Problem G. Optimal Coin Change

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1s

In a 10-dollar shop, everything is worthy 10 dollars or less. In order to serve customers more effectively at the cashier, change needs to be provided in the minimum number of coins.

In this problem, you are going to provide a given value of the change in different coins. Write a program to calculate the number of coins needed for each type of coin.

The input includes a value v, a size of the coinage set n, and a face value of each coin, $f_1, f_2, \cdots, f_n$. The output is a list of numbers, namely, $c_1, \cdots, c_n$, indicating the number of coins needed for each type of coin. There may be many ways for the change. The value v is an integer satisfying $0 < v \leq 2000$, representing the change required in cents. The face value of a coin is less than or equal to 10000. The output of your program should take the combination with the least number of coins needed.

For example, the Hong Kong coinage issued by the Hong Kong Monetary Authority consists of 10 cents, 20 cents, 50 cents, 1 dollar, 2 dollars, 5 dollars and 10 dollars would be represented in the input by $n = 7$, $f_1 = 10$, $f_2 = 20$, $f_3 = 50$, $f_4 = 100$, $f_5 = 200$, $f_6 = 500$, $f_7 = 1000$.

Input

The test data may contain many test cases, please process it to the end of the file.

Each test case contains integers $v, n, f_1, \cdots, f_n$ in a line. It is guaranteed that $n \leq 10$ and $0 < f_1 < f_2 < \cdots < f_n$.

Output

The output be n numbers in a line, separated by space. If there is no possible change, your output should be a single -1. If there are more than one possible solutions, your program should output the one that uses more coins of a lower face value.

Example

standard input	standard output
2000 7 10 20 50 100 200 500 1000	0 0 0 0 0 0 2
250 4 10 20 125 150	0 0 2 0
35 4 10 20 125 150	-1
48 4 1 8 16 20	0 1 0 2
40 4 1 10 13 37	3 0 0 1
43 5 1 2 21 40 80	1 1 0 1 0