day34 prime number finder
Due: Tuesday 11/2, 6th period
Wednesday 11/3, 7th period

Your task is to write a program that finds and prints all the prime numbers between whatever two numbers the user enters. For example, if the user enters 3 and 11, your program will print the following:

```
We're going to find prime numbers.
Enter the first number: 3
Enter the last number: 11

3     5     7     11

4 prime numbers found between 3 and 11
```

Make sure your program actually checks the last number, as shown above. See that it checked and found that 11 was prime?

Requirements:

- Write a function called "prime" which does the checking. The function should return "True" if a number is prime, and "False" if the number is not prime.
- Print the prime numbers in 5 columns.
- Report how many prime numbers were found at the end.
- Use try/except structures to make sure the user enters two valid numbers for the start and the end values.
- Also add a check to make sure that the end value is larger than the start value.

Tips:

1. We have recently done programs where we figured out if a number is prime. You can find and use that code from before.

2. Because the function returns True or False, you can and should do an if statement where you check each number, like this:

```
if prime(n):
```

That's the same as saying "`if prime(n) == True:`".

See next page for sample runs

```
Sample run:

We're going to find prime numbers.
Enter a starting value: 22
Enter an ending value: 2
The end value has to be higher than the start value.

Enter a starting value: 1
Enter an ending value: 300

2     3     5     7     11
13    17    19    23    29
31    37    41    43    47
53    59    61    67    71
73    79    83    89    97
101   103   107   109   113
127   131   137   139   149
151   157   163   167   173
179   181   191   193   197
199   211   223   227   229
233   239   241   251   257
263   269   271   277   281
283   293

62 primes found between 1 and 300


Another sample:

We're going to find prime numbers.
Enter a starting value: 1000
Enter an ending value: 1500

1009  1013  1019  1021  1031
1033  1039  1049  1051  1061
1063  1069  1087  1091  1093
1097  1103  1109  1117  1123
1129  1151  1153  1163  1171
1181  1187  1193  1201  1213
1217  1223  1229  1231  1237
1249  1259  1277  1279  1283
1289  1291  1297  1301  1303
1307  1319  1321  1327  1361
1367  1373  1381  1399  1409
1423  1427  1429  1433  1439
1447  1451  1453  1459  1471
1481  1483  1487  1489  1493
1499

71 primes found between 1000 and 1500
The process took 0.564054012298584 seconds.
```

In my last example I added code to keep track of how long the whole run took. You can do that if you want. Add "import time" at the top, then before the main for loop say startTime = time.time(). Then after you print the summary message, say "endTime = time.time(). Then to see how many seconds the whole thing took, subtract **startTime** from **endtime**:

```
print(endTime-startTime,"seconds")
```

This is optional, I just wanted to share it.