Problem B. Golden Medals

The task is to calculate $\lceil n/12 \rceil$. There are lot of ways to do that (for example, check if n is divisible by 12, and choose $n/12$ or $n/12+1$, or just print $(n+11)/12$, in both case the division is integer division like one implemented in C/C++).

Problem D. Key Fragments

To solve the problem, we use two arrays:

1. `subs[]`, in which we store the number of zeros from 0 to each of the elements of the string.
2. `br[i]`, in which we store the number of different subsequences starting at element 0 and having exactly i zeroess.

We loop through the string and simultaneously populate the subs[] array, doing the following:

We check if $k - subs[i] \geq 0$ (ie if 0 to i element has more zeros than searched). If this is held we add to the answer the number of fragments starting from the beginning and having $k - subs[i]$ zeros ($br[sub[i] - k]$).

The complexity is $O(|l|)$.

Problem E. Bugs and Jars

The minimal number of bugs to put in k jars such that there is atleast one bug in each jar and no two jars contain the same number of the bugs, is $k(k+1)/2$. So if $n < k(k+1)/2$, there are not enough bugs and we shall print -1.

Otherwise the answer is $k-1$, if n is divisible by k (we put the bugs in jars in the minimal way 1,2,3, then we put $(n - k(k+1))/k$ bugs in each jar additionally), and k in other cases (we solve the task for $n' = [n/k] * k$ and then add one bug for all jars, starting from the one containing the maximum number of bugs, then the property of the different number of bugs in the jars will still be held).

Problem G. Kangaroo

Its easy to show, that if we are iterating through kangaroos from left to right, and assign the kangaroos to the leftmost possible meal, we can reach optimal distribution. Let we have the distribution where this is not held. We can take the leftmost kangaroo, for which the property is not held, and reassign the meal. Then we have the non-assigned meal moving right. Because we are already processed all kangaroos in the left, the non-assigned meal moves closer to the non-processed kangaroos, so the number of the kangaroos that can get a meal after the reassign is the same or more.

So we need to iterate through kangaroos from left to right and then for each kangaroo find the leftmost possible non-assigned food. It can be easily done by two loops, with the time complexity $O(nk)$.

Problem I. Stations

Let us assume that the sequence is divided by the triples (i, j, k) that satisfy $1 \leq i < j < k < n$. For the partial sum array $S_i = s_1 + \ldots + s_i$, the sum of the first part is $s_1 + \ldots + s_i = S_i$. The sum of the second part is $s_{i+1} + \ldots + s_j = S_j - S_i$, the sum of the third part is $s_{j+1} + \ldots + s_k = S_k - S_j$, and the sum of the fourth part is $s_{k+1} + \ldots + s_n = S_n - S_k$.

The sum of each part must be the same, and since the sum of the entire sequence is S_n, the sum of each part must be $1/4S_n$. The only case that satisfies this is $S_i = 1/4S_n$, $S_j = 2/4S_n$, and $S_k = 3/4S_n$. Therefore, the problem can be solved by counting the number of (i, j, k) pairs that satisfy $S_i = 1/4S_n$, $S_j = 2/4S_n$, and $S_k = 3/4S_n$ with $1 \leq i < j < k < n$.

Let D_i be the number of j that satisfies $S_j = 1/4S_n$. If $D_0 = 0$ and $S_i = 1/4S_n$ for i greater than 1, then $D_i = D_{i-1} + 1$, otherwise $D_i = D_{i-1}$.

Let E_i be the number of j that satisfies $i \leq j < n$ and $S_j = 3/4S_n$. $E_n = 0$, and for i less than or equal to $n-1$, if $S_i = 3/4S_n$, $E_i = E_{i+1} + 1$, otherwise $E_i = E_i + 1$.

For all j such that $2 \leq j < n-1$ and $S_j = 2/4S_n$ and for all i such that $1 \leq i < j$ and $S_i = 1/4S_n$, for all k such that $j < k < n$ and $S_k = 3/4S_n$, the pair (i, j, k) is a correct partition. Since there are $D_j - 1$ such i and E_{j+1} such k, the number of correct divisions can be counted when j is fixed in $O(1)$, thus the problem can be solved in $O(n)$.