

JT-55 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 重複出現，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：從 1 到 N 有幾個快樂數？

解題思惟

1. 整數 x 的數字平方和會有多大？如果設 x 為 n 位數，每個數字不超過 9，則數字平方和不超過 $9*9*n=81*n$ ，如果 n 是 10 位數的話（整數的最大值是 10 位數），也不過 810。所以在計算數字平方和的時候，值應該不會超過 1000。
2. 算出數字平方和之後，要繼續重複計算，直到最後和為 1（快樂數），或者重複出現相同的數字為止（非快樂數）。
3. 怎麼知道計算過程中有沒有出現重複呢？基本的想法，可以設一個陣列 $a[1000]$ ，一開始全設為 0，如果該數的數字平方和為 k ，就把 $a[k]$ 設為 1。那麼計算過程中，如果發現數字平方和為 x ，且 $a[x]=1$ ，即表示已經重複，反之，如果 $a[x]=0$ ，表示還沒有重複過，那就把 $a[x]$ 設為 1。重複到最後，要嘛數字和為 1，再不然就是碰到重複。（為什麼？）
4. 這樣就可以寫一個函數 `int is_happy(int n)`，來判斷 n 是否為快樂數了。是的話回傳 1，不是的話回傳 0。主程式就很容易寫了。
5. 為了方便，還可以寫一個函數 `int d_square(int n)`，來幫我們算 n 的數字平方和。

程式碼

算數字平方和

```
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;
}
```

檢查是否為快樂數

```
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;
    }
}
```

主程式

```
#include <stdio.h>

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

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