1 Homework 1

1.1 Problem 1

```python
inp = "When half way through the journey of our life I found that I was "
inp = inp + "in a gloomy wood, because the path which led aright was lost. "
inp = inp + "And ah, how hard it is to say just what this wild and "
inp = inp + "rough and stubborn woodland was, the very thought of which renews_ "
inp = inp + "my fear!"
print(inp)
```

When half way through the journey of our life I found that I was in a gloomy wood, because the path which led aright was lost. And ah, how hard it is to say just what this wild and rough and stubborn woodland was, the very thought of which renews my fear!

```python
# We need to count words...words can be separated by
# blank spaces, but also various punctuations.
# Save the original entry, then substitutes blank
# spaces for commas etc.
original = inp
inp = inp.replace("","" , " ")
inp = inp.replace("." , " ")
inp = inp.replace(";" , " " )
inp = inp.replace(":" , " " )
inp = inp.replace("!" , " " )
inp = inp.replace("?" , " " )

# This is most likely not necessary, but just in case
# remove leading and trailing whitespaces
inp = inp.strip()
```

```python
# Split the strings into pieces, using whitespaces
# The output is a list. Each item is a list is a
# word. The number of items in the list is then the
# number of words
w = inp.split()
```
The number of words in the sentence is 52

# To count the number of non blank characters we will take
# the original input string and remove all whitespaces.
# The length of the string will then be the number of characters.
inpNoBlank = original.replace(" ","")
print("The number of non blank characters in the sentence is ", len(inpNoBlank))

The number of non blank characters in the sentence is 204

1.2 Problem 2

```python
inp = "When half way through the journey of our life I found that I was"
inp = inp + "in a gloomy wood, because the path which led aright was lost. "
inp = inp + "And ah, how hard it is to say just what this wild and "
inp = inp + "rough and stubborn woodland was, the very thought of which renews my fear!"
print(inp)
```

When half way through the journey of our life I found that I was in a gloomy wood, because the path which led aright was lost. And ah, how hard it is to say just what this wild and rough and stubborn woodland was, the very thought of which renews my fear!

```python
# Replace punctuations with blank spaces.
inp = inp.replace(""," ","")
inp = inp.replace(":"," ","")
inp = inp.replace("!"," ","")
inp = inp.replace("?"," ","")
```

# This is most likely not necessary, but just in case
# remove leading and trailing whitespaces
inp = inp.strip()

# Split the strings into pieces, using whitespaces
# The output is a list. Each item in the list is a word
words = inp.split()

# This is a brute force way of doing it.
# There are surely better ways
# a new empty list to hold all the words with no duplicates
wordsNoDup = []  # an empty list

# Loop over words from the original list.
# If the word is not in the new list, add it to it
for w in words:
    if w not in wordsNoDup:
        wordsNoDup.append(w)

# Build the output string and send it to the screen
output = " "  # an empty string
for w in wordsNoDup:
    output = output + w + " ">
print(output.strip())

When half way through the journey of our life I found that was in a gloomy wood because path which led aright was lost And ah how hard it is to say just what this wild and rough stubborn woodland very thought renews my fear

1.3 Problem 3

[8]: # Although the first and last are already specified
    # in the question, it is often a good idea to define
    # them at the beginning instead of harwiring them
    # throughout the code. This way if you are asked to
    # change them, you know exactly where to go to make the change.
    first = 100
    last = 400

[9]: # Loop over numbers from first to last
    for i in range(first, last+1):  # last+1 is skipped!!!
        s = str(i)  # turn number into string
        good = True  # assume the number i is "good"
        for k in list(s):
            if int(k)%2 != 0:  # k is a string, int(k) is an integer
                good = False  # this number is no good
                break  # found odd character, no point in looking further
        if good:
            print(i)
1.4 Problem 4

[10]: # Note: I am using "//" for division instead of "/"  
# The reason is that I want to make absolutely sure  
# that when I divide x/y, where x and y are integers,  
# and the remainder of the division (calculated using  
# the % operator) is zero, then x/y is an integer and  
# not a float. This is most likely not necessary, but  
# "better safe than sorry"

# We need this package in order to take square roots.  
import math

# The number to factorize  
thisNumber = 9572121  
# list to keep the factors  
factors = []

[11]: # treat 2 in a "special" way  
if thisNumber % 2 == 0: factors.append(2)  
    N = thisNumber  
    while N % 2 == 0:  
        N = N // 2

[12]: keepLooping = True  
while keepLooping and N>1 :  
    # Loop over all possible factors
# Note to point checking any number > sqrt-of-N
# (the "+3" is to account for non-integer sqrts,
# and may be excessive...ie, +1 or +2 may be enough)
# Note
# The business with the isUniqueFactor is probably also not
# needed. In any case, it does not hurt.
foundFactor = False
for i in range(3, int(math.sqrt(N))+3, 2):
    if (N % i) == 0:
        foundFactor = True
        isUniqueFactor = True
        for j in factors:
            if (i % j) == 0: isUniqueFactor = False
        if isUniqueFactor: factors.append(i)
        while N % i == 0:
            N = N // i
        break

# We found the last factor.
# No more trying to break it down
# Add it to the list unless it is a repeat
if not foundFactor:
    keepLooping = False
    isAlreadyThere = N in factors
    if (not isAlreadyThere): factors.append(int(N))

[13]: # if we found no factors we are done
if len(factors)==0:
    print(thisNumber," is prime")
else:
    powers=[0]*len(factors)
    index = -1
    for i in factors:
        index = index + 1
        p = 1
        while thisNumber % (i**p) == 0: # keep dividing
            p = p+1
            powers[index] = p - 1

    # and now output
    for index in range(0,len(factors)):
        print(factors[index], " to the power ", powers[index])

3 to the power  3
13 to the power  1
27271 to the power  1