Find the smallest number whose digits multiply to a given number n

Given a number ‘n’, find the smallest number ‘p’ such that if we multiply all digits of ‘p’, we get ‘n’. The result ‘p’ should have minimum two digits.

Examples:

```
Input:  n = 36
Output: p = 49
// Note that 4*9 = 36 and 49 is the smallest such number

Input:  n = 100
Output: p = 455
// Note that 4*5*5 = 100 and 455 is the smallest such number

Input: n = 1
Output:p = 11
// Note that 1*1 = 1

Input: n = 13
Output: Not Possible
```

For a given n, following are the two cases to be considered.
Case 1: n < 10 When n is smaller than n, the output is always n+10. For example for n = 7, output is 17. For n = 9, output is 19.

Case 2: n >= 10 Find all factors of n which are between 2 and 9 (both inclusive). The idea is to start searching from 9 so that the number of digits in result are minimized. For example 9 is preferred over 33 and 8 is preferred over 24.
Store all found factors in an array. The array would contain digits in non-increasing order, so finally print the array in reverse order.

Following is C implementation of above concept.

```
#include<stdio.h>

// Maximum number of digits in output
#define MAX 50

// prints the smallest number whose digits multiply to n
void findSmallest(int n)
{
    int i, j=0;
    int res[MAX]; // To sore digits of result in reverse

    // Case 1: If number is smaller than 10
    if (n < 10)
    {
        printf("%d", n+10);
        return;
    }

    // Case 2: Start with 9 and try every possible digit
    for (i=9; i>1; i--)
    {
        // If current digit divides n, then store all
        // occurrences of current digit in res
        while (n%i == 0)
        {
            n = n/i;
            res[j] = i;
            j++;
        }
    }

    // If n could not be broken in form of digits (prime
    // are greater than 9)
    if (n > 10)
    {
        printf("Not possible");
        return;
    }

    // Print the result array in reverse order
    for (i=j-1; i>=0; i--)
        printf("%d", res[i]);
}

// Driver program to test above function
int main()
```

```
{
    findSmallest(7);
    printf("\n");

    findSmallest(36);
    printf("\n");

    findSmallest(13);
    printf("\n");

    findSmallest(100);
    return 0;
}
```

Output:

```
17
49
Not possible
455
```