

MC102 - Algoritmos e Programação de Computadores

MC102 Horários Plano de desenvolvimento Plano de aulas

Oferecimentos anteriores

Caça-Palavras

Nesta tarefa trabalharemos com estruturas bidimensionais e dicionários em Python utilizando como tema o passatempo de buscar palavras em um diagrama.

Em 08/outubro, às 11h00, foram alterados os testes 7, 9 e 10 (fechado). Veja a nova versão do arquivo [aux09.zip](#). As notas obtidas em submissões com os testes anteriores continuarão válidas.

Descrição da entrada

As primeiras linhas da entrada conterão um diagrama com letras minúsculas separadas por espaços em branco. A próxima linha indicará o número de palavras a serem procuradas. Em seguida, cada linha conterá uma palavra. Observe o exemplo abaixo:

```
g z h y z l y n s g q j u
i n w a k b n x w w t n y
i w o u i p o f d b o y o
m k o h k l h r k h j w u
j l i o t r t c t b c o p
a h e d n y y y e g t p t
f n o h t y p y t h o n f
r l q o s y y y h n a f x
c e z x t k t m t x e u z
t g m h c v h x c h j n a
s n o w t q o b v n o z j
u n u x x a n p u i g n b
s i x o d j v s f e o n z
1
python
```

Note que as palavras poderão estar dispostas horizontalmente, verticalmente ou em diagonal. Cada palavra poderá ocorrer zero ou mais vezes no diagrama. Uma palavra poderá aparecer mais de uma vez na lista de palavras a serem procuradas.

Descrição da saída

O diagrama de saída conterá as palavras encontradas nas suas respectivas posições e caracteres "." nas demais posições. Em seguida, deverão ser apresentadas, em ordem alfabética, cada uma das palavras solicitadas acompanhada do número de vezes em que foram encontradas no diagrama. Cada palavra deve aparecer apenas uma vez na saída, mesmo que tenha aparecido várias vezes na entrada. Observe a saída esperada para o exemplo acima:

```
. . . . .
. n . . . . n . . . . n .
. . o . . . o . . . o . .
. . . h . . h . . h . . .
. . . . t . t . t . . . .
. . . . . y y y . . . . .
. n o h t y p y t h o n .
. . . . . y y y . . . . .
. . . . t . t . t . . . .
. . . h . . h . . h . . .
. . o . . . o . . . o . .
. n . . . . n . . . . n .
. . . . .
python: 8
```

Testes com o SuSy

O conjunto de testes será formado por 9 testes abertos e 1 teste fechado. A tabela abaixo mostra os diagramas de saída (sem a frequência das palavras) e resume as características principais destes testes, de maneira a facilitar o desenvolvimento e depuração do seu programa.

Teste	Diagrama	Características
arq1.in a l g o r i t m o	<ul style="list-style-type: none"> Apenas uma palavra na horizontal.
arq2.in	. . . i n t r o d u c a o a l g o r i t m o p r o g r a m a	<ul style="list-style-type: none"> Três palavras na horizontal. Palavras com letras próximas às bordas do diagrama. Presença de palavra incompleta no diagrama.
arq3.in i n v e r s o o s r e v n i	<ul style="list-style-type: none"> Apenas uma palavra, com duas ocorrências. Posição horizontal, da esquerda para direita. Posição horizontal, da direita para esquerda.
arq4.in v . p . . e . a . . r . l . . t . a . . i . v . . c	<ul style="list-style-type: none"> Palavras na vertical. Palavras com letras próximas às bordas do diagrama. Presença de palavra incompleta no diagrama. Pedido de busca de palavra inexistente no diagrama.

	<pre> . r . . a . a . . l . s . . . </pre>	
arq5.in	<pre> n o h t y . . a m a r g o r p </pre>	<ul style="list-style-type: none"> • Uma palavra na vertical, de baixo para cima. • Uma palavra na horizontal, da direita para esquerda. • Palavras compartilhando a mesma letra.
arq6.in	<pre> l a n o g a i d </pre>	<ul style="list-style-type: none"> • Palavra na diagonal principal. • Presença de palavras incompletas no diagrama.
arq7.in	<pre> d d . i i . . . a . . a g g o o n . . n . . . a a . l l </pre>	<ul style="list-style-type: none"> • Uma palavra na diagonal secundária, de cima para baixo e da esquerda para direita. • Uma palavra na diagonal principal, de baixo para cima e da direita para esquerda. • Pedido de busca de palavras inexistentes no diagrama. • Pedido da mesma palavra mais de uma vez.
arq8.in	<pre> n n n o . . . o . . . o h . . h . . h t . t . t y y y n o h t y p y t h o n y y y t . t . t h . . h . . h o . . o . . o . . . n n </pre>	<ul style="list-style-type: none"> • Exemplo desta tarefa, palavra python em todas as direções.
arq9.in	<pre> n s d . o . . . a t s i l . . . e . . h r b . . . t . . . i u y . . . n . . . g . . . t u p l a . . . g . n a i l . . m r . . . o . . . a t t . o r . . . s a p g . . . o . . a l g o r i t m o . . l r . f . i n t p </pre>	<ul style="list-style-type: none"> • Teste com várias palavras e casos contemplados nos outros testes.
arq10.in		<ul style="list-style-type: none"> • Teste fechado. • Semelhante ao teste 9 (mesmo diagrama de entrada, conjunto de palavras procuradas diferente).

Releia, se necessário, as instruções para fazer os testes em [Testes com o SuSy](#).

Dicas de Python 3 para esta tarefa:

- Os diagramas de entrada e saída poderão ser armazenados como uma lista de lista de caracteres.

- O código abaixo lê uma lista de caracteres da entrada:

```
l = input().split()
```

- O código abaixo cria um diagrama vazio e em seguida adiciona a lista `l` a este diagrama:

```
diagrama_entrada = []  
diagrama_entrada.append(l)
```

- Você poderá utilizar a função `isdigit()` para verificar se uma linha da entrada é um número ou uma lista de caracteres:

```
l = input()  
if l.isdigit() :  
    n = int(l)  
else :  
    l = l.split()
```

- Para criar um diagrama base para a saída de dimensões `largura` x `altura` você pode utilizar:

```
diagrama_saida = [["." for x in range(largura)] for y in range(altura)]
```

- Recomenda-se a escrita de funções para estruturar o seu programa.
- **O uso de bibliotecas em Python para resolver esta tarefa é permitido desde que o pacote esteja instalado no SuSy.**

Orientações para submissão

Veja [aqui](#) a página de submissão da tarefa. O arquivo a ser submetido deve se chamar `lab09.py`. No link [Arquivos auxiliares](#) há um arquivo [aux09.zip](#) que contém todos os arquivos de testes abertos e seus respectivos resultados compactados.

O limite máximo será de 20 submissões. Serão considerados os resultados da última submissão.

O peso desta tarefa é 3.

O prazo final para submissão é 10/11/2019.

A nota desta tarefa é proporcional ao número de testes que executaram corretamente, desde que o código esteja coerente com o enunciado. **A submissão de um código que não implementa o algoritmo requisitado, mas que exibe as saídas esperadas dos testes abertos a partir da comparação de trechos da entrada será considerada fraude e acarretará a atribuição de nota zero à média final da disciplina.**
