

2.8

```
//*****  
// IMPORTANTE  
//  
// Ana Álava Papí  
// E01  
// E01  
//  
//*****
```

```
#include <iostream>  
#include <fstream>  
#include <vector>  
#include <string>  
#include <cmath>  
#include <algorithm>
```

```
//*****  
// Aqui explicacion del algoritmo y calculo del coste.  
// Llamo a la función recursivamente modificando en cada vez el número n por n/10 para coger  
// el último dígito, y calculando sumPrec.  
// Cuando ya he recorrido el número entero, compruebo que divide tanto a sumPrec como a  
// sumSuc y a su vez voy calculando la sumSuc del resto
```

```
//  
//  
// Coste es O(n)  $O(\log n) - 0.2$   
/*  
    T(n) = c                n=1  
    T(n) = T(n/10) + c      n > 1    0.3  
*/  
//*****
```

```
// Aqui la funcion que resuelve el problema
```

```
bool resolver(int n, int sumaPre, int& sumaSuc, int& ant) {  
    if (n >= 10)  
        resolver(n / 10, sumaPre + n % 10, sumaSuc, ant);
```

```
    int aux = n * 10;  
    n -= ant;  
    ant = aux;
```

```
    if (sumaPre % n == 0 && sumaSuc % n == 0) {  
        sumaSuc += n;  
        return true;  
    }  
    else return false;
```

```
}
```

→ Hay que obtener el resultado de la función y si es falso devolver falso.
- 0.5

```

// Para lectura de datos y mostrar los resultados
void resuelveCaso() {
    // Lectura de datos
    int n, sumaPre, sumaSuc, ant;

    std::cin >> n;

    sumaPre = 0;
    sumaSuc = 0;
    ant = 0;

    // LLamar a la función que resuelve el problema
    if (resolver(n, sumaPre, sumaSuc, ant))
        std::cout << "SI";
    else
        std::cout << "NO";

    std::cout << "\n";
    // Escribir el resultado
}

```

```

int main() {
    // Para redireccionar la entrada a un fichero
#ifdef DOMJUDGE
    std::ifstream in("E2.txt");
    auto cinbuf = std::cin.rdbuf(in.rdbuf()); //save old buf and redirect std::cin to casos.txt
#endif

    int numCasos; std::cin >> numCasos; std::cin.ignore();
    for (int i = 0; i < numCasos; ++i)
        resuelveCaso();

#ifdef DOMJUDGE // para dejar todo como estaba al principio
    std::cin.rdbuf(cinbuf);
    system("PAUSE");
#endif

    return 0;
}

```