problem set 2 实验报告

16337183 孟衍璋

首先，按照说明文档的需求，从**word.txt**文档中读入单词，并随机选择一个作为**secret_word**，这些步骤在给的代码里已经实现。

我们要实现的功能主要分为两个部分：**无提示（hangman）**与**有提示（hangman_with_hints）**版本。下面分别介绍实现它们的功能需要的辅助函数。

hangman（无提示版本）

要实现这个功能，主要需要以下三个辅助函数：

- `is_word_guessed`
- `get_guessed_word`
- `get_available_letters`

is_word_guessed

这个函数接收两个参数：**secret_word与letters_guessed**，如果玩家猜的字母已经包含了选中单词的所有字母，则返回True；如果玩家还有没猜到的字母，则返回False。

该函数的实现利用了集合的去重性与差集运算，实现代码如下：

```python
def is_word_guessed(secret_word, letters_guessed):
    '''
    secret_word: string, the word the user is guessing; assumes all letters are
      lowercase
    letters_guessed: list (of letters), which letters have been guessed so far;
      assumes that all letters are lowercase
    returns: boolean, True if all the letters of secret_word are in letters_guessed;
      False otherwise
    '''
    # FILL IN YOUR CODE HERE AND DELETE "pass"

    # 将secret_word与letters_guessed都转化为set
    # 若不存在secret_word中有而letters_guessed中没有的字母
    # 那么就猜中该词语
    if len(set(secret_word).difference(set(letters_guessed))) == 0:
        return True
    else:
        return False
```

get_guessed_word

该函数返回玩家猜测单词的情况，显示猜中的字母，未猜中的字母用`_`表示，为了保证可读性，每个字符之间有一个空格的距离。实现代码如下：

```
def get_guessed_word(secret_word, letters_guessed):
    '''
    secret_word: string, the word the user is guessing
    letters_guessed: list (of letters), which letters have been guessed so far
    returns: string, comprised of letters, underscores (_), and spaces that represents
      which letters in secret_word have been guessed so far.
    '''
    # FILL IN YOUR CODE HERE AND DELETE "pass"

    # 遍历secret_word中所有字母
    # 如果已经在letters_guessed中，则打印出来
    # 如果不在，则打印'_'
    word = ''
    for i in secret_word:
        if i in letters_guessed:
            word += i
            word += ' '
        else:
            word += '_ '
    return word
```

get_available_letters

该函数按照字母表顺序返回还未猜过的字母，这里利用了集合的差集运算，再转换为字符串后对字母进行排序。实现代码如下：

```
def get_available_letters(letters_guessed):
    '''
    letters_guessed: list (of letters), which letters have been guessed so far
    returns: string (of letters), comprised of letters that represents which letters have not
      yet been guessed.
    '''
    # FILL IN YOUR CODE HERE AND DELETE "pass"

    # 先利用集合的差集运算计算出还可以猜的字母
    # 再将集合转化为字符串
    # 再对字符串元素进行排序
    all_letters = string.ascii_lowercase
    left_letters = set(all_letters).difference(set(letters_guessed))
    left_letters = ''.join(left_letters)
    left_letters = ''.join((lambda x:(x.sort(),x)[1])(list(left_letters)))
    return left_letters
```

有了前面三个辅助函数，便可以开始编写`hangman`函数，这个函数主要实现用户与计算机之间的交互，完成`hangman`游戏的步骤。

在实现过程中，有一些需要注意的地方。首先是允许玩家错误的机会共有6次，每猜错一次就扣一次；且存在`warning`机制，即玩家输入不合法的字符或者输入之前猜过的字母时，计算机会警告玩家，共有三次警告的机会，若3次过后，再受到警告，则扣除一次猜单词的机会。

实现代码如下：

```
def hangman(secret_word):
    '''
    secret_word: string, the secret word to guess.

    Starts up an interactive game of Hangman.
    * At the start of the game, let the user know how many
      letters the secret_word contains and how many guesses s/he starts
with.

    * The user should start with 6 guesses

    * Before each round, you should display to the user how many guesses
      s/he has left and the letters that the user has not yet guessed.

    * Ask the user to supply one guess per round. Remember to make
      sure that the user puts in a letter!

    * The user should receive feedback immediately after each guess
      about whether their guess appears in the computer's word.

    * After each guess, you should display to the user the
      partially guessed word so far.

    Follows the other limitations detailed in the problem write-up.
    '''
    # FILL IN YOUR CODE HERE AND DELETE "pass"

    letters_guessed = []
    guesses_left = 6
    warnings_left = 3

    print('Welcome to the game Hangman!')
    print('I am thinking of a word that is', len(secret_word), 'letters
long.')
    print('You have', warnings_left, 'warnings left.')
    print('-------------')

    while(is_word_guessed(secret_word, letters_guessed) == 0 and
guesses_left > 0):
        print('You have', guesses_left, 'guesses left.')
        print('Available letters:',
get_available_letters(letters_guessed))
        letter = input('Please guess a letter:')
        if letter.isalpha() == 0:  # 如果输入不合法
            warnings_left -= 1
            if warnings_left >= 0:  # 如果warnings的次数未用光
```

```
                print('Oops! That is not a valid letter. You have', warnings_left, 'warnings left:', get_guessed_word(secret_word, letters_guessed))
                print('-------------')
            else:  # 如果warnings的次数用光了
                print('Oops! That is not a valid letter. You have no warnings left so you lose one guess:', get_guessed_word(secret_word, letters_guessed))
                guesses_left -= 1
                print('-------------')
                warnings_left = 3
            continue

        letter = letter.lower()

        if letter in letters_guessed:  # 输入了之前猜过的字母
            warnings_left -= 1
            if warnings_left >= 0:  # 如果warnings的次数未用光
                print('Oops! You\'ve already guessed that letter. You have', warnings_left, 'warnings left:', get_guessed_word(secret_word, letters_guessed))
                print('-------------')
            else:  # 如果warnings的次数用光了
                print('Oops! You\'ve already guessed that letter. You have no warnings left so you lose one guess:', get_guessed_word(secret_word, letters_guessed))
                guesses_left -= 1
                print('-------------')
                warnings_left = 3

        elif secret_word.find(letter) != -1:  # 在secret_word中找到相应字母
            letters_guessed.append(letter)
            print('Good guess: ', get_guessed_word(secret_word, letters_guessed))
            print('-------------')
        elif secret_word.find(letter) == -1:  # 在secret_word中没有找到相应字母
            letters_guessed.append(letter)
            guesses_left -= 1
            print('Oops! That letter is not in my word:', get_guessed_word(secret_word, letters_guessed))
            print('-------------')
```

```
    if is_word_guessed(secret_word, letters_guessed):
        print('Congratulations, you won!')
        print('Your total score for this game is:', guesses_left *
len(set(secret_word)))
    else:
        print('Sorry, you ran out of guesses. The word was else. It\'s',
secret_word)
```

hangman_with_hints

接下来就需要实现有提示的版本，当玩家输入的字符为**星号（*）**时，打印出所有与当前猜测匹配的候选词。为了实现这个功能，需要有两个辅助函数。

match_with_gaps

该函数对比当前输入与候选词，若匹配则返回True，不匹配则返回False。首先判断两个字符串长度是否相等，若不相等自然返回False；之后判断相同位置的字母是否相同，不相同则返回False；最后需要排除例如`a_ple`与`apple`对比的情况，应返回False。不属于以上情况的，则返回True。

代码实现如下：

```
def match_with_gaps(my_word, other_word):
    '''
    my_word: string with _ characters, current guess of secret word
    other_word: string, regular English word
    returns: boolean, True if all the actual letters of my_word match the
        corresponding letters of other_word, or the letter is the special
symbol
        _ , and my_word and other_word are of the same length;
        False otherwise:
    '''
    # FILL IN YOUR CODE HERE AND DELETE "pass"

    my_word = my_word.replace(' ', '')
    # 若两个字符串长度不相等，返回False
    if len(my_word) != len(other_word):
        return False
    # 若相同位置字母不相同，返回False
    for i in range(len(my_word)):
        if my_word[i] != '_' and my_word[i] != other_word[i]:
            return False
    # 排除例如'a_ple'与'apple'比较的情况，应返回False
    for i in range(len(my_word)):
        if my_word[i] == '_' and other_word[i] in my_word:
            return False
    return True
```

show_possible_matches

该函数返回列表中与玩家当前输入匹配的所有候选词，代码实现如下：

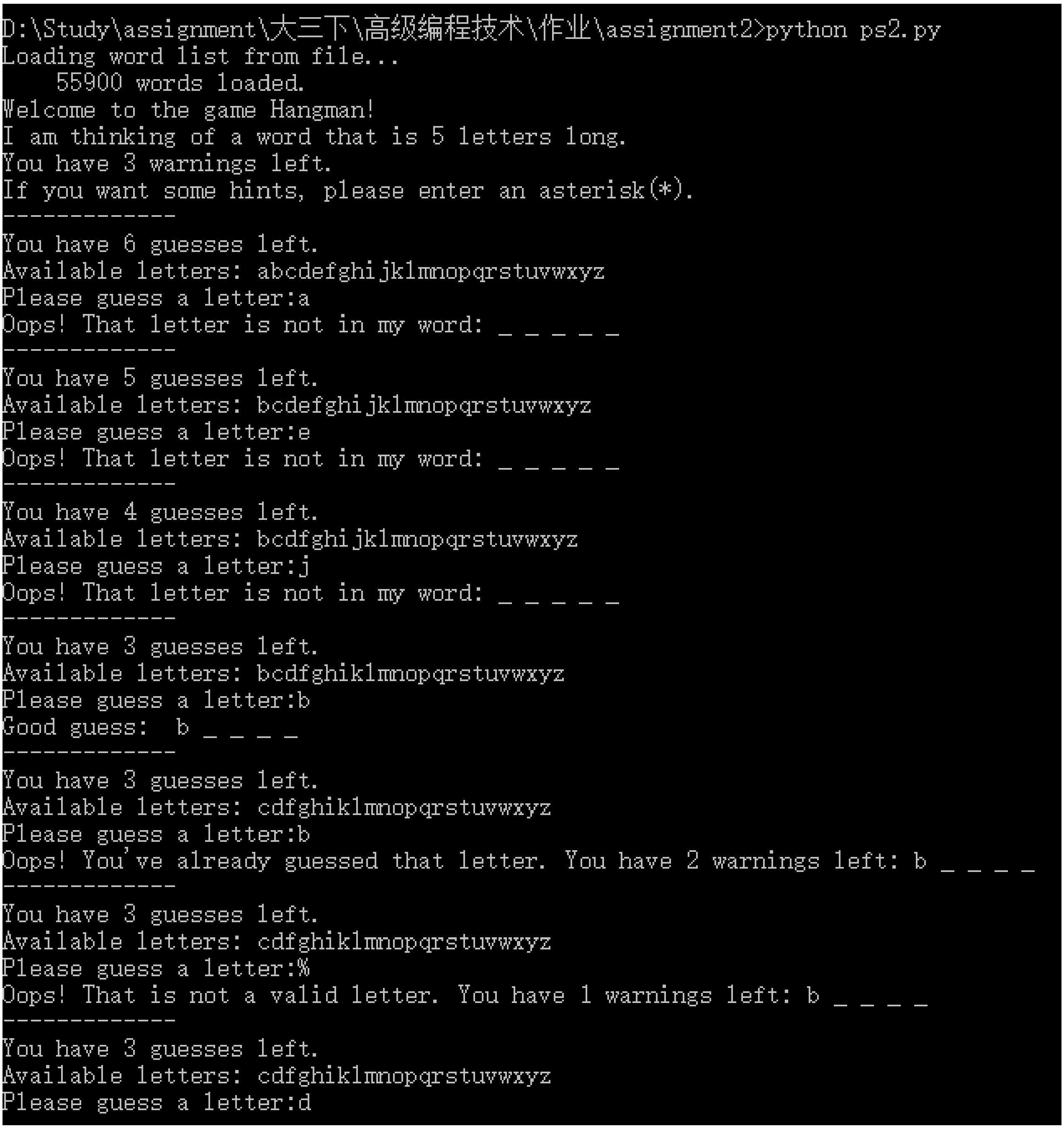
```
def show_possible_matches(my_word):
    '''
    my_word: string with _ characters, current guess of secret word
    returns: nothing, but should print out every word in wordlist that
matches my_word
             Keep in mind that in hangman when a letter is guessed, all
the positions
             at which that letter occurs in the secret word are revealed.
             Therefore, the hidden letter(_ ) cannot be one of the letters
in the word
             that has already been revealed.

    '''
    # FILL IN YOUR CODE HERE AND DELETE "pass"
    flag = 0
    for word in wordlist:
        if match_with_gaps(my_word, word):
            print(word, end = ' ')
            flag = 1
    if flag == 0:
        print('No matches found', end = '')
    print('\n', end = '')
```

现在已经可以开始实现有提示的功能了，主体代码与`hangman`函数相同，主要是增加了输入提示的情况。增加的部分如下：

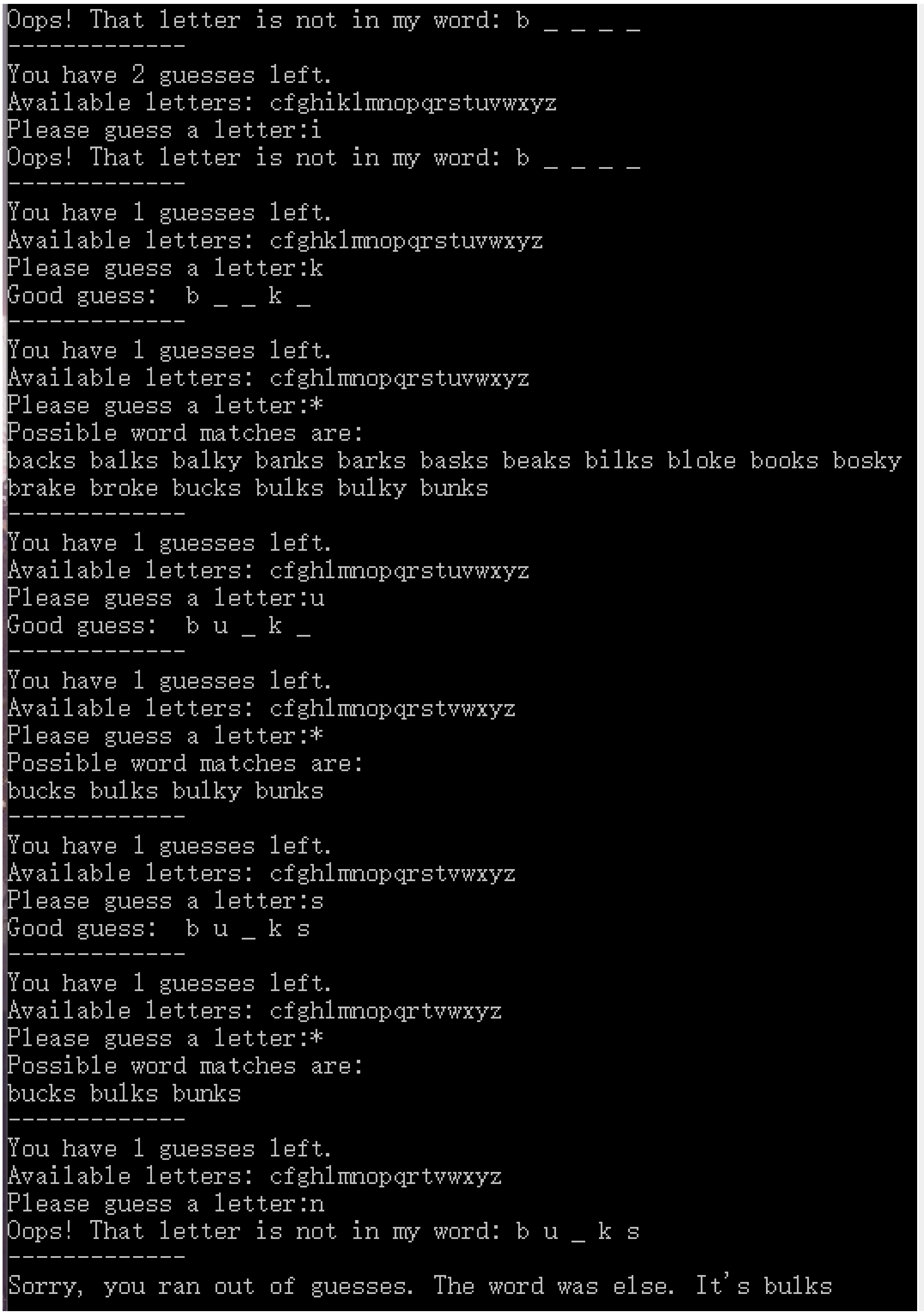
```
if letter == '*':
    print('Possible word matches are:')
    show_possible_matches(get_guessed_word(secret_word, letters_guessed))
    print('-------------')
    continue
```

实验结果

猜测失败

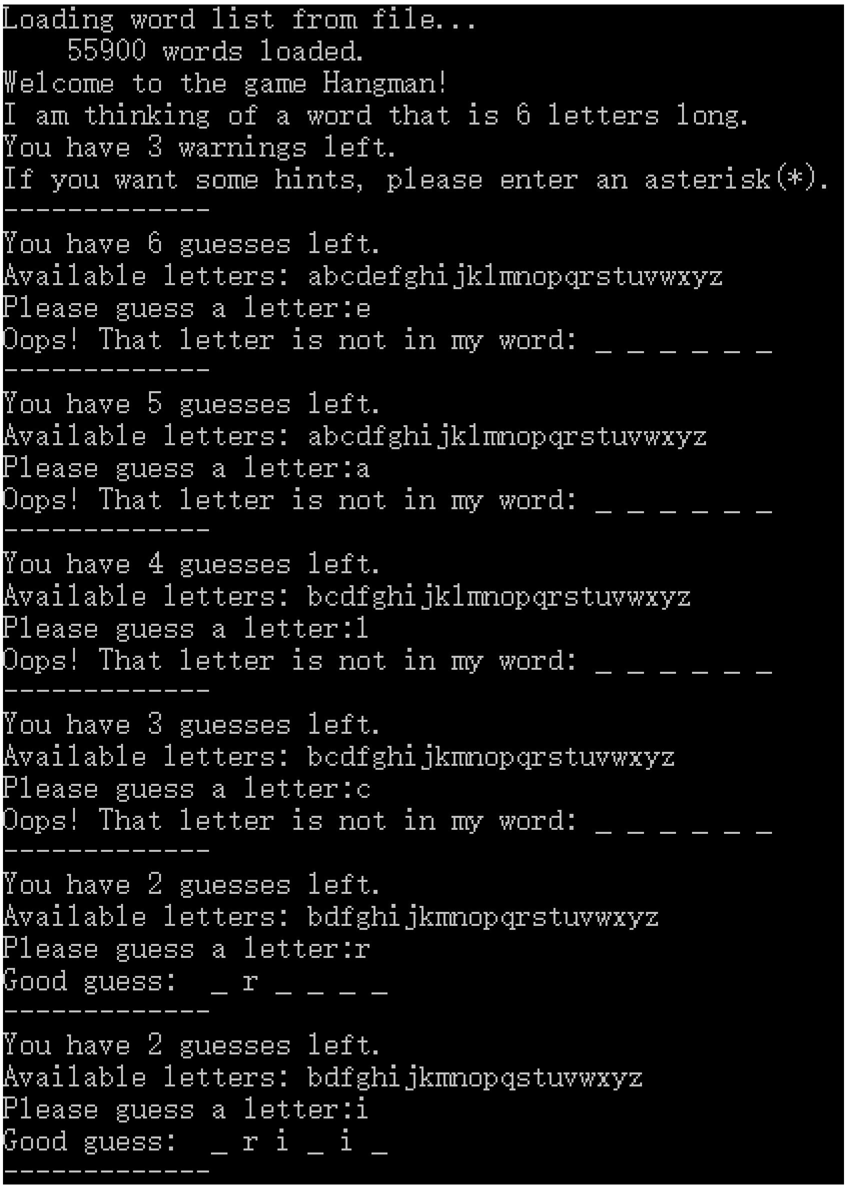


```
D:\Study\assignment\大三下\高级编程技术\作业\assignment2>python ps2.py
Loading word list from file...
    55900 words loaded.
Welcome to the game Hangman!
I am thinking of a word that is 5 letters long.
You have 3 warnings left.
If you want some hints, please enter an asterisk(*).
-------------
You have 6 guesses left.
Available letters: abcdefghijklmnopqrstuvwxyz
Please guess a letter:a
Oops! That letter is not in my word: _ _ _ _ _
-------------
You have 5 guesses left.
Available letters: bcdefghijklmnopqrstuvwxyz
Please guess a letter:e
Oops! That letter is not in my word: _ _ _ _ _
-------------
You have 4 guesses left.
Available letters: bcdfghijklmnopqrstuvwxyz
Please guess a letter:j
Oops! That letter is not in my word: _ _ _ _ _
-------------
You have 3 guesses left.
Available letters: bcdfghiklmnopqrstuvwxyz
Please guess a letter:b
Good guess:  b _ _ _ _
-------------
You have 3 guesses left.
Available letters: cdfghiklmnopqrstuvwxyz
Please guess a letter:b
Oops! You've already guessed that letter. You have 2 warnings left: b _ _ _ _
-------------
You have 3 guesses left.
Available letters: cdfghiklmnopqrstuvwxyz
Please guess a letter:%
Oops! That is not a valid letter. You have 1 warnings left: b _ _ _ _
-------------
You have 3 guesses left.
Available letters: cdfghiklmnopqrstuvwxyz
Please guess a letter:d
```

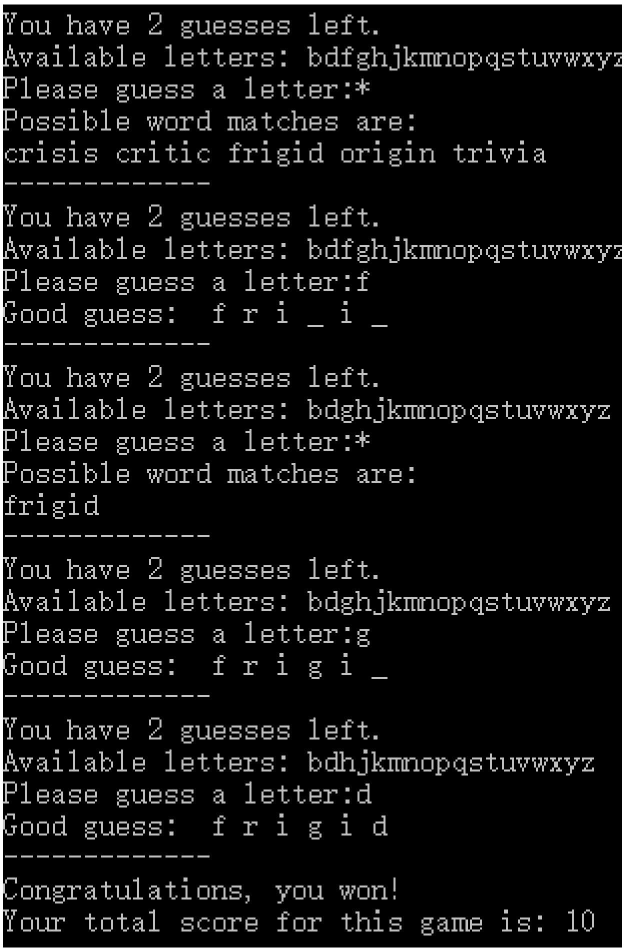


```
Oops! That letter is not in my word: b _ _ _ _
-------------
You have 2 guesses left.
Available letters: cfghiklmnopqrstuvwxyz
Please guess a letter:i
Oops! That letter is not in my word: b _ _ _ _
-------------
You have 1 guesses left.
Available letters: cfghklmnopqrstuvwxyz
Please guess a letter:k
Good guess:  b _ _ k _
-------------
You have 1 guesses left.
Available letters: cfghlmnopqrstuvwxyz
Please guess a letter:*
Possible word matches are:
backs balks balky banks barks basks beaks bilks bloke books bosky
brake broke bucks bulks bulky bunks
-------------
You have 1 guesses left.
Available letters: cfghlmnopqrstuvwxyz
Please guess a letter:u
Good guess:  b u _ k _
-------------
You have 1 guesses left.
Available letters: cfghlmnopqrstvwxyz
Please guess a letter:*
Possible word matches are:
bucks bulks bulky bunks
-------------
You have 1 guesses left.
Available letters: cfghlmnopqrstvwxyz
Please guess a letter:s
Good guess:  b u _ k s
-------------
You have 1 guesses left.
Available letters: cfghlmnopqrtvwxyz
Please guess a letter:*
Possible word matches are:
bucks bulks bunks
-------------
You have 1 guesses left.
Available letters: cfghlmnopqrtvwxyz
Please guess a letter:n
Oops! That letter is not in my word: b u _ k s
-------------
Sorry, you ran out of guesses. The word was else. It's bulks
```

猜测成功



```
Loading word list from file...
    55900 words loaded.
Welcome to the game Hangman!
I am thinking of a word that is 6 letters long.
You have 3 warnings left.
If you want some hints, please enter an asterisk(*).
-------------
You have 6 guesses left.
Available letters: abcdefghijklmnopqrstuvwxyz
Please guess a letter:e
Oops! That letter is not in my word: _ _ _ _ _ _
-------------
You have 5 guesses left.
Available letters: abcdfghijklmnopqrstuvwxyz
Please guess a letter:a
Oops! That letter is not in my word: _ _ _ _ _ _
-------------
You have 4 guesses left.
Available letters: bcdfghijklmnopqrstuvwxyz
Please guess a letter:l
Oops! That letter is not in my word: _ _ _ _ _ _
-------------
You have 3 guesses left.
Available letters: bcdfghijkmnopqrstuvwxyz
Please guess a letter:c
Oops! That letter is not in my word: _ _ _ _ _ _
-------------
You have 2 guesses left.
Available letters: bdfghijkmnopqrstuvwxyz
Please guess a letter:r
Good guess:  _ r _ _ _ _
-------------
You have 2 guesses left.
Available letters: bdfghijkmnopqstuvwxyz
Please guess a letter:i
Good guess:  _ r i _ i _
-------------
```



```
You have 2 guesses left.
Available letters: bdfghjkmnopqstuvwxyz
Please guess a letter:*
Possible word matches are:
crisis critic frigid origin trivia
-------------
You have 2 guesses left.
Available letters: bdfghjkmnopqstuvwxyz
Please guess a letter:f
Good guess:  f r i _ i _
-------------
You have 2 guesses left.
Available letters: bdghjkmnopqstuvwxyz
Please guess a letter:*
Possible word matches are:
frigid
-------------
You have 2 guesses left.
Available letters: bdghjkmnopqstuvwxyz
Please guess a letter:g
Good guess:  f r i g i _
-------------
You have 2 guesses left.
Available letters: bdhjkmnopqstuvwxyz
Please guess a letter:d
Good guess:  f r i g i d
-------------
Congratulations, you won!
Your total score for this game is: 10
```