HDOJ 1757 – A Simple Math Problem

——by A Code Rabbit

Description

给出一个递推式，求 f(x) 对 m 取模的结果。

Types

Maths :: Matrix

Analysis

很典型的矩阵乘法和快速幂的题目，需要 10 * 10 的矩阵。矩阵也是很经典的～

矩阵如下：

$$\begin{pmatrix}
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}$$

在两边添加递推式相邻的十项就可以用了，左边是 f(n) + ... + f(n – 9)，右边是 f(n – 1) + ... + f(n – 10)。

Solution

```
// HDOJ 1757
// A Simple Math Problem
// by A Code Rabbit

#include <cstdio>

const int ORDER = 10;

int k, m;
int a[ORDER];
```

```
const int f[] = {
    9, 8, 7, 6, 5, 4, 3, 2, 1, 0,
};

struct Matrix {
    int element[ORDER][ORDER];
};

Matrix mat_one;
Matrix mat_ans;

void INIT();

void Multiply(Matrix& mat_a, Matrix mat_b);

int main() {
    while (scanf("%d%d", &k, &m) != EOF) {
        // Inputs.
        for (int i = 0; i < ORDER; i++) {
            scanf("%d", &a[i]);
        }
        // INIT.
        INIT();
        // Judge special situations.
        if (k < 10) {
            printf("%d\n", k % m);
            continue;
        }
        // Quick Power.
        k -= 9;
        while (k) {
            if (k & 1) {
                Multiply(mat_ans, mat_one);
            }
            Multiply(mat_one, mat_one);
            k >>= 1;
        }
        // Compete.
        int ans = 0;
        for (int i = 0; i < ORDER; ++i) {
            ans += f[i] * mat_ans.element[i][0];
        }
        // Outputs.
        printf("%d\n", ans % m);
    }

    return 0;
}

void INIT() {
    for (int i = 0; i < ORDER; ++i) {
        mat_one.element[i][0] = a[i];
    }
    for (int i = 0; i < ORDER; ++i) {
        for (int j = 1; j < ORDER; ++j) {
            mat_one.element[i][j] = i == j - 1 ? 1 : 0;
        }
    }
    for (int i = 0; i < ORDER; ++i) {
        for (int j = 0; j < ORDER; ++j) {
            mat_ans.element[i][j] = i == j ? 1 : 0;
        }
    }
}

void Multiply(Matrix& mat_a, Matrix mat_b) {
    Matrix mat_c;
    for (int i = 0; i < ORDER; ++i)
        for (int j = 0; j < ORDER; ++j) {
            mat_c.element[i][j] = 0;
            for (int k = 0; k < ORDER; ++k) {
                mat_c.element[i][j] += mat_a.element[i][k] * mat_b.element[k][j];
            }
            mat_c.element[i][j] %= m;
        }
    mat_a = mat_c;
}
```