

Introdução à Programação – Projeto em Grupo

Instruções:

- Projeto para ser feito em grupos de, no máximo, 3 alunos.
- **Data de Entrega: até o dia 16/05/2012 às 23H59.**
- O algoritmo deve ser feito em Python (2.X, programas feitos em versões 3.X não serão aceitos.
- Na data de entrega, enviar o arquivo com a solução juntamente com o arquivo de entrada para o email cecafac@gmail.com, informando os nomes dos componentes do grupo.

Condições para receber nota 0 (zero):

1. Entrega fora do prazo estabelecido.
2. Programa com erros de sintaxe e/ou lógica.
3. Arquivo(s) não abre(m) ou está(ão) corrompido(s).
4. Programa incompleto.
5. Programa com alta similaridade com algum de outros grupos.

Onde está Waldorf?

Dado um grid (matriz) $m \times n$ de letras e uma lista de palavras, construir um programa para encontrar a localização de cada palavra no grid.

Uma palavra deve ser procurada no grid de forma *case-insensitive*, ou seja, letras maiúsculas e minúsculas devem ser tratadas da mesma maneira. Por exemplo, a palavra “AnOTAr” deve ser considerada igual a “anotar”. Uma palavra pode estar contida em uma única linha do grid ou não. Assim, a pesquisa deve ser feita em todas as direções do grid. São 8 direções ao todo, entre horizontais (direita e esquerda), verticais (direita e esquerda) e diagonais.

Entrada

A entrada inicia com uma linha contendo um inteiro positivo indicando o número de casos, seguido por uma linha em branco. Também há uma linha em branco entre dois casos consecutivos. Um caso representa um grid mais uma sequência de palavras a serem procuradas neste grid.

A entrada contém vários casos de testes. Cada caso inicia com um par de inteiros m e n em uma única linha onde $1 \leq m, n \leq 50$. As próximas m linhas contêm n letras cada, representando o grid de letras no qual as palavras devem ser encontradas. As letras podem ser maiúsculas ou minúsculas. Seguindo o grid de letras, vem outro inteiro k em uma única linha ($1 \leq k \leq 20$). As próximas k linhas de entrada contêm as palavras

(uma em cada linha) que devem ser procuradas no grid. Estas palavras também podem conter maiúsculas e/ou minúsculas e não podem ter espaços, hífens ou caracteres especiais. As palavras podem conter acentuação e “ç”.

Saída

Para cada palavra a ser procurada no grid, o programa deve gerar e exibir na tela um par de inteiros representando sua localização. Cada par de inteiros deve vir numa única linha separados por um espaço. O primeiro inteiro do par indica a linha do grid na qual se encontra a primeira letra da palavra (1 representa a primeira linha no topo do grid, e m representa a última linha). O segundo inteiro é a coluna do grid na qual se encontra a primeira letra da palavra (1 representa a coluna mais à esquerda do grid, e n representa a coluna mais à direita). Considerar que uma palavra tem apenas uma ocorrência no grid. As saídas de dois ou mais casos consecutivos devem vir separadas por uma linha em branco.

RECOMENDAÇÕES IMPORTANTES:

- (1) Todos os dados de entrada, descritos acima devem ser lidos de um arquivo de extensão “.txt”. Não esquecer de enviar também este arquivo!
- (2) Usar estrutura de matriz (elaborada através de dicionários em Python) para armazenar o grid.

Exemplo de Entrada

```
1
8 11
abcDEFghigg
hEbkWalDorf
FtyAwaldORm
FtsimrLqsrc
byoArBeDeyv
Klcbqwikomk
strEBGadhrb
yUiqlxcnBjf
4
Waldorf
Bambi
Betty
Dagbert
```

Exemplo de Saída

```
2 5
2 3
1 2
7 8
```