

Algoritmos e Programação 2

Atividade 1

Observações Importantes:

- Envie, para cada exercício, um arquivo contendo seu código fonte. Os nomes dos arquivos deverão seguir o seguinte padrão: <RA>_EX<numero do exercício>.c. Exemplo: 99999_EX1.c
- Todos os arquivos enviados deverão conter um cabeçalho no seguinte formato:

```
/*
Algoritmos e Programacao II
Nome: <nome do aluno>
RA: <ra do aluno>
Exercicio: <numero do exercicio>
Maio/2016
*/
```

- Exercícios com erros de compilação serão desconsiderado.
- Exercícios com suspeita de fraude serão desconsiderados (nota zero para todos os envolvidos).

Exercícios:

1. Faça um programa para controlar as informações dos vereadores de uma cidade. Você deve armazenar o nome do vereador, o CPF, o partido e o status do vereador na câmara (ativo ou inativo). A quantidade de vereadores é desconhecida. Seu programa deverá oferecer as seguintes opções:
 - (a) **Adicionar vereador.** O mesmo vereador não pode ser inserido duas vezes no sistema.
 - (b) **Excluir vereador.** Para excluir um vereador o usuário deverá inserir o CPF do mesmo.
 - (c) **Alterar partido de um vereador.** Para alterar o partido de um vereador o usuário deverá informar o CPF e o novo partido.
 - (d) **Alterar status de um vereador.** Para alterar o status de um vereador o usuário deverá informar o CPF e o novo status.
 - (e) **Listar todos os vereadores.** Os vereadores devem ser listados em ordem alfabética.
 - (f) **Listar os nomes dos vereadores de um determinado partido.** O usuário deverá informar o nome do partido desejado.

2. Faça um programa que usa funções recursivas para imprimir uma árvore de natal. A entrada deve ser um número **n** que representa o tamanho da base da árvore. Para facilitar, você pode considerar que a entrada é sempre um número ímpar.

Exemplos de entrada e saída:

Entrada:	Saída:
9	<pre> * *-* *---* *-----* ***** </pre>
5	<pre> * *-* ***** </pre>

3. O jogo de caça-palavras é um passatempo que consiste de letras arranjadas aparentemente aleatórias em uma grade quadrada ou retangular. O objetivo do jogo é encontrar as palavras escondidas na grade. Na maioria dos jogos, as palavras podem estar escondidas verticalmente, horizontalmente ou diagonalmente dentro da grade.

Faça um programa que ao receber uma grade de letras e uma lista de palavras, encontra a posição das palavras na grade. As palavras podem estar escondidas na grade na vertical ou horizontal.

Entrada:

A primeira linha da entrada consiste de dois números **L** ($1 \leq L \leq 50$) e **C** ($1 \leq C \leq 50$) que indicam o número de linhas e de colunas do caça palavras respectivamente. As **L** linhas seguintes terão cada uma **C** caracteres minúsculos do alfabeto indicando as letras da grade do caça palavras. A linha seguinte contém um número **P** ($1 \leq P \leq 50$) que indica quantas palavras devem ser procuradas. As próximas **P** linhas contém, cada uma, uma única palavra a ser pesquisada. As palavras a serem pesquisadas terão tamanhos condizentes com os limites do caça palavras e conterão apenas letras minúsculas do alfabeto.

Saída:

Imprima para cada palavra, em uma única linha, a posição do primeiro caractere da palavra na grade, na mesma ordem da entrada. Se a palavra não aparecer, imprima -1.

Exemplo de entrada e saída:

Entrada:	Saída:
4 5	2 0
rtoai	1 1
casaa	-1
wrato	
saqef	
3	
casa	
rato	
sapo	

Explicação do exemplo:

```
Entrada:
4 5      // tamanho da grade 4x5
rtoai    // linha 0 da grade com 5 caracteres
casaa    // linha 1 da grade com 5 caracteres
wrato    // linha 2 da grade com 5 caracteres
saqef    // linha 3 da grade com 5 caracteres
3        // quantidade de palavras para pesquisar
casa     // palavra 1
rato     // palavra 2
sapo     // palavra 3

Saída:
2 0
// o primeiro caractere da palavra 'casa' está localizado
// na posição (1, 0) da grade de letras
1 1
// o primeiro caractere da palavra 'rato' está localizado
// na posição (2, 1) da grade de letras
-1
// a palavra 'sapo' não se encontra na grade de letras
```