素数と三角形

定理1
3で割ると1余る素数は、120°整数三角形の120°の対辺になる
7, 13, . . .

3で割ると2余る素数は、120°整数三角形の120°の対辺にならならない
5, 11, 17

定理1'
3で割ると1余る素数を(重複を許して)かけあわせて作られる自然数は、120°整数三角形で、3辺の長さの最大公約数が1となるものの、120°の対辺になる
49, . . . 91

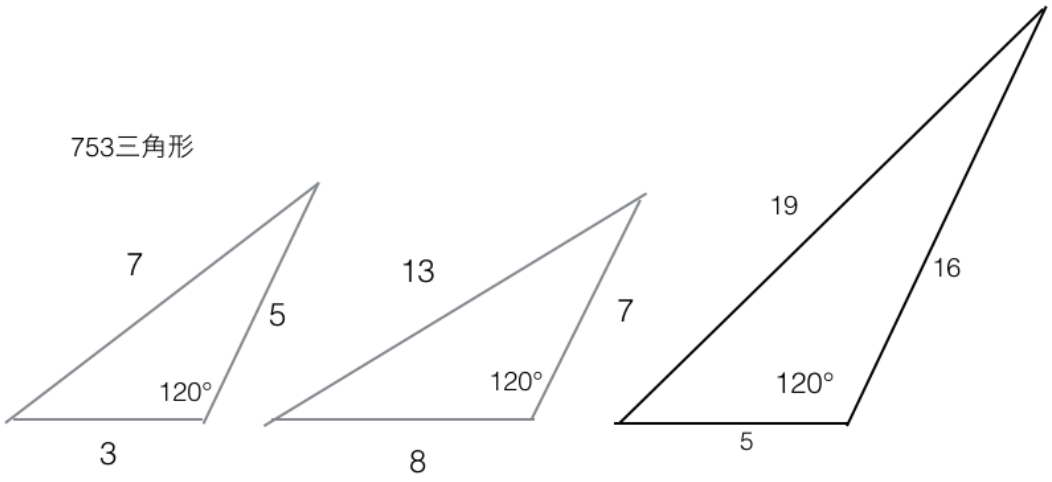
定理2
4で割ると1余る素数は、整数直角三角形の斜辺に なる
5, 13, 17,

定理2'
4で割ると1余る素数を(重複を許して)かけあわせて作られる自然数は、整数直角三角形で、3辺の長さの最大公約数が1となるものの、斜辺の対辺になる
25, 65

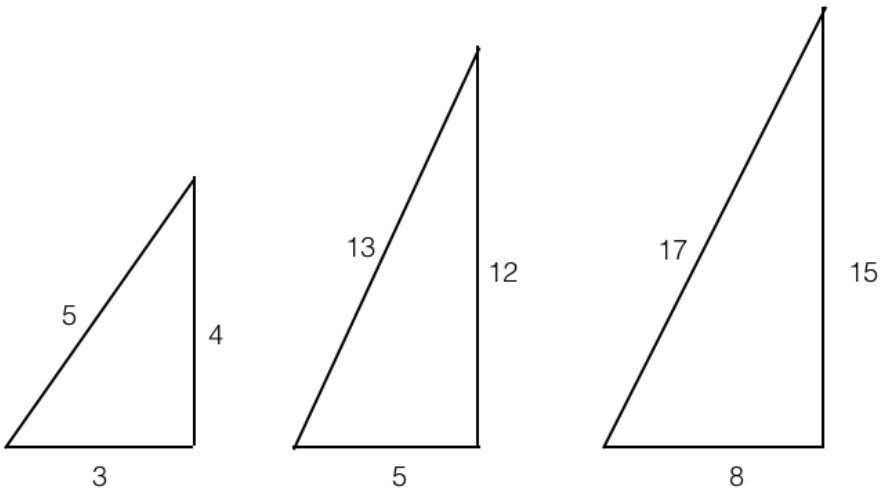
定理3
4で割ると1余る素数は、a^2 + b^2 の形にかける
5(2,1), 13(3,2), 17,(4,1) 29(5,2), 37(6,1), 41(5,4) ...

4で割ると3余る素数は、a^2 + b^2 の形にかけない
7

753三角形



3で割ると1余る素数



4で割ると1余る素数

抽出した素数から、さらに３で割ると1余る素数と４で割ると1余る素数を選別するプログラム

```
#include <stdio.h>
// Find prime numbers - n個の素数を見つける
/* 3で割って 1余る素数 */
int mod3(int m){
  return m % 3;
  }
/* 4で割って1余る素数 */
int mod4(int m){
   return m % 4;
   }

/*  エラトステネスの篩で素数を抽出 */
int main()
{
   int n, i = 3, count, c;

   printf("Enter the number of prime numbers required\n");
   scanf("%d",&n);

   if ( n >= 1 )
   {
      printf("First %d prime numbers are :\n",n);
      printf("2\n");
   }

   for ( count = 2 ; count <= n ;  )
   {
      for ( c = 2 ; c <= i - 1 ; c++ )
      {
         if ( i%c == 0 )
            break;
      }
      if ( c == i )
      {

         printf("%d ",i);
         count++;

// 素数の分類
         printf(" %d ", mod3(i));
         printf(" %d \n", mod4(i));
      }
      i++;
   }
```

```
   return 0;
}
```

計算結果:

```
Enter the number of prime numbers required
First 100 prime numbers are :
2
3  0  3
5  2  1
7  1  3
11  2  3
13  1  1
17  2  1
19  1  3
23  2  3
29  2  1
31  1  3
37  1  1
41  2  1
43  1  3
47  2  3
53  2  1
59  2  3
61  1  1
67  1  3
71  2  3
73  1  1
79  1  3
83  2  3
89  2  1
97  1  1
101  2  1
```