西安电子科技大学

安全前沿讨论班（I） 课程实验报告

实验名称 实现 hangman 游戏

学 院 网络与信息安全学院 班 级 1618019

姓 名 曹寅峰 学 号 16020610025

实验日期 2019 年 4 月 17 日

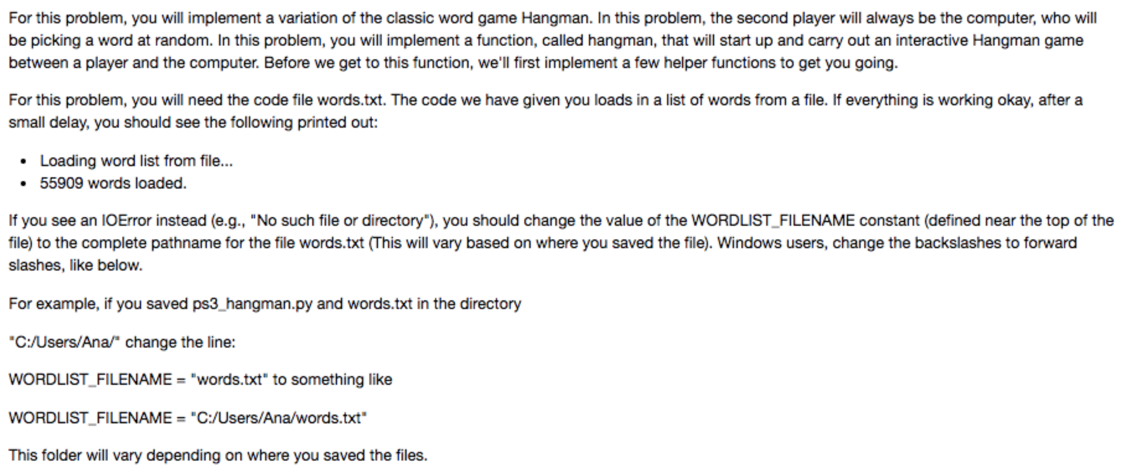
指导教师评语：

指导教师：

______年____月____日

实验报告内容基本要求及参考格式

一、实验目的



For this problem, you will implement a variation of the classic word game Hangman. In this problem, the second player will always be the computer, who will be picking a word at random. In this problem, you will implement a function, called hangman, that will start up and carry out an interactive Hangman game between a player and the computer. Before we get to this function, we'll first implement a few helper functions to get you going.

For this problem, you will need the code file words.txt. The code we have given you loads in a list of words from a file. If everything is working okay, after a small delay, you should see the following printed out:

- Loading word list from file...
- 55909 words loaded.

If you see an IOError instead (e.g., "No such file or directory"), you should change the value of the WORDLIST_FILENAME constant (defined near the top of the file) to the complete pathname for the file words.txt (This will vary based on where you saved the file). Windows users, change the backslashes to forward slashes, like below.

For example, if you saved ps3_hangman.py and words.txt in the directory

"C:/Users/Ana/" change the line:

WORDLIST_FILENAME = "words.txt" to something like

WORDLIST_FILENAME = "C:/Users/Ana/words.txt"

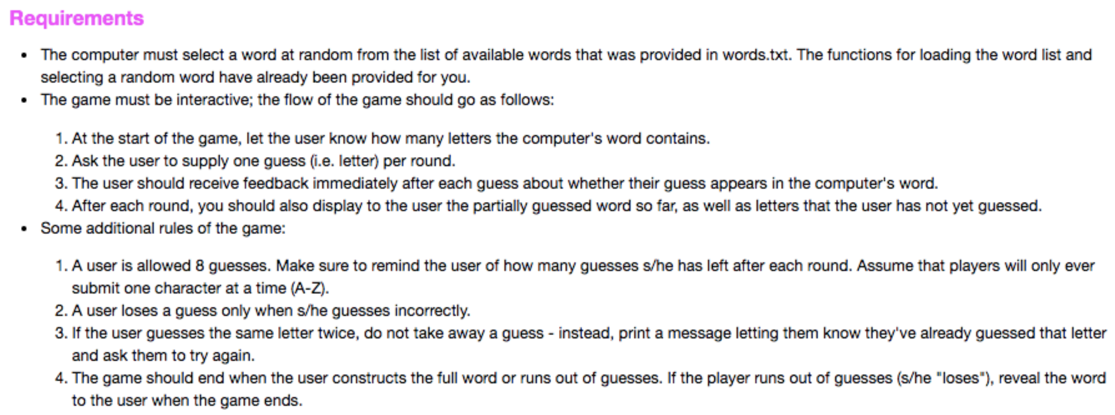
This folder will vary depending on where you saved the files.

实现一个 hangman 交互式猜字游戏，第二个玩家总是计算机，它会随机选择一个单词，根据用户猜测的字母返回相应的结果，用户一共有 8 次机会。

二、实验环境

Python3 jupyternotebook

三、实验基本原理及步骤（或方案设计及理论计算）



Requirements

- The computer must select a word at random from the list of available words that was provided in words.txt. The functions for loading the word list and selecting a random word have already been provided for you.
- The game must be interactive; the flow of the game should go as follows:

  1. At the start of the game, let the user know how many letters the computer's word contains.
  2. Ask the user to supply one guess (i.e. letter) per round.
  3. The user should receive feedback immediately after each guess about whether their guess appears in the computer's word.
  4. After each round, you should also display to the user the partially guessed word so far, as well as letters that the user has not yet guessed.
- Some additional rules of the game:

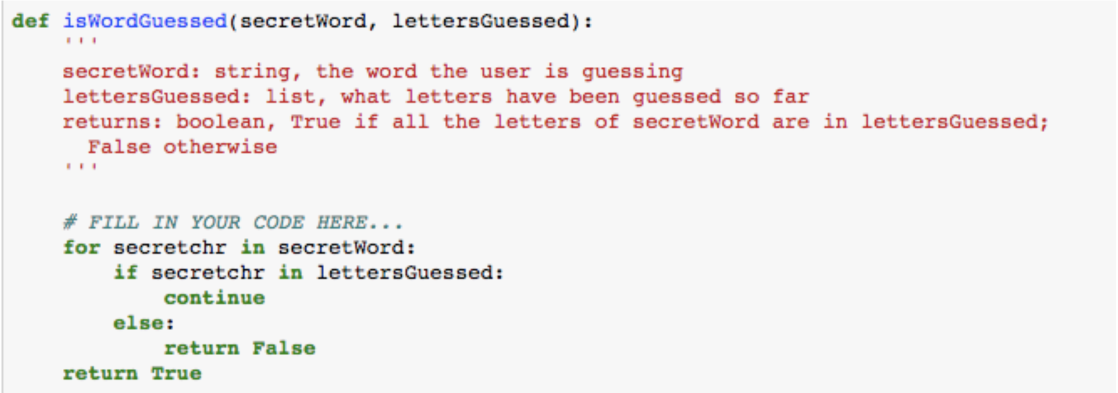
  1. A user is allowed 8 guesses. Make sure to remind the user of how many guesses s/he has left after each round. Assume that players will only ever submit one character at a time (A-Z).
  2. A user loses a guess only when s/he guesses incorrectly.
  3. If the user guesses the same letter twice, do not take away a guess - instead, print a message letting them know they've already guessed that letter and ask them to try again.
  4. The game should end when the user constructs the full word or runs out of guesses. If the player runs out of guesses (s/he "loses"), reveal the word to the user when the game ends.

首先根据范例弄懂游戏的具体细节，每次输入一个字母，如果存在则返回欠缺的部分（重复的时候把字母都填上）

根据需求，我们把游戏判断部分分为三个函数：

1. isWordGuessed(secretWord, lettersGuessed)

这个函数用作胜利的判断条件，如果输入过的猜测字母所组成的列表包含所有的秘密单词中的字母，则胜利。

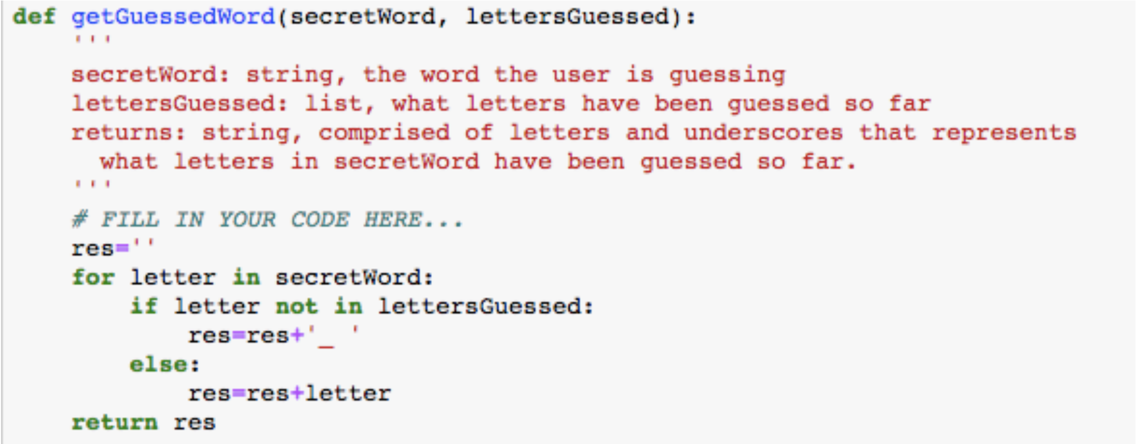


```
def isWordGuessed(secretWord, lettersGuessed):
    '''
    secretWord: string, the word the user is guessing
    lettersGuessed: list, what letters have been guessed so far
    returns: boolean, True if all the letters of secretWord are in lettersGuessed;
      False otherwise
    '''

    # FILL IN YOUR CODE HERE...
    for secretchr in secretWord:
        if secretchr in lettersGuessed:
            continue
        else:
            return False
    return True
```

列举 secretword 中的每个单词，如果在 lettersGuessed 内，则继续列举下一个，直到所有的字母都在 lettersGuessed 内；如果有一个不在，则返回失败。

2. getGuessedWord(secretWord, lettersGuessed):

这个函数用于返回用户的猜测状态，如果没有出现，则用 _ 代替字母

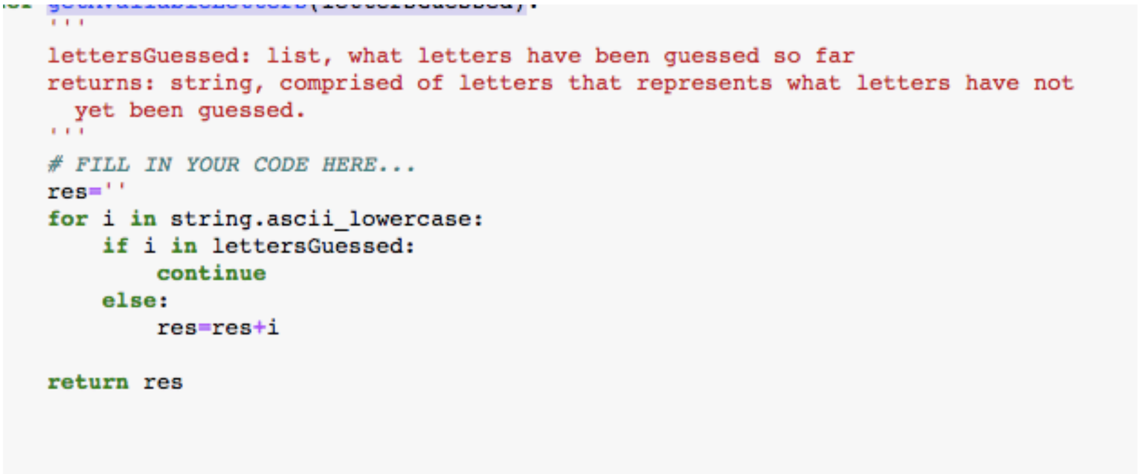


```
def getGuessedWord(secretWord, lettersGuessed):
    '''
    secretWord: string, the word the user is guessing
    lettersGuessed: list, what letters have been guessed so far
    returns: string, comprised of letters and underscores that represents
      what letters in secretWord have been guessed so far.
    '''
    # FILL IN YOUR CODE HERE...
    res=''
    for letter in secretWord:
        if letter not in lettersGuessed:
            res=res+'_ '
        else:
            res=res+letter
    return res
```

思路同样是对比 secretWord 中的字母，如果在 lettersGuessed 中，则在 res 中加上该字母；如果不在，则补上 _

3. getAvailableLetters(lettersGuessed):

这个函数用于更新用户猜测后的字母列表

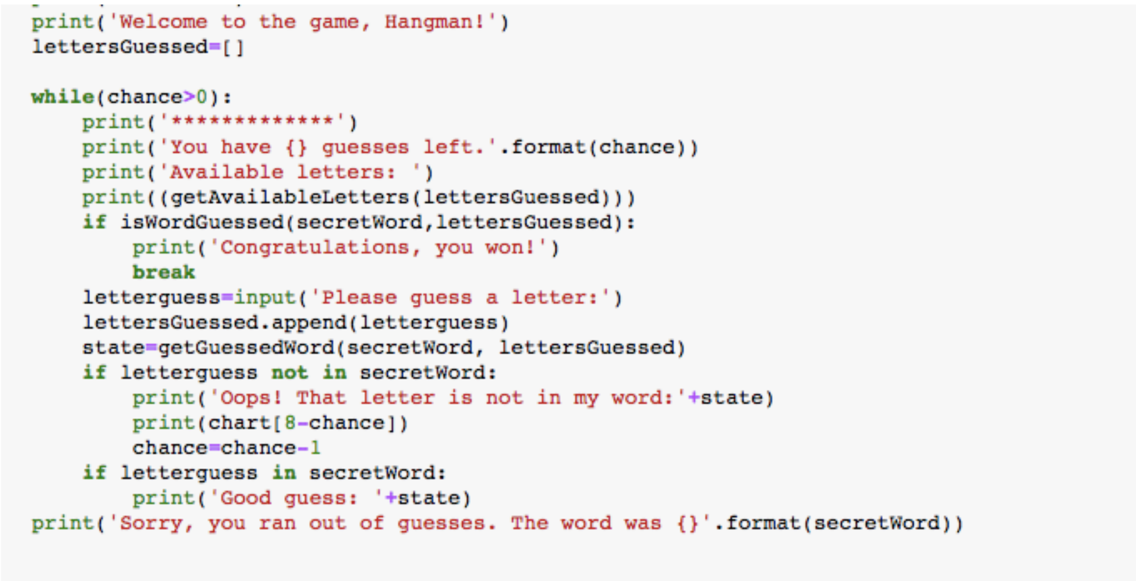


```
def getAvailableLetters(lettersGuessed):
    '''
    lettersGuessed: list, what letters have been guessed so far
    returns: string, comprised of letters that represents what letters have not
      yet been guessed.
    '''
    # FILL IN YOUR CODE HERE...
    res=''
    for i in string.ascii_lowercase:
        if i in lettersGuessed:
            continue
        else:
            res=res+i

    return res
```

思路同样类似，每次对比字母表与当前列表，如果存在说明猜测过，则跳过；不存在说明还没有猜测过，该字母尚可以使用

4.hangman():

最终我们需要将上面 3 个函数结合起来，完成整个游戏。内置的文件读取部分已经写好，完成其余部分即可。



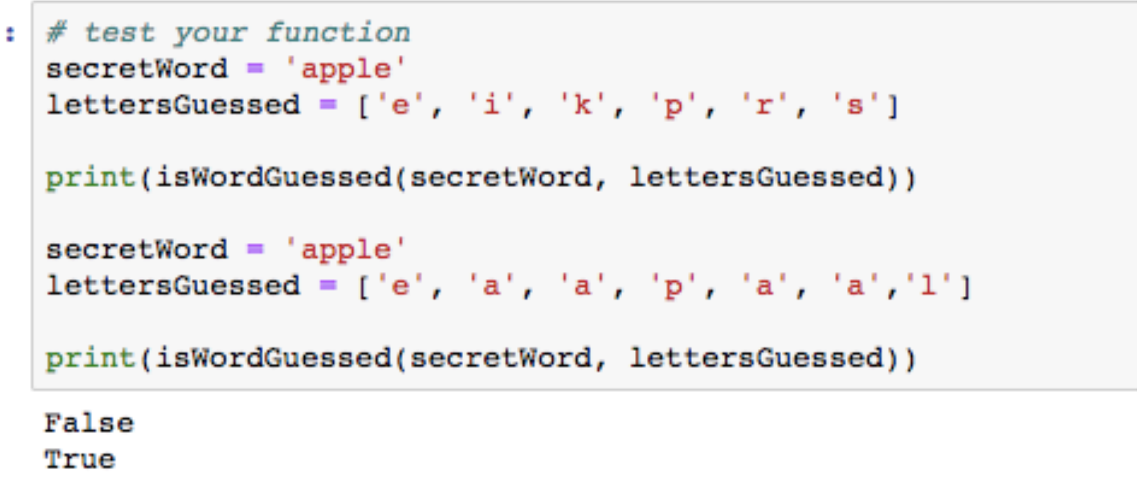
```
print('Welcome to the game, Hangman!')
lettersGuessed=[]

while(chance>0):
    print('*************')
    print('You have {} guesses left.'.format(chance))
    print('Available letters: ')
    print((getAvailableLetters(lettersGuessed)))
    if isWordGuessed(secretWord,lettersGuessed):
        print('Congratulations, you won!')
        break
    letterguess=input('Please guess a letter:')
    lettersGuessed.append(letterguess)
    state=getGuessedWord(secretWord, lettersGuessed)
    if letterguess not in secretWord:
        print('Oops! That letter is not in my word:'+state)
        print(chart[8-chance])
        chance=chance-1
    if letterguess in secretWord:
        print('Good guess: '+state)
print('Sorry, you ran out of guesses. The word was {}'.format(secretWord))
```

首先要调用 getAvailableLetters 打印出当前的单词，其次调用 isWordGuessed 判断是否已经可以获得胜利，如果没有的话则判断这个字母是否正确，通过调用

getGuessedWord 返回尝试结果。将整个过程循环八次（最后加上字符画）

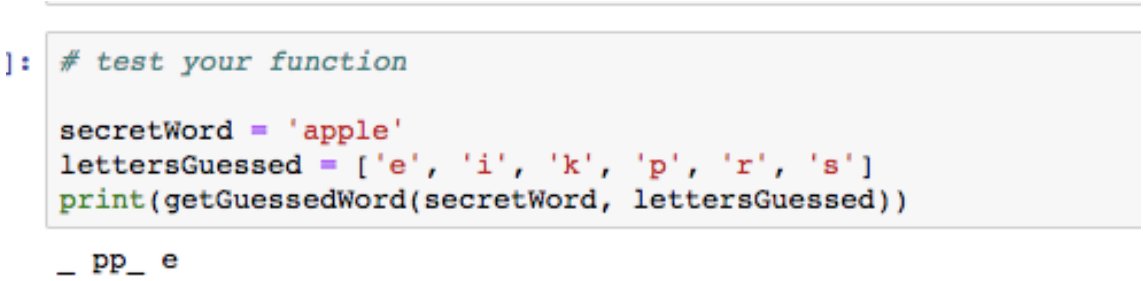
四、实验结果分析及回答问题（或测试环境及测试结果）

1.



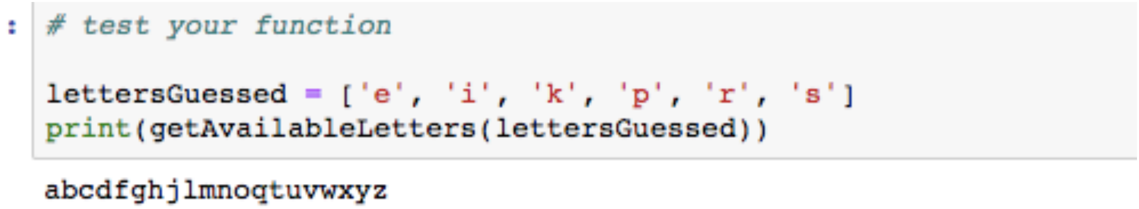
```
:  # test your function
   secretWord = 'apple'
   lettersGuessed = ['e', 'i', 'k', 'p', 'r', 's']

   print(isWordGuessed(secretWord, lettersGuessed))

   secretWord = 'apple'
   lettersGuessed = ['e', 'a', 'a', 'p', 'a', 'a','l']

   print(isWordGuessed(secretWord, lettersGuessed))

   False
   True
```

2.



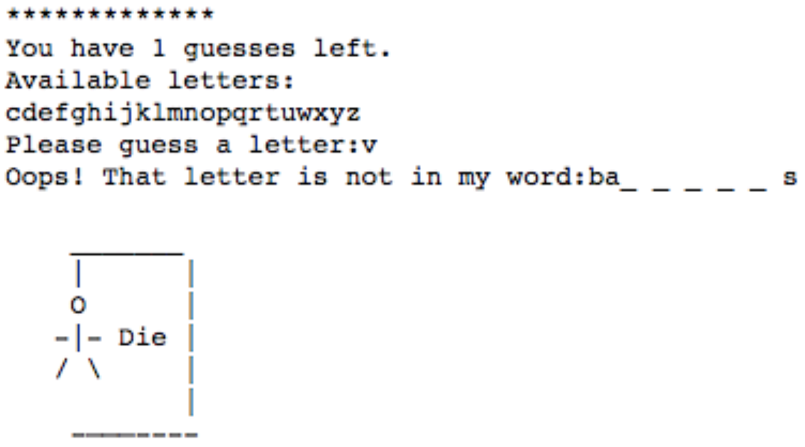
```
]: # test your function

   secretWord = 'apple'
   lettersGuessed = ['e', 'i', 'k', 'p', 'r', 's']
   print(getGuessedWord(secretWord, lettersGuessed))

   _ pp_ e
```

3.

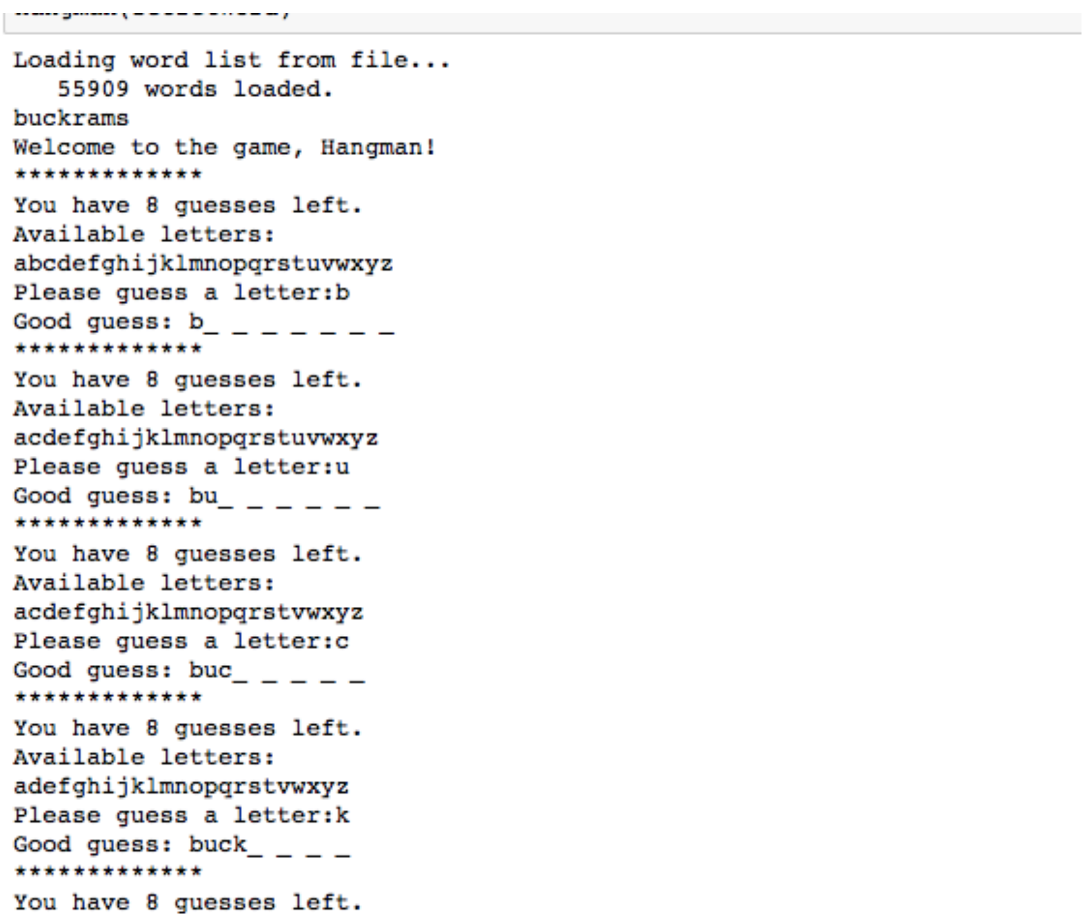


```
:  # test your function

   lettersGuessed = ['e', 'i', 'k', 'p', 'r', 's']
   print(getAvailableLetters(lettersGuessed))

   abcdfghjlmnoqtuvwxyz
```

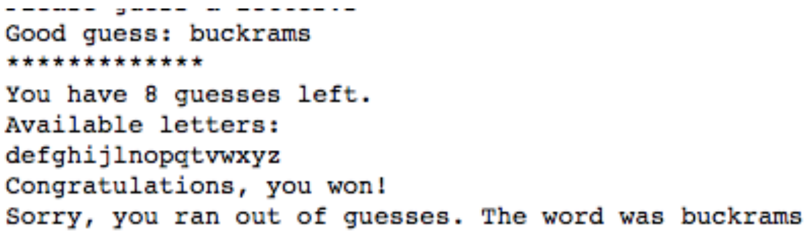
4.



```
*************
You have 1 guesses left.
Available letters:
cdefghijklmnopqrtuwxyz
Please guess a letter:v
Oops! That letter is not in my word:ba_ _ _ _ _ s


    _______
    |      |
    O      |
  -|- Die  |
  / \      |
           |
    --------
```



```
hangman(secret_word)
Loading word list from file...
   55909 words loaded.
buckrams
Welcome to the game, Hangman!
*************
You have 8 guesses left.
Available letters:
abcdefghijklmnopqrstuvwxyz
Please guess a letter:b
Good guess: b_ _ _ _ _ _ _
*************
You have 8 guesses left.
Available letters:
acdefghijklmnopqrstuvwxyz
Please guess a letter:u
Good guess: bu_ _ _ _ _ _
*************
You have 8 guesses left.
Available letters:
acdefghijklmnopqrstvwxyz
Please guess a letter:c
Good guess: buc_ _ _ _ _
*************
You have 8 guesses left.
Available letters:
adefghijklmnopqrstvwxyz
Please guess a letter:k
Good guess: buck_ _ _ _
*************
You have 8 guesses left.
```



```
Please guess a letter:
Good guess: buckrams
*************
You have 8 guesses left.
Available letters:
defghijlnopqtvwxyz
Congratulations, you won!
Sorry, you ran out of guesses. The word was buckrams
```

可以看到，结果均与示例相同