lab04

```
$ gcc lab04.c
lab04.c:24:20: warning: format specifies type 'int' but the argument has type 'long'
 [-Wformat]

                            k++, n, a, b);

                                 ^
lab04.c:24:23: warning: format specifies type 'int' but the argument has type 'long'
 [-Wformat]

                            k++, n, a, b);

                                    ^
lab04.c:24:26: warning: format specifies type 'int' but the argument has type 'long'
 [-Wformat]

                            k++, n, a, b);

                                       ^
3 warnings generated.
$ ./a.out
1: 2 ^ 3 = 2 ^ 2 + 2 ^ 2
2: 5 ^ 3 = 5 ^ 2 + 10 ^ 2
3: 8 ^ 3 = 16 ^ 2 + 16 ^ 2
...
...
1399: 4993 ^ 3 = 159776 ^ 2 + 314559 ^ 2
1400: 4996 ^ 3 = 149880 ^ 2 + 319744 ^ 2
1401: 5000 ^ 3 = 50000 ^ 2 + 350000 ^ 2
1401 solutions found for n <= 5000.
CPU time: 0.0326636 sec
score: 74
o. Compilation warnings.
o. [Output] Program output is correct, good.
o. [Format] Program format can be improved
o. [Coding] lab04.c spelling errors: provement(1)
o. [Efficiency] can be improved.
```

lab04.c

```c
//EE231002 Lab04. Solving Integer Equations
// EE231002 Lab04. Solving Integer Equations
//109061158, 簡佳吟
// 109061158, 簡佳吟
//Date: 2020.10.19
// Date: 2020/10/19


#include <stdio.h>
#define MAX 5000
Need a blank line here.
int main(void)
{
    long int n, a, b;       // satisfies the given equation n^3 = a^2 + b^2
    long int x, y;          /* satisfies a = x * (x^2 + y^2)
                                          b = y * (x^2 + y^2) */
    int k = 1;              // serial number of solution


    for (n = 2; n <= MAX ; n++)  {                  // scan for 1 < n < MAX
    for (n = 2; n <= MAX; n++)  {                  // scan for 1 < n < MAX
        if (n % 2 == 0  || n % 4 == 1) {            /* n must be a square of
                                                    some number */
This line has more than 80 characters
            for (x = 1; x <= 71; x++) {             // scan x from 1 to 71
                    Why 71?
                for (y = x; y <= 71; y++) {         // scan y form x to 71
                    Why 71?
                    a = x * (x * x + y * y);        // assign a and b
                    b = y * (x * x + y * y);
                    if ((a * a + b * b) == (n * n * n)) {       // select
                        printf("%d: %d ^ 3 = %d ^ 2 + %d ^ 2\n",
                                                        k++, n, a, b);
                                                        // prompt
                        x = 72;
                        y = 72;                     // jump out of the loop
                    }

                }

```

```
32              }
33
34
35          }
36
37      }
38      printf("%d solutions found for n <= %d.\n", k - 1, MAX);    // prompt
39      return 0;                                                  // done and return
40  }
41  /* provement
42   * Let x, y, z be positive  integers which satisfy the equation
43   * x ^ 2 + y ^ 2 = Z ^ 2
44   * multiply both sides by z ^ 4,and we can get the equation
45   * (x ^ 2 * z ^ 4) + (y ^ 2 * z ^ 4) = z ^ 6
46   * (x * z ^ 2) ^ 2 + (y * z ^ 2) = (z ^ 2) ^ 3
47   * (x * (x ^ 2 + y ^ 2)) ^ 2 + (y * (x ^ 2 + y ^ 2)) ^ 2
48   *                                 = ((x ^ 2 + y ^ 2) ^ 3
49   * Then, let a = x * (x ^ 2 + y ^ 2) and b = y * (x ^ 2 + y ^ 2)
50   * and n = x ^ 2 + y ^ 2, which is square of some integer
51   * if n is a square of an even number, n % 2 = 0
52   * if n is a square of an odd number,
53   * let the odd number be 2k+1, and k is a positive integer
54   * (2k+1) ^ 2 = 4k ^ 2 + 4k + 1
55   * Thus, (2k+1) ^ 2 % 4 = 1 */
56
57
58
59
60
61
62
63
64
65
```