

JT-69 比 N 大的最小快樂數

題目

把一個整數的所有數字 (digits) 的平方相加，得到一個新數，然後再次求所有數字的平方和，如此重複進行，如果最後可以得到 1，則稱該數為快樂數，如果不能得到 1，則不是快樂數。

例如，

```
28 → 4+64=68 → 36+64=100 → 1+0+0=1
32 → 9+4=13 → 1+9=10 → 1+0=1
37 → 9+49=58 → 25+64=89 → 64+81=145 → 1+16+25=42 → 16+4=20 → 4+0=4
    → 16 → 1+36=37.....
```

因此 28 和 32 是快樂數，而在 37 的計算過程中，37 重覆出現，繼續計算的結果只會是上述數字的循環，不會出現 1，因此 37 不是快樂數。

輸入：整數 N。

輸出：比 N 大的最小快樂數？

解題思惟

1. 這一題與課堂上所講的<JT-55 快樂數>類似，JT-55 是求 1~n 有幾個快樂數，本題是求比 n 大的最小快樂數。可以用相同方式來解答。先寫出 `is_happy(int n)` 的函數，用來檢查 n 是否為快樂數。
2. 接著從 n+1 開始往上檢查，檢查到的第一個快樂數就是答案了。

程式碼

```
#include <stdio.h>

int d_square(int n);
int is_happy(int n);

int main()
{
    int i, n;
    scanf("%d", &n);
    for (i=n+1; ; i++) {
        if (is_happy(i)) break;
    }
    printf("%d", i);
    return 0;
}

int d_square(int n)
{
    int ans=0;
    while (n) {
        ans += (n%10)*(n%10); // 加上個位數平方
        n /= 10;             // n 去掉個位
    }
    return ans;
}
```

```
int is_happy(int n)
{
    int a[1000] = {0};
    if (n==1) return 1;
    while (1) {
        n = d_square(n);      // n 變成數字平方和
        if (n==1) return 1;    // 和為1，回傳1
        if (a[n]==1) return 0; // 重覆，回傳0
        a[n]=1;
    }
}
```