

アルゴリズム論特講 (塩田)

2006年4月13日の課題

課題 次の問題を解く Python プログラムを作成せよ。

- 1 から n までの整数の 2 乗和が $111111 (= 3 \times 7 \times 11 \times 13 \times 37)$ で割り切れるような最小の自然数 n を求めよ。

提出期限 : 未定 (512 号室ポストまで)

- 参考
- 菊地先生の「専門コア情報処理演習」のページ:
<http://zope.lab.tkikuchi.net/edu/ci2004/>
 - M.Lutz, D.Ascher 著、紀太 章 訳、初めての Python、オライリー・ジャパン

関数定義のサンプル

```
# myfunction.py
# 関数定義部のみを記述したファイル
#
# シャープマークが注釈文の始まり

from math import *           # 数学関数を使うためのまじない
from random import *        # 乱数を使うためのまじない
from sys import *           # system 命令を使うためのまじない

# n が素数なら 1, そうでなければ 0 を返す関数
# def は関数宣言を表す ( 行末に : を書く )
# 引数や値には型宣言が要らない
def primetest(n):
    if n <= 1:                # 条件判定は、( ) は要らない
        return 0              # ブロック構造は字下げで表し、
                                # : のある見出し行の制御下に入る
                                # == は C 言語風

    if n == 2:
        return 1

    if n % 2 == 0:            # 剰余の % も C 言語風
        return 0

    h = 1
    m = sqrt(n)
    i = 3L                    # 最後に L をつけるといくらでも大きい
                                # and, or, not は PASCAL 風
    while i <= m and h == 1:
        if n % i == 0:
            h = 0
```

```

        else:
            i = i + 2
    if h == 1:
        return 1
    else:
        return 0

# 整数 a を自然数 m で割った余りを返す関数
def mod(a,m):
    a = a % m
    if a < 0:
        a = a + m
    return a
# a が負の時は余りを 0 以上
    に取り直す

# 自然数 n 未満の乱数ロング整数を返す関数
def rand(n):
    n0 = n
    x = 0L
    while n0 > 0:
        n0 = n0/10
        x = 10L * x + randint(0,9)
    return mod(x,n)
# randint(0,9) は 0 から 9 ま
での乱数整数

# k ビットの乱数ロング整数を返す関数
def randbit(k):
    x = 1L
    while k > 0:
        k = k - 1
        x = 2L * x + randint(0,1)
    return x

# n 以上の最小の素数を返す関数
def getprime(n):
    if n <= 2:
        return 2
    if n%2 == 0:
        n = n + 1
    while primetest(n) == 0:
        n = n + 2
    return n

# k ビット (以上) のランダムな素数を返す関数
def getprimebit(k):
    n = randbit(k)
    if n <= 2:
        return 2
    if n%2 == 0:
        n = n + 1
    while primetest(n) == 0:
        n = n + 2
    return n

```

メインプログラムのサンプル

```
#!/bin/env python
# 一行目は python プログラムとして実行するためのおまじない
#
# /home/shiota/sample1.py
#
# 実行命令は ./sample1.py
# 最初の実行の前に chmod +x を忘れずに

from math import *          # 数学関数を使うためのまじない
from myfunction import *    # myfunction.py に定義してある関数を使うためのまじない

# いきなりメインルーチンが始まる

print
print '素数判定のサンプルプログラムです。'

n = 100000000L
p = getprime(100000000)
print
print '----- その1 -----'
print 'n =',n,' を越える最小の素数は',p,' です。'
    # 長大な整数は末尾に L を書く

print
print '----- その2 -----'
n = input('好きな自然数を入力してください :')    # コンソール入力
                                                    # (末尾の L は要らない)

p = getprime(n)
print 'n =',n,' を越える最小の素数は',p,' です。'

n = 1000
ls = []          # 空配列
print
print '----- その3 -----'
print n,' までの素数表を作ります。'
c = 0;
for i in range(2,n+1):    # range(i,j) は i 以上 j 未満の整数の
    集合                # 1 ずつ制御変数の増える for 文はこう書
    <                    # (複雑な制御をするループは while 文で
    書く)
        if primetest(i) == 1:
            ls.append(i)    # i が素数なら配列 ls に追加してゆく
            c = c + 1

print ls
print n,' までに素数は',c,' 個ありました。'
```