Tuple and List Basics

1. Given a list of 5 integers, determine: the first and last entries in the list; the smallest and largest values; and the total of all of the values.
2. Have the user enter a character and a positive integer. Create and display a list containing that many instances of the character.
3. Initialize a list with ten zeroes. Obtain ten strings from the user, and store them in the list. Display these values in two ways: one by printing the raw list itself, and the other using a loop to display the values on a single line of output.
4. *Fibonacci Numbers*: The Fibonacci sequence of numbers begins 1, 1, 2, 3, 5, 8, etc. Each term after the two initial 1s is generated by adding together the two terms that precede it. For example, 1+1=2, 1+2=3, 2+3=5, and so on. Initialize a 20-element list, then use a loop to generate the first 20 terms in the Fibonacci sequence.
5. *Micro Database*: Create a three-element list called *StudentInfo*, which will contain three pieces of information: first name, last name, and student number. Write a loop that allows the user to do one of four things: update the first name, update the last name, update the student number, or quit the program. When the user quits, display the entered information in a nice format.
6. *Sieve of Eratosthenes*: Here is an ancient method for generating prime numbers less than *n*. Start with a list of numbers from 0 to *n*-1. Since neither 0 nor 1 are prime, cross them off. 2 is the smallest prime number. Cross off all multiples of 2 in the list that are greater than 2. Find the next value in the list that has not been crossed off. In this case, this is 3. This value is prime. Cross off all multiples of 3 that are greater than 3. Find the next value in the list that has not been crossed off. In this case, this is 5. Cross of multiples of 5 greater than 5, etc. Repeat until it is not possible to find a next uncrossed value. Use the `list` and `range` functions to create a list containing the values from 0 to *n*-1, then perform the steps above to mark all non-prime numbers from the list. Print the remaining prime numbers.