

MC102 - Algoritmos e Programação de Computadores

MC102 Oferecimento anterior

```

+++++
|    |
|    0
|   / \
|  /   \
+_____ T a _ e _ a   d e   l a _ o _ a t o _ i o 0 8
  
```

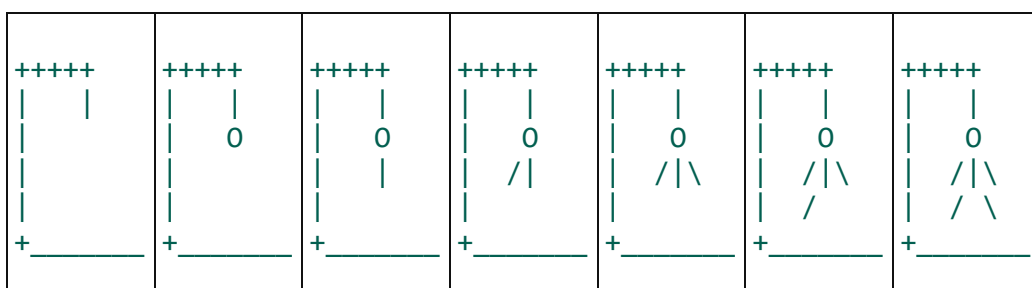
No jogo da forca, o desafio é descobrir uma palavra oculta, tendo como dica o número de letras e podendo palpar uma letra a cada vez. Para aumentar a emoção, a cada letra que não pertence à palavra, um membro de um boneco palito em uma forca é desenhado. O jogo termina se a palavra foi descoberta ou se o número máximo de tentativas foi atingido e o boneco está completo.

Para implementarmos o jogo, utilizaremos como base uma lista de cinquenta palavras que está armazenada no arquivo [forca.py](#).

```

lista_palavras = ["perfeitamente", "liberdade", "enfermidade",
                  "significado", "outono", "chuva", "ilha",
                  "infinito", "solidariedade", "ameixa",
                  "felicidade", "arte", "paternidade",
                  "criatividade", "virtude", "guerra",
                  "democracia", "teatro", "saúde", "adeus",
                  "paz", "honestidade", "horizonte", "sabedoria",
                  "sossego", "maternidade", "esperteza",
                  "primavera", "coragem", "igualdade",
                  "navio", "montanha", "queijo",
                  "gentileza", "tempestade", "joalheria",
                  "paralelogramo", "melancolia", "trem",
                  "inverno", "amizade", "atriz",
                  "computador", "borboleta", "aeroporto",
                  "nascimento", "uva", "oceano", "orquestra",
                  "melancia"]
  
```

Neste mesmo arquivo, há um vetor `cenas_forca` com as cenas para cada situação do jogo em termos de palpites incorretos.



Descrição da entrada e da saída

Nesta tarefa, adotaremos um modelo mais amigável para interação com o usuário e, por isso, a entrada e saída serão descritas na mesma seção.

Inicialmente, o seu programa deverá escrever a mensagem "Escolha um numero entre 0 e 49: ". Caso o número digitado não esteja dentro do intervalo, emitiremos a mensagem "Numero invalido." e o programa será encerrado.

Um número válido servirá para selecionarmos a palavra-desafio na lista. Por exemplo, se o usuário digitar 20, a palavra será **paz** e deveremos apresentar o seguinte cenário:

```
+++++
|      |
|      |
|      |
|      |
+-----
Palavra: _ _ _
Proxima letra:
```

Se o usuário digitar uma letra correta como, por exemplo, **a**, o próximo cenário será:

```
+++++
|      |
|      |
|      |
|      |
+-----
Palavra: _ a _
Proxima letra:
```

Quando o usuário digitar uma letra incorreta como, por exemplo, **m**, o próximo cenário conterá uma linha extra com o registro das letras incorretas. Observe:

```
+++++
|      |
|      0
|      |
|      |
+-----
Palavra: _ a _
Tentativa(s) incorreta(s): m
Proxima letra:
```

Se o usuário repetir uma tentativa, emitiremos a mensagem "Voce jah escolheu esta letra." e reapresentaremos o cenário, sem descontar esta tentativa do número máximo permitido.

O jogo prosseguirá neste esquema. Se o usuário não conseguir completar a palavra, o cenário conterá todas as tentativas incorretas, o boneco completo e a palavra que estava oculta.

```
+++++
|      |
|      0
|     /|\
|     / \
+-----
Palavra: p a _
Tentativa(s) incorreta(s): m s t u i o
Palavra oculta: paz
```

Se o usuário conseguir completar, o cenário final conterá o boneco no estágio referente às tentativas incorretas, a palavra completa, as letras das tentativas incorretas e a mensagem "Palavra encontrada!".

```
+++++
|    |
|    0
|   /|
|
+_____
Palavra: p  a  z
Tentativa(s) incorreta(s): m t l
Palavra encontrada!
```

Dicas de Python 3 para esta tarefa:

- Para utilizar as listas que estão definidas no arquivo `forca.py` escreva no início do arquivo `lab08.py`:

```
from forca import lista_palavras, cenas_forca
```

- Após o `import`, para imprimir o desenho referente a `n_tentativas_incorretas` escreva simplesmente:

```
print(cenas_forca[n_tentativas_incorretas])
```

- A palavra que está sendo construída pode ficar armazenada em uma lista de caracteres. Para iniciar uma lista de tamanho `n` com caracteres `"_"` escreva:

```
palavra = ["_" for i in range(n)]
```

- Para fazer uma leitura emitindo uma mensagem, observe os exemplos abaixo:

```
letra = input("Proxima letra: ")
```

Testes para o SuSy

Esta tarefa de laboratório foi projetado para interação com um usuário humano. No entanto, podemos fazer os testes automáticos como nos laboratórios anteriores. Devemos apenas observar que a saída dos testes abertos conterá as perguntas feitas ao usuário, mas não conterá as respostas nem o efeito do `enter` digitado pelo usuário. Compare as saídas dos arquivos `arq1.in` e `arq4.in` no SuSy e com os efeitos das mesmas entradas na linha de comando.

SuSy		linha de comando
arq1.in	arq1.res	
60	Escolha um numero entre 0 e 49: Numero invalido.	\$ python3 lab08.py Escolha um numero entre 0 e 49: 60 Numero invalido.
SuSy		linha de comando
arq4.in	arq4.res	
20 p a a i i z	Escolha um numero entre 0 e 49: +++++ +_____ Palavra: _ _ _ Proxima letra: +++++	\$ python3 lab08.py Escolha um numero entre 0 e 49: 20 +++++ +_____ Palavra: _ _ _

```
Proxima letra: p

+++++
|      |
|      |
|      |
|      |
+-----
Palavra: p _ _
Proxima letra: a

+++++
|      |
|      |
|      |
|      |
+-----
Palavra: p a _
Proxima letra: a
Voce jah escolheu esta letra.

+++++
|      |
|      |
|      |
|      |
+-----
Palavra: p a _
Proxima letra: i

+++++
|      |
|      0
|      |
|      |
+-----
Palavra: p a _
Tentativa(s) incorreta(s): i
Proxima letra: i
Voce jah escolheu esta letra.

+++++
|      |
|      0
|      |
|      |
+-----
Palavra: p a _
Tentativa(s) incorreta(s): i
Proxima letra: z

+++++
|      |
|      0
|      |
|      |
+-----
Palavra: p a z
Tentativa(s) incorreta(s): i
Palavra encontrada!
```

Esta tarefa contém sete testes abertos e três testes fechados, incluindo casos em que a palavra é encontrada ou não.

Orientações para submissão

Veja [aqui](#) a página de submissão da tarefa. Lembre-se que o arquivo a ser submetido deve se chamar `lab08.py`. Não é necessário submeter o arquivo `forca.py`.

No link [Arquivos auxiliares](#) há um arquivo [aux-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. O arquivo `forca.py` pode ser obtido [aqui](#) ou [aqui](#).

Observe o limite máximo de 40 submissões.

A nota final 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 exhibe 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 3.

O prazo final para submissão é 11/11/2018.
