행렬 거듭제곱

Description

행렬의 곱셈을 여러 번 하면 행렬의 거듭제곱을 구할 수 있다.

예를 들어, 행렬 A의 5 거듭제곱은 다음과 같이 계산할 수 있다.

A^5 = A * A * A * A * A

양의 정수 N과 2 * 2 행렬 A가 주어졌을 때, 이 행렬의 N 거듭제곱을 출력하시오.

단, 계산과정에서 행렬의 원소는 -2^32 <= A[i][j] <= 2^32 범위에 있는 테스트 케이스만 주어진다고 가정해도 된다.

거듭제곱을 재귀적으로 정의하면 다음과 같다.

```
void power(int n, lont_t A[][2], lont_t B[][2]) {
    if (n == 1) {
        // exit condition: copy elements of B to A;
    }
    else {
        // recursive condition:
        lont_t T[2][2];
        // Let T be A powered by n - 1.
        // Multiply A with T;
    }
}
```

다음 코드에서 mult() 함수와 power() 함수를 구현하여 제출하시오.

```
#include <stdio.h>

typedef long long lont_t;

/* muliplies two matrices A and B into a matrix C */
void mult(lont_t A[][2], lont_t B[][2], lont_t C[][2]);

/* powers a matrix A with exponent n into a matrix B */
void power(int n, lont_t A[][2], lont_t B[][2]);

int main()
{
    int n;
    scanf("%d", &n);
    lont_t A[2][2], B[2][2];
    scanf("%lld %lld %lld %lld", &A[0][0], &A[0][1], &A[1][0], &A[1][1]);
    power(n, A, B);
    printf("%lld %lld \n%lld %lld ", B[0][0], B[0][1], B[1][0], B[1][1]);
}
```

Input

첫 번째 줄에 양의 정수 N이 주어진다. 1 <= N <= 1024

두 번째 줄에 행렬 A의 원소 두 개가 주어진다. (a00, a01)

세 번째 줄에 행렬 A의 원소 두 개가 주어진다.(a10, a11)

행렬의 원소는 정수값으로만 주어진다.

Output

행렬 A의 N 거듭제곱 행렬 B를 출력한다.

첫 번째 줄에 행렬 B의 원소 두 개를 출력한다.

두 번째 줄에 행렬 B의 원소 두 개를 출력한다.

Sample Input 1

```
2
1 1
1 1
```

Sample Output 1

```
2 2
2 2
```

Sample Input 2

```
32
1 1
1 1
```

Sample Output 2

```
2147483648 2147483648
2147483648 2147483648
```

Sample Input 3

```
32
-1 1
1 -1
```

Sample Output 3

```
2147483648 -2147483648
-2147483648 2147483648
```

Language: C

Theme: Solarized Light

```
void mult(lont_t A[][2], lont_t B[][2], lont_t C[][2])
{
    C[0][0] = A[0][0] * B[0][0] + A[0][1] * B[1][0];
    C[0][1] = A[0][0] * B[0][1] + A[0][1] * B[1][1];
    C[1][0] = A[1][0] * B[0][0] + A[1][1] * B[1][0];
    C[1][1] = A[1][0] * B[0][1] + A[1][1] * B[1][1];
}

void power(int n, lont_t A[][2], lont_t B[][2])
{
    if (n == 0)
    {
        B[0][0] = 1;
        B[0][1] = 0;
        B[1][0] = 0;
        B[1][1] = 1;
        return;
    }

    long long temp[2][2];
    power(n / 2, A, temp);
    mult(temp, temp, B);

    if (n % 2 == 1)
    {
        long long temp2[2][2];
        mult(A, B, temp2);
        B[0][0] = temp2[0][0];
        B[0][1] = temp2[0][1];
        B[1][0] = temp2[1][0];
        B[1][1] = temp2[1][1];
    }
}
```

You have solved the problem

Submit

Contest has ended