Lab session 2: Programming Fundamentals

In this lab session, you are not allowed to use any function from the math library. You are only allowed to make use of constructs that are taught in the first two weeks of the course (so no arrays, functions, recursion, etc).

Problem 1: Smoking problem

A poor homeless man has a smoking problem. He does not have money to buy cigarettes. On the streets he searches for cigarette butts. He can make exactly one cigarette out of four cigarette butts. One day he finds 31 butts. How many cigarettes can he smoke that day?

Well, from 31 butts he can make 7 cigarettes (using $7 \times 4 = 28$ butts, leaving 3 butts). So, after having smoked these 7 cigarettes he has 7 new butts, totalling $7 + 3 = 10$ butts, which is enough to make another 2 cigarettes (leaving $10 - 2 \times 4 = 2$ butts). After having smoked these two cigarettes, he again has two new butts. Together with the remaining 2 butts, he can make another (last) cigarette. So, in total he can smoke $7 + 2 + 1 = 10$ cigarettes.

Write a program that reads from the input a positive integer n, which is the number of butts that the man finds on some day. The output should be the number of cigarettes that he can smoke that day. You may assume that $0 \leq n < 1000000$.

Example 1:

input:
```
3
```
output:
```
0
```

Example 2:

input:
```
4
```
output:
```
1
```

Example 3:

input:
```
31
```
output:
```
10
```

Problem 2: Divisible by 13

To check if a positive integer n is divisible by 13, we have to add four times its last digit to the number that is obtained by removing the last digit. We repeat this process until we get a two-digit number. The number n is divisible by 13 if and only if this two-digit number is divisible by 13.

For example, the number 2795 is converted into the number $279 + 4 \times 5 = 299$. Since 299 is a three-digit number, we repeat this step to obtain $29 + 4 \times 9 = 65$. This is a two-digit number, and we find that $65 = 5 \times 13 + 0$, so 2795 is divisible by 13.

The input for this problem is a positive `int` n. The output of your program must show the computation steps to determine whether n is divisible by 13 and should also print the conclusion. Make sure that the output is in the format as given in the following examples. The output should not contain any white space (tabs or spaces).

Example 1:

input:

```
2795
```

output:

```
2795->299->65=5*13+0
YES
```

Example 2:

input:

```
2796
```

output:

```
2796->303->42=3*13+3
NO
```

Example 3:

input:

```
26
```

output:

```
26=2*13+0
YES
```

Problem 3: Divisible by 11

There exists also an easy way to test for divisibility by 11. The only thing you need to do is to compute the difference of the sum of digits in odd places and the sum of digits in even places. If it results in the number 0, then the original number is divisible by 11.

If it results in a negative number, then take the absolute value of this result. If this number is less than 11, then the original number is not divisible by 11. Otherwise, repeat the process with the obtained result.

For example, for 31415 the difference of the sums is $3 - 1 + 4 - 1 + 5 = 10$. This is non-zero and less than 11, so 31415 is not divisible by 11. For 2728 we find the alternating sum $2 - 7 + 2 - 8 = -11$. We take the absolute value 11, and repeat the process: $1 - 1 = 0$. The conclusion is that 2728 is divisible by 11.

The input for this problem is a positive integer n (where $11 \leq n \leq 10^{1000}$). The output of your program must show the computation steps to determine whether n is divisible by 11 and should also print the conclusion. Make sure that the output is in the format as given in the following examples. The output should not contain any white space (tabs or spaces).

Example 1:

input:

```
31415
```

output:

```
+3-1+4-1+5=10
NO
```

Example 2:

input:

```
2728
```

output:

```
+2-7+2-8=-11
+1-1=0
YES
```

Example 3:

input:

```
1909090
```

output:

```
+1-9+0-9+0-9+0=-26
+2-6=-4
NO
```

Problem 4: Mulpals

A positive integer n is called a *mulpal* (multiplicative palindrome) if there exists an integer d such that $d \cdot n$ equals n read in reverse direction (in decimal notation). For example, $2178 \times 4 = 8712$, so $n = 2178$ is a mulpal.

Write a program that reads two integers a and b from the input (you may assume that $1 \leq a < b < 10000000 = 10^7$), and outputs the number of mulpals n with $a \leq n \leq b$.

Example 1:

input:

```
1 10
```

output:

```
9
```

Example 2:

input:

```
1 100
```

output:

```
18
```

Example 3:

input:

```
1 1000
```

output:

```
108
```

Problem 5: K-composites

We call a positive integer a *k-composite* if it is divisible by the numbers $1, 2, 3, \ldots, k$. For example, the number 2520 is the smallest number that is divisible by $1, 2, 3, .., 9$, and 10.

Write a program that reads from the input a value k (where $1 \leq k \leq 22$), and outputs the smallest k-composite. Note that the smallest 22-composite is 232792560, which fits in a standard `int`. Nevertheless, be careful not to overflow your calculations. It is not allowed to submit a program that contains precomputed answers.

Example 1:

input:

```
3
```

output:

```
6
```

Example 2:

input:

```
5
```

output:

```
60
```

Example 3:

input:

```
10
```

output:

```
2520
```