```
# Charles Thomas Wallace Truscott
# Sorting and Searching for the Fundamental Theorem of Arithmetic, not yet in lowest prime
multiples
# n log n complexity
def find_factors(n):
        Factors = []
        for x in range(1, n):
                if n % x == 0:
                        Factors.append(x)
        if Factors == [1]:
#               fh.write("{} is prime \n\n".format(n))
                return False
        else:
                return Factors

def Fundamental_Theorem(n):
        Factors = find_factors(n)
        Multiples = {}
        if Factors == False:
#               fh.write("{} is a prime number. \n\n".format(n))
                print("{} is a prime number. \n\n".format(n))
                Multiples[n] = 1
        else:
                for Factor in Factors:
                        h = n
                        l = 0
                        g = (h + l) / 2.0
#                       print("Factor: {}".format(Factor))
                        while (int(abs(round(g * Factor, 0))) != abs(round(n, 0))):
#                               print("g: {} h: {} l: {}".format(g, h, l))
                                if abs(round(g * Factor, 0)) > n:
                                        h = g
                                elif abs(round(g * Factor, 0)) < n:
                                        l = g
                                g = (h + l) / 2.0
                        Multiples[Factor] = int(abs(round(g, 0)))
#       print("The factorisations of {} are the following {}".format(n, Multiples))
        return Multiples
def Charles():
#       text = open('Factors_ctruscott.txt', 'w+')
#       text.write("Charles Thomas Wallace Truscott Watters. Love you Dad Mark William
Watters. Culinary delights and Byron Bay. Missing our pup Bert.\n\n\n")
        for number in range(1, 100000, 1):
#               text.write("# Factors of {} \n\n".format(number))
                print("# Factors of {} \n\n".format(number))
                Factors = Fundamental_Theorem(number)
                for factor in Factors:
#                       text.write("{} is {} x {}\n\n".format(number, factor,
Factors[factor]))
                        print("{} is {} x {}\n\n".format(number, factor, Factors[factor]))

if __name__ == """__main__""": Charles()
```