Some phone usage rate may be described as follows:

- first minute of a call costs `min1` cents,
- each minute from the 2nd up to 10th (inclusive) costs `min2_10` cents
- each minute after 10th costs `min11` cents.

You have `s` cents on your account before the call. What is the duration of the longest call (in minutes rounded down to the nearest integer) you can have?

Example

For `min1 = 3`, `min2_10 = 1`, `min11 = 2` and `s = 20`, the output should be

`phoneCall(min1, min2_10, min11, s) = 14`.

Here's why:

- the first minute costs `3` cents, which leaves you with `20 - 3 = 17`cents;
- the total cost of minutes `2` through `10` is `1 * 9 = 9`, so you can talk `9` more minutes and still have `17 - 9 = 8` cents;
- each next minute costs `2` cents, which means that you can talk `8 / 2 = 4` more minutes.

Thus, the longest call you can make is `1 + 9 + 4 = 14` minutes long.

Input/Output

- **[execution time limit] 4 seconds (py)**
-
- **[input] integer min1**
- *Guaranteed constraints:*
- `1 ≤ min1 ≤ 10`.
-
- **[input] integer min2_10**
- *Guaranteed constraints:*
- `1 ≤ min2_10 ≤ 10`.
-
- **[input] integer min11**
- *Guaranteed constraints:*
- `1 ≤ min11 ≤ 10`.
-
- **[input] integer s**
- *Guaranteed constraints:*
- `2 ≤ s ≤ 60`.
-
- **[output] integer**