem (codificação) da mensagem efectuam-se os seguintes passos:

1. retiram-se espaços e caracteres de pontuação;
2. convertem-se todas as letras em maiúsculas ou em minúsculas;

3. considera-se a seguinte tabela de equivalência:

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T ...
S	A	P	O	B	C	D	E	F	G	H	I	J	K	L	M	N	Q	R	T ...

   A construcção desta tabela limita-se a colocar a palavra chave no início e completar com o alfabeto, não colocando as letras repetidas.

4. para a codificação, substitui-se cada um dos caracteres da primeira linha pelo respectivo caracter da seguinda linha.

Construa um programa que dada uma cifra, codifique a mensagem e um outro que, dada a mensagem e a cifra descodifique a mensagem (claro que neste processo perdemos os espaços e os caracteres de pontuação!).

2.2 Cifra de Vigenère

A cifra de César consiste no processo de adicionar a cada caracter um valor entre 0 e 25, ou seja, $C(x) = (x+n)\ mod\ 26 \quad 0 \leq n \leq 26$. Dado que esta codificação é facilmente quebrada, Blaise Vigenère aplicou uma pequena variante que se manteve inquebrável até meados do século XIX.

O processo de codificação de Vigenère também se baseia numa cifra (palavra chave). Cada uma das letras desta cifra corresponde a um valor inteiro (A=0, B=1, ... Z=25). O processo de codificação consiste nos seguintes passos:

1. retiram-se espaços e caracteres de pontuação;
2. convertem-se todas as letras em maiúsculas ou em minúsculas;
3. divide-se a mensagem em blocos do tamanho da cifra e, para cada um destes blocos, aplica-se a cifra de César a cada um dos caracteres, usando para valor n a correspondente letra;

Considere a palavra chave "A C B" e a mensagem a codificar "sapato". Dividimos "sapato" em grupos de três: "sap" e "ato". Para cada uma das letras de "sap" vamos aplicar a cifra de César ("sap" + "ACB" = "(s+A)(a+C)(p+B)" = "(s+0)(a+2)(p+1)" = "scq"). Aplicando o mesmo algoritmo a "ato" obtemos "avp". Então, a palavra codificada será "scqavp".

Construa um programa que dada uma cifra, codifique a mensagem e um outro que, dada a mensagem e a cifra descodifique a mensagem (claro que neste processo perdemos os espaços e os caracteres de pontuação!).

2.3 Minesweeper

Implemente o conhecido jogo "minesweeper". Dado um tabuleiro $m \times n$ e um conjunto de p bombas (que o programa distribuirá aleatoriamente no tabuleiro), o objectivo do jogador é acertar em todas as casas do tabuleiro sem acertar nas bombas. As jogadas são realizadas da seguinte forma:

1. o jogador indica uma casa do tabuleiro;
2. se acertou numa bomba, perdeu!
3. se existe alguma bomba na vizinhança dessa casa (portanto, nas 8 casas circundantes), o programa deverá mostrar (nessa casa) o número de bombas que existem à sua volta;
4. se não existe nenhuma bomba na vizinhança, o programa deve realizar jogadas automaticamente para cada uma das 8 casas vizinhas, prosseguindo assim até já não haver mais casas nestas condições.

2.4 Jogo da Forca

Implemente o jogo da forca. Nesta implementação, o programa deve pedir inicialmente a frase "mistério" com que vai jogar. Depois, mostra-a na forma de um mapa com a posição das letras em relação aos espaços.

Em cada uma das 6 tentativas (este número pode ser variável), o jogador vai fornecer uma letra que, caso exista na frase "mistério", será substituída no mapa. No caso de não existir, deve ser apresentada, em todas as posteriores jogadas, numa zona de letras usadas em vão.

Como resultado final, o computador deve mostar sempre a frase "mistério" indicando se o Jogador consegui descobrir todas as letras, ou se perdeu por ter esgotado as tentativas.

Opcionalmente, poderá tentar retirar a palavra (ou frase) aleatóriamente de um ficheiro. Uma vez que este trabalho terá um prazo mais apertado do que o que seria necessário para apresentar o funcionamento com ficheiros, está por sua conta para investigar os comandos `fopen`, `fgets`, `fclose` e relacionados.

2.5 Jogo do Galo, ou Todos em Linha

O jogo do galo, com três vezes três casas, torna-se aborrecido porque ao fim de quatro ou cinco jogadas, já existe vencedor ou empate. Sugerimos, então, que se generalize o jogo para $n \times n$ casas.

Neste caso, o computador irá servir de árbitro entre 2 jogadores externos (cada um joga com uma marca diferente), verificando se as jogadas são válidas (se a casa está vazia) e se algum dos jogadores já ganhou.

Um jogador ganha quando consegue preencher todas as n casas de uma linha com a sua marca (linhas horizontais, colunas verticais ou as duas diagonais principais).

2.6 MasterMind de 4 cores

Neste jogo, existem seis cores das quais o computador vai seleccionar quatro, em determinada ordem, o que passa a constituir a chave do jogo.

O objectivo do jogador é adivinhar a chave, isto é, que cores foram seleccionadas e por que ordem.

Para resolver este problema, o jogador tem 10 tentativas. Nestas 10 tentativas, vai palpitar a chave, colocando quatro das seis cores por determinada ordem. Ganha, quando acertar na chave seleccionada pelo computador. Para que o jogo não seja totalmente orientado pela sorte do jogador, em cada jogada o computador indica quantas cores estão correctas, no sítio correcto (número de bolas pretas) e quantas cores estão correctas, mas no sítio errado (número de bolas brancas).