

MC102 - Algoritmos e Programação de Computadores

Caça-Palavras

Nesta tarefa iremos exercitar o uso estruturas bidimensionais em Python utilizando como tema o passatempo de buscar palavras em um diagrama.

Entrada

As primeiras linhas da entrada conterão um diagrama com letras 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:

```
v e d j n a e o
i p y t h o n u
s u e w e t a e
1
python
```

Note que as palavras poderão estar nas direções horizontal, vertical e nas diagonais. Cada palavra ocorrerá no máximo uma vez no diagrama. Nem todas as palavras a serem procuradas estarão nos diagramas.

Saída

O diagrama de saída conterá as palavras encontradas nas suas respectivas posições e caracteres "." nas demais posições. Observe a saída esperada para o exemplo acima:

```
. . . . .
. p y t h o n .
. . . . .
```

Testes para o SuSy

Veja, como exemplo, os testes 2, 6 e 7. Para verificar o conteúdo dos outros testes, favor consultar diretamente os arquivos.

arq2.in	<pre>u l i o a e a p b e a r y l s j i t u a e e h u b e u o r r c t n e p b i o s b 1 python</pre>	<pre>. p y t h o n</pre>	arq2.res
arq6.in	<pre>n a u a n a l n g p y t h o a o u l a m l a v l r d q b a q h g a p a h o b t o a a n g l t a q a d h e w j g c e o p w y l u m r b l t i r s l q l i p r o g z u i u t e b u y f b a a o e t o x t m p u e p l w m e t t o a m p b i e l r a i a l h e e a r d z u r r i b t r u u y r o h h e a i e a u n g h g g g t o p z h s a o o v o i o</pre>	<pre>n o h t y p a m a r g o</pre>	arq6.res

	<pre> r t n a y q k n a o s c r r a e s k n t u n a o a u i p 2 programa python </pre>	<pre> r p </pre>	
arq7.in	<pre> e i r v s n u o t q a a b n y a a s o t a l u v l i t e a h a s a s x p l r r n a m i s o g a n e i a a r f b i s g f i l h a h u u a j e o i w r e b v o h r o a a u d w b e w o o o z a w i n f u y r w r e s i a z q a l u l o a b x t a y s v x c m r p a a e a i a u e i n s w w w a p r s r b o b y s e s b a o i c r t e u i l a c u i u c p e j t p a y a e e p a r a c o a d r n d s l p j e a a f u m r t u e r k e u u e e g y i a t h t a r q x h o n o u a t d e s o d a i e i b o f f f c b r a a e g e i t u y t z b a z a l c m f e i a p e i a a v w f a c u a z o a t a b a m b c o u a s b a e e c u a a i n u m t i a g n i r t s u s a n y d t y q b r e o u r a r o w e l u z c v k m a p s l u e y e u z a s l i u e a p q w t i s y n u c i a f r u i r z f p p m m t e t t t b t u b s o b o o l e a n 10 python string float int boolean if else for while list </pre>	<pre> e t . l n . . i f i . . . h w l i s t r o f f l o a g n i r t s t e l s e b o o l e a n </pre>	arq7.res

Esta tarefa inclui três testes fechados.

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 × 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 outras bibliotecas padrão Python para resolver esta tarefa é permitido, mas não é necessário.

Orientações para submissão

Veja [aqui](#) a página de submissão da tarefa. Lembre-se que o arquivo a ser submetido deve se chamar `main.py`. No link [Arquivos auxiliares](#) há um arquivo [args-08.zip](#) que contém todos os arquivos de testes abertos e seus respectivos resultados compactados. Os arquivos `executa-testes.py` e `executa-testes-windows.py` também estão neste pacote.

Observe o limite máximo de 20 submissões

A nota final é proporcional ao número de testes que executaram corretamente. **A submissão de um código que não implementa o algoritmo solicitado, 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.**

O peso desta tarefa é 2.

O prazo final para submissão é 27/05/2018.
