Introduction to Computer Programming for the Physical Sciences

Joseph F. Hennawi

Spring 2024

☐ Open a new Jupyter notebook

☐ Name your notebook with your name and Homework 1

☐ Open a Markdown cell at the top and write your name and Homework 1

☐ Open a Markdown cell before each problem and write e.g. Problem 1, Problem 2(a), etc.

☐ Please abide by the **Policy and Guidelines on Using AI Tools**

☐ Once you finish the problems: 1) Restart the Python kernel and clear all cell outputs. 2) Rerun the notebook from start to finish so that all answers/outputs show up. 3) Save your notebook as a single .pdf file and upload it to Gradescope on Canvas by the deadline. **No late homeworks will be accepted except for illness accompanied by a doctor's note.**

☐ For parts of problems that require analytical solutions you can perform your calculations using a pencil and paper. Then photograph your work and convert the photograph to a .pdf file using an online tool. Homework assignments can only be submitted as a single .pdf file, so you will also need to figure out how to concatenate your photo .pdf file and your notebook .pdf file into a single .pdf file that you can submit. Online websites can do this for you. Alternatively, you can code up the analytical solution to your problems in a notebook Markdown cell using the LaTeX mathematical rendering language. This is harder but a chatbot can help you learn it.

Homework 1

Problem 1

Write a program that takes a sentence and prints out a count of the number of words and the number of characters in the sentence. Note that words can be separated by white spaces but also punctuation (commas, semicolon, etc.), so your program should deal with that. For example `hello world` and `hello, world` and `hello,world` all consists of two words. Soon we will learn how to make such a program into a function. For now, test your code on this sentence:

"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!"

In [ ]:

```
# Your solution here please
sentence = "When half way through the journey of our life I found that I was in
#the input sentence
word_count=0
#start with 0 words
i=0
while(i<len(sentence)):
    #going through all the elements of sentence.
    if not ((sentence[i]>='a' and sentence[i]<='z') or (sentence[i]>='A' and ser
        #finding out if it is not a part of word.
        for j in range(i,len(sentence)):
            #looking form the first none work chracter and try to find net chrac
            if((sentence[j]>='a' and sentence[j]<='z') or (sentence[j]>='A' and
                i=j
                word_count=word_count+1
                break
    i=i+1
word_count=word_count+1
print(word_count)
```

```
52
```

Problem 2

Write a program that prints to the screen the set of integers between 100 and 400 (inclusive) for which all of the *digits* are even.

In [ ]:

```
# Your solution here please
for i in range(50,201,1):
    #100/2=50,400/2=200,since need even integers from 100 to 400, equivient to 2
    #so we can use range 50 to 201 since 201 is not included.
    print(2*i)
```

```
100
102
104
106
108
110
112
114
116
118
120
122
124
126
128
130
132
134
136
138
140
142
144
146
148
150
152
154
156
158
160
162
164
166
168
170
172
174
176
178
180
182
184
186
188
190
192
194
196
198
200
202
204
206
208
210
212
214
216
218
```

```
220
222
224
226
228
230
232
234
236
238
240
242
244
246
248
250
252
254
256
258
260
262
264
266
268
270
272
274
276
278
280
282
284
286
288
290
292
294
296
298
300
302
304
306
308
310
312
314
316
318
320
322
324
326
328
330
332
334
336
338
```

```
340
342
344
346
348
350
352
354
356
358
360
362
364
366
368
370
372
374
376
378
380
382
384
386
388
390
392
394
396
398
400
```

Problem 3

Write a program that calculates the prime number factors of any number, with their relative powers. For example if the number was 4567956 it should return

2 to the power of 2
3 to the power of 1
191 to the power 1
1993 to the power 1

Check your program by running it on test cases for which you know the answer (e.g. prime numbers). Calculate the prime factors for the number 9572121. If you want to further test your code you can run it on sample numbers using a website like https://www.calculatorsoup.com/calculators/math/prime-factors.php.

In [ ]:

```
# Your solution here please
factors = {}
def prime_factors_with_powers(n):
    #the function compute all the factors with powers
    i = 2
    #setting up innitial values
    while i * i <= n:
        #since we are looking for facotrs, the largest possible factor execpt it
        #the following method make sure that all the facotrs are prime, since th
        while n % i == 0:
            #we find a factor and see the power of it
            n = n // i
            #update the n, reducing the time of loops we have to take
            if i in factors:
                factors[i] += 1
            else:
                factors[i] = 1
            #we are using i as the possible prime factor, and factors[i] is the
        i = i + 1
    if n > 1:
        factors[n] = 1
    #This is for if the n it self is a prime.
    return

n=4567956
prime_factors_with_powers(n)
for i in range(n+1):
    if i in factors:
        print(i,"to the power of",factors[i])
```

```
2 to the power of 2
3 to the power of 1
191 to the power of 1
1993 to the power of 1
```