1.9Hangman

(a)

The fifteen most frequent 5-letter words:(higher prob to lower prob)

THREE SEVEN EIGHT WOULD ABOUT
THEIR WHICH AFTER FIRST FIFTY
OTHER FORTY YEARS THERE SIXTY

The fourteen least frequent 5-letter word:(higher prob to lower prob)

CCAIR CLEFT FABRI FOAMY NIAID
PAXON SERNA TOCOR YALOM BOSAK
CAIXA MAPCO OTTIS TROUP

The result makes sense because people use numeral, preposition a lot, the probability of which will be higher while the probability of rare words is quite low.

(b)

Correctly guessed	Incorrectly guessed	Best next guess l	P(Li=l for some i∈{1,2,3,4,5}|E)
—————	**{}**	**E**	**0.5394**
—————	**{E,A}**	**O**	**0.5340**
A———S	**{}**	**E**	**0.7715**
A———S	**{I}**	**E**	**0.7127**
——O——	**{A,E,M,N,T}**	**R**	**0.7454**

Source code for Part(a):

```
from operator import itemgetter
#part(a)
data=[]
with open('hw1_word_counts_05.txt',newline='')as inputfile:
    for line in inputfile:
        data.append(line.strip().split(' '))
total_count=0
for sub in range(len(data)):
    total_count=total_count+int(data[sub][1])
for sub in range(len(data)):
    data[sub].append(int(data[sub][1])/total_count)
data_b_to_s=sorted(data,key=itemgetter(2),reverse=True)
print('The fifteen most frequent 5-letter words:(higher prob to lower prob)')
for i in range(0,15):
    print(data_b_to_s[i][0])
print('The fourteen least frequent 5-letter word:(higher prob to lower prob)')
for i in range(-14,-1):
    print(data_b_to_s[i][0])
print(data_b_to_s[-1][0])
print(data_b_to_s[:15])
```

part(b)(1)

```
from operator import itemgetter
def update_prob(letter):
    data1=[]
    for i in range(len(data)):
        if letter in data[i][0]:
            data1.append(data[i])
    total_count1=0
    for sub1 in range(len(data1)):
        total_count1=total_count1+int(data1[sub1][1])
    prob.append([letter,total_count1/total_count])
#import original data
data=[]
with open('hw1_word_counts_05.txt',newline='')as inputfile:
    for line in inputfile:
        data.append(line.strip().split(' '))
total_count=0
for sub in range(len(data)):
    total_count=total_count+int(data[sub][1])

prob=[]
```

```
letters=['A','B','C','D','E','F','G','H','I','G','K','L','M','N','O','P','Q','R','S','T','U','V','W','X','Y','Z']
for i in letters:
    update_prob(i)
sorted_prob=sorted(prob,key=itemgetter(1),reverse=True)
print(sorted_prob[0])
```

part(b)(2)

```
from operator import itemgetter
def update_prob(letter):
    data1=[]
    for i in range(len(data)):
        if letter in data[i][0]:
            data1.append(data[i])
    total_count1=0
    for sub1 in range(len(data1)):
        total_count1=total_count1+int(data1[sub1][1])
    prob.append([letter,total_count1/total_count])
#
original=[]
with open('hw1_word_counts_05.txt')as inputfile:
    for line in inputfile:
        original.append(line.strip().split(' '))
data=[]
for i in range(len(original)):
     if ('A' not in original[i][0]) and ('E' not in original[i][0]):
        data.append(original[i])
total_count=0
for sub in range(len(data)):
    total_count=total_count+int(data[sub][1])

prob=[]
letters=['A','B','C','D','E','F','G','H','I','G','K','L','M','N','O','P','Q','R','S','T','U','V','W','X','Y','Z']
for i in letters:
    update_prob(i)
sorted_prob=sorted(prob,key=itemgetter(1),reverse=True)
print(sorted_prob[0])
```

part(b)(3)

```
from operator import itemgetter
def update_prob(letter):
    data1=[]
    for i in range(len(data)):
        if letter in data[i][0]:
            data1.append(data[i])
```

```
   total_count1=0
   for sub1 in range(len(data1)):
      total_count1=total_count1+int(data1[sub1][1])
   prob.append([letter,total_count1/total_count])

original=[]
with open('hw1_word_counts_05.txt')as inputfile:
   for line in inputfile:
      original.append(line.strip().split(' '))

data=[]
for i in range(len(original)):
    if ( original[i][0].startswith('A')) and (original[i][0].endswith('S'))and(original[i][0][1] != "A")an
d(original[i][0][1]!="S")and(original[i][0][2] != "A")and(original[i][0][2]!="S")and(original[i][0][3]
 != "A")and(original[i][0][3]!="S"):
      data.append(original[i])
total_count=0
for sub in range(len(data)):
   total_count=total_count+int(data[sub][1])

prob=[]
letters=['B','C','D','E','F','G','H','I','G','K','L','M','N','O','P','Q','R','T','U','V','W','X','Y','Z']
for i in letters:
   update_prob(i)
sorted_prob=sorted(prob,key=itemgetter(1),reverse=True)
print(sorted_prob[0])
```

part(b)(4)

```
from operator import itemgetter
def update_prob(letter):
   data1=[]
   for i in range(len(data)):
      if letter in data[i][0]:
         data1.append(data[i])
   total_count1=0
   for sub1 in range(len(data1)):
      total_count1=total_count1+int(data1[sub1][1])
   prob.append([letter,total_count1/total_count])

original=[]
with open('hw1_word_counts_05.txt')as inputfile:
   for line in inputfile:
      original.append(line.strip().split(' '))
```

```
data=[]
for i in range(len(original)):
    if ( original[i][0].startswith('A')) and (original[i][0].endswith('S'))and(original[i][0][1] != "A")and(original[i][0][1]!="S")and(original[i][0][2] != "A")and(original[i][0][2]!="S")and(original[i][0][3] != "A")and(original[i][0][3]!="S")and(original[i][0][1] != "I")and(original[i][0][2] != "I")and(original[i][0][3] != "I"):
      data.append(original[i])
total_count=0
for sub in range(len(data)):
   total_count=total_count+int(data[sub][1])

prob=[]
letters=['B','C','D','E','F','G','H','I','G','K','L','M','N','O','P','Q','R','T','U','V','W','X','Y','Z']
for i in letters:
   update_prob(i)
sorted_prob=sorted(prob,key=itemgetter(1),reverse=True)
print(sorted_prob[0])
```

part(b)(5)

```
from operator import itemgetter
def update_prob(letter):
   data1=[]
   for i in range(len(data)):
      if letter in data[i][0]:
         data1.append(data[i])
   total_count1=0
   for sub1 in range(len(data1)):
      total_count1=total_count1+int(data1[sub1][1])
   prob.append([letter,total_count1/total_count])

original=[]
with open('hw1_word_counts_05.txt')as inputfile:
   for line in inputfile:
      original.append(line.strip().split(' '))

data=[]
for i in range(len(original)):
    if (original[i][0][2]=="O")and(original[i][0][0]!="A")and(original[i][0][0]!="E")and(original[i][0][0]!="M")and(original[i][0][0]!="N")and(original[i][0][0]!="T")and(original[i][0][0]!="O")and(original[i][0][1]!="A")and(original[i][0][1]!="E")and(original[i][0][1]!="M")and(original[i][0][1]!="N")and(original[i][0][1]!="T")and(original[i][0][1]!="O")and(original[i][0][3]!="A")and(original[i][0][3]!="E")and(original[i][0][3]!="M")and(original[i][0][3]!="N")and(original[i][0][3]!="T")and(original[i][0][3]!="O")and(original[i][0][4]!="A")and(original[i][0][4]!="E")and(original[i][0][4]!="M")and(original[i][0][4]!="N")and(original[i][0][4]!="T")and(original[i][0][4]!="O"):
        data.append(original[i])
total_count=0
for sub in range(len(data)):
    total_count=total_count+int(data[sub][1])

prob=[]
letters=['A','B','C','D','E','F','G','H','I','G','K','L','M','N','P','Q','R','S','T','U','V','W','X','Y','Z']
for i in letters:
    update_prob(i)
sorted_prob=sorted(prob,key=itemgetter(1),reverse=True)
print(sorted_prob[0])
```