```python
# -*- coding: cp1252 -*-
"""
The prime factors of 13195 are 5, 7, 13 and 29.

What is the largest prime factor of the number 600851475143 ?
"""




x=600851475143
i=2
while x!=1:
    while x%i==0:
        x=x/i
        print i,' and ' , x
    i=i+1
print 'The biggest prime factor is' ,i-1


"""
The answer is 6857
"""



"""
from itertools import *

firstPrimeGen = lambda magicnum: ifilter(lambda x: magicnum%x==0, chain((i for i in
[2]),count(3,2)))

def mainIter(magicnum):
    while magicnum > 1:
        n = firstPrimeGen(magicnum).next()
        yield n
        magicnum /= n

print max(list(mainIter(600851475143)))
"""



"""
sayi = 600851475143
asalBolen = 2
while sayi != 1:
```

```python
    if sayi%asalBolen == 0:
        sayi /=asalBolen
    else:
        asalBolen+=1
        continue
print("En büyük asal bölen =",asalBolen)
"""




"""
number = 600851475143

x = 0
prime = 0
check = 1
loop = True

while loop == True:
    x = x + 1
    if (number / x + 0.0) % 1 == 0.0:
        print(x)
        prime = x
        check = check * x
        if check == number:
            loop = False
    if x >= (number / 2):
        loop = False

print("")
print("The biggest prime factor equals " + str(prime))
"""




"""
#find the divisors in the form of a list
divisors=[]
n = 600851475143
i=1
while i<=n:
```

```python
        if n%i==0:
            n/=i
            divisors.append(i)
        i+=1
#find the list of prime divisors
pm=[]
for x in divisors:
    list=[]
    for i in range(2,int(sqrt(n))+1):
        while x%i==0 and len(list)<2:
            list.append(i)
    if len(list)==0:
        pm.append(x)
#Find the max prime
max(pm)
"""

"""

list1 = []

for n in range (2,600851475143):
    if 600851475143%n == 0:
        list1.append(n)

print(list1)

"""




"""
In Python3:

def problem3(num):
    list_num = []
    for i in range(2,math.floor(math.sqrt(num))+1):
        while  num % i == 0:
            list_num.append(i)
            num /= i
    print(max(list_num))

problem3(600851475143)
"""
```

```
"""
umber = 600851475143
for prime in range(2,600851475143):
    if (number %prime == 0):
        number = number/prime
        print(prime)
    elif (number == 1.0):
        break
"""
```

```
"""
from math import sqrt

def isPrimer(num):
    k = 0
    for i in range(2, num + 1):
        if num % i == 0:
            k += 1
    if k == 1:
        return 'Yes'

def largerPrimer(num):
    listP = []
    for i in range(int(sqrt(num)), 1, -1):
        if num % i == 0 and isPrimer(i) == 'Yes':
            listP += [i]
    return listP[0]

print('The largest prime factor of the number 600851475143: \n' +
str(largerPrimer(600851475143)))

"""
```

```
"""
```

```python
                    project euler 3(This is my method).py
import math
def MaxPrime(Prime):
        isPrime=lambda x: all(x % i != 0 for i in range(int(x**0.5)+1)[2:])
        for i in range(int(math.sqrt(Prime)),0,-1):
                if(Prime%i==0):
                        if(isPrime(i)):
                                return i
"""




"""
class prime_generator():
  '''
  prime_generator(num) -> generator

  Returns a generator that generates prime numbers up to num.
  '''
  def __init__(self, num):
    self.current_primes = set([])
    self.maximum = num
    self.working_num = 2

  def __iter__(self):
    return self

  def next(self):
    self.working_num += 1
    while not is_prime(self.working_num):

      if self.working_num in self.current_primes:
        return self.working_num

      self.working_num += 1

    self.current_primes.add(self.working_num)
    return self.working_num


def is_prime(num):
  '''
  is_prime(int) -> bool

  Finds if a number is a prime number.
  '''
```

```
  x = 3
  while x < num:
    if num % x == 0:
      return False

    x += 2

  return True

def largest_prime(num):
  '''
  largest_prime(int) -> int

  Returns the largest prime that a number is divisible by.
  '''
  x = num
  largest = 1
  while True:
    primes = prime_generator(x)
    for prime in primes:
      if x == prime:
        return x
      if not x % prime:
        x = x / prime
        largest = prime
        break

      if prime > x:
        break

  return largest

print largest_prime(600851475143)
"""




"""
My inefficient solution(8 minutes). If anyone know a way of optimize this solution,
could help me? Thanks in advance.

import math
from time import *

#Encontrar los numeros primos hasta un valor dado
def numPrimos(tope):
```

```python
        primos = [2]
        for num in range(3, tope+1, 2):
                aux = (num-1)/2
                if not any(num%x == 0 for x in primos):
                        primos.append(num)
        return primos

def factPrimos(num):
        raizCuad = int(round(math.sqrt(num)))
        primosHastaNum = numPrimos(raizCuad)
        for primo in primosHastaNum[::-1]:
                if num%primo == 0:
                        return primo

tiempoIni = time()
print (factPrimos(600851475143))
tiempoFin = time()
print (tiempoFin-tiempoIni) #8 minutes!!!!
"""




"""
def isPrime(n):
    for i in range(2,int(n**0.5)+1):
        if n%i==0:
            return False

    return True

n = 1

while n <= 600851475143:
  if 600851475143 % n == 0:
    if isPrime(n) == True:
        print n
  n += 2
"""




"""
def factor(n):
```

```python
from math import sqrt
Factor=list()
while n%2==0:
    Factor.append(2)
    n=n//2
factor=3
maxfactor=sqrt(n)
while n>1 and factor<=maxfactor:
    while n%factor==0:
        Factor.append(factor)
        n=n//factor
        maxfactor=sqrt(n)
    factor+=2
else:
    if n!=1:
        Factor.append(n)
return Factor

print(factor(600851475143)[-1])
"""


"""

number = 600851475143

def is_prime_number(x):
        for j in range(2,i):
                isPrime = True
                if i%j == 0:
                        isPrime = False
                        break
        return isPrime

for i in xrange(2,number):
        if number%i == 0 and is_prime_number(i) == True:
                print i

"""
```