#include <cstdio>

#include <algorithm>

#include <cstring>

using namespace std;

struct BigInteger{

int A[25];

enum{MOD = 10000};

BigInteger(){memset(A, 0, sizeof(A)); A[0]=1;}

void set(int x){memset(A, 0, sizeof(A)); A[0]=1; A[1]=x;}

void print(){

printf("%d", A[A[0]]);

for (int i=A[0]-1; i>0; i--){

if (A[i]==0){printf("0000"); continue;}

for (int k=10; k\*A[i]<MOD; k\*=10) printf("0");

printf("%d", A[i]);

}

printf("\n");

}

int& operator [] (int p) {return A[p];}

const int& operator [] (int p) const {return A[p];}

BigInteger operator + (const BigInteger& B){

BigInteger C;

C[0]=max(A[0], B[0]);

for (int i=1; i<=C[0]; i++)

C[i]+=A[i]+B[i], C[i+1]+=C[i]/MOD, C[i]%=MOD;

if (C[C[0]+1] > 0) C[0]++;

return C;

}

BigInteger operator \* (const BigInteger& B){

BigInteger C;

C[0]=A[0]+B[0];

for (int i=1; i<=A[0]; i++)

for (int j=1; j<=B[0]; j++){

C[i+j-1]+=A[i]\*B[j], C[i+j]+=C[i+j-1]/MOD, C[i+j-1]%=MOD;

}

if (C[C[0]] == 0) C[0]--;

return C;

}

};

int main() {

BigInteger a, b;

a.set(1); b.set(1);

(a+b).print();

return 0;

}