1006 – Hex-a-bonacci

Given a code (not optimized), and necessary inputs, you have to find the output of the code for the inputs. The code is as follows:

```
int a, b, c, d, e, f;
int fn( int n ) {
    if( n == 0 ) return a;
    if( n == 1 ) return b;
    if( n == 2 ) return c;
    if( n == 3 ) return d;
    if( n == 4 ) return e;
    if( n == 5 ) return f;
    return( fn(n-1) + fn(n-2) + fn(n-3) + fn(n-4) + fn(n-5) + fn(n-6) );
}
int main() {
    int n, caseno = 0, cases;
    scanf("%d", &cases);
    while( cases-- ) {
        scanf("%d %d %d %d %d %d %d", &a, &b, &c, &d, &e, &f, &n);
        printf("Case %d: %d\n", ++caseno, fn(n) % 10000007);
    }
    return 0;
}
```

Input

Input starts with an integer **T (≤ 100)**, denoting the number of test cases.

Each case contains seven integers, **a, b, c, d, e, f and n**. All integers will be non-negative and **0 ≤ n ≤ 10000** and the each of the others will be fit into a 32-bit integer.

Output

For each case, print the output of the given code. The given code may have integer overflow problem in the compiler, so be careful.

Sample Input	Output for Sample Input
5 0 1 2 3 4 5 20 3 2 1 5 0 1 9 4 12 9 4 5 6 15 9 8 7 6 5 4 3 3 4 3 2 54 5 4	Case 1: 216339 Case 2: 79 Case 3: 16636 Case 4: 6 Case 5: 54