Si te da

|  |  |  |  |
| --- | --- | --- | --- |
| Puntos |  | Límite de memoria | 128MB |
| Límite de tiempo (caso) | 1s | Límite de tiempo (total) | 60s |

No tienes nada que hacer y un amigo te comento que existe un problema de teoría de números que se te ha hecho muy interesante y quieres resolverlo a través de una programa de computo.

**Problema**

Se tienen 2 números de un digito cada uno, el tercer numero se forma de la suma de los 2 números anteriores, tomando de el, solo la posición del digito, y así sucesivamente hasta que el ciclo regresa a los 2 numero originales.

**Entrada**

Dos números enteros de un solo dígito, en líneas separadas.

**Salida**

En la primera línea tu programa deberá escribir la serie completa separada por espacios. En la segunda línea tu programa deberá imprimir el número de pasos para regresar a los números iniciales.

**Ejemplo**

| **Entrada** | **Salida** |
| --- | --- |
| 1  8 | 1 8 9 7 6 3 9 2 1 3 4 7 1 8  12 |

**Consideraciones**

Asume que un dígito es cualquier número 0≤n≤9

*Fuente: OMI Training Gate*

Problema subido por: [luison.cpp](https://omegaup.com/profile/luison.cpp/)

| **ID** | **Status** | **Porcentaje** | **Penalty** | **Lenguaje** | **Memoria** | **Tiempo** | **Detalles** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| [Nuevo envío](https://omegaup.com/arena/problem/siteda#problems/new-run) | | | | | | | | |
| bcd7e2b3 | Respuesta correcta | 100.00% | 0 | cpp | 2.97 | 0.00 |  |  |
| **Envíos** | | | | | | | |

<https://omegaup.com/arena/problem/siteda#problems>

#include <iostream>

#include <stdio.h>

#include <vector>

#include <algorithm>

#include <conio.h>

using namespace std;

int main() {

//int a=1, b=8;

int a,b;

scanf("%d %d", &a, &b);

std::vector<int> nums;

nums.push\_back(a);

nums.push\_back(b);

int i;

for ( i = 2; ; i++)

{

nums.push\_back((nums[i-1] + nums[i - 2]) % 10);

if (nums[i - 1] == a && nums[i] == b)

{

break;

}

}

for(int i =0; i < nums.size(); i++) {

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

}

printf("\n%d", i-1);

getch();

return 0;

}