Write a program to find, in a given list of naturals, the number of segments whose sum is a given natural.

Let $L$ be the given list, and $n$ be the given natural. The first problem is to say formally “the number of segments in $L$ whose sum is $n$”. Instead of “segments”, we can say “the number of naturals $a$ and $b$ such that $0 \leq a \leq b \leq \#L$ and $\Sigma L [a .. b] = n$”. The quantifier $\Sigma$ turns a predicate into a bunch, and then $\varphi$ tells the size of the bunch, but unfortunately $\Sigma$ works on only one variable, not two. Still, we can sum up the sizes. Formally,

$$\Sigma a \cdot \forall b \cdot 0 \leq a \leq b \leq \#L \land (\Sigma L [a .. b]) = n$$

But that’s ugly. To get a neater, more workable expression, add axioms $\top=1$ and $\bot=0$ equating binary values and numbers. Now the number of segments is

$$\Sigma a, b \cdot 0 \leq a \leq b \leq \#L \land (\Sigma L [a .. b]) = n$$

Suppose the items of $L$ are all 0, and $n=0$. Then there are $(\#L+1) \times (\#L+2)/2$ segments with the right sum, so the best solution is probably quadratic. Let $i$, $j$, $s$, and $c$ be natural variables. The desired result of the computation is $R$, defined as

$$R \equiv c' = \Sigma a, b \cdot 0 \leq a \leq b \leq \#L \land (\Sigma L [a .. b]) = n$$

I will need two more similar specifications $A$ and $B$, defined as

$$A \equiv c' = c + \Sigma a, b \cdot 0 \leq a < b \leq \#L \land (\Sigma L [a .. b]) = n$$

$$B \equiv i = i \land c' = c + \Sigma b \cdot 0 \leq b \leq \#L \land s + (\Sigma L [j .. b]) = n$$

Now the refinements are

$$R \iff i := 0. \ c := 0. \ A$$

$$A \iff j := i. \ s := 0. \ B. \ \text{if} \ i = \#L \ \text{then} \ ok \ \text{else} \ i := i+1. \ A \ \text{fi}$$

$$B \iff \text{if} \ s = n \ \text{then} \ c := c+1 \ \text{else} \ ok. \ \text{fi}$$

$$\text{if} \ j = L \lor s > n \ \text{then} \ ok \ \text{else} \ s := s + L. \ j := j+1. \ B \ \text{fi}$$

We prove the refinement of $R$ by two substitutions. The refinement of $A$ can be proven by cases. First:

$$j := i. \ s := 0. \ B. \ i = \#L \ \land \ ok \ \text{substitutions in} \ B$$

$$\equiv i = i \land c' = c + \Sigma b \cdot 0 \leq i \leq \#L \land (\Sigma L [i .. b]) = n. \ i = \#L \ \land \ ok \ \text{remove sequential composition}$$

$$\equiv i = i \land c' = c + \Sigma b \cdot 0 \leq i \leq \#L \land (\Sigma L [i .. b]) = n$$

Since $i = \#L$, the sum is just the single value when $i = b = \#L$. So it doesn't change anything to put an $a$ in there, $i = a = b = \#L$.

$$\equiv i = i \land c' = c + \Sigma a, b \cdot 0 \leq i \leq \#L \land (\Sigma L [a .. b]) = n$$

$$\Rightarrow A \ \text{The other case is}$$

$$j := i. \ s := 0. \ B. \ i = \#L \land (i := i+1). \ A \ \text{substitutions into} \ B \ \text{and} \ A$$

$$\equiv i = i \land c' = c + \Sigma b \cdot 0 \leq i \leq \#L \land (\Sigma L [i .. b]) = n$$

$$c' = c + \Sigma a, b \cdot 0 \leq i+1 \leq \#L \land (\Sigma L [a .. b]) = n$$

$$\text{remove sequential composition}$$

$$\equiv c' = c + \Sigma b \cdot 0 \leq i \leq \#L \land (\Sigma L [i .. b]) = n$$

$$+ \Sigma a, b \cdot 0 \leq i+1 \leq \#L \land (\Sigma L [a .. b]) = n$$

The first sum looks at all segments starting at $i$.

The second sum looks at all segments starting at or after $i+1$.

Together, they look at all segments starting at or after $i$.

$$\Rightarrow A$$

The refinement of $B$ can be broken into various cases.

$$B \iff s = n \land j = \#L \land (c := c+1)$$

$$B \iff s = n \land j = \#L \land (c := c+1. \ s := s + L. \ j := j+1. \ B)$$

$$B \iff s > n \land ok$$

$$B \iff s < n \land j = \#L \land ok$$
\[ B \Leftarrow s<n \land j<\#L \land (s:= s + L \land j:= j+1. \ B) \]

All five are very easy, so I leave them here. The disjunct \( s\geq n \) is not necessary for correctness. Without it, execution time is exactly \( #L \times (#L+1)/2 \). With it, that’s an upper bound. So for time,

- replace \( R \) with \( t' \leq t + #L \times (#L+1)/2 \)
- replace \( A \) with \( i \leq #L \Rightarrow t' \leq t + (#L-i) \times (#L-i+1)/2 \land i' \leq #L \)
- replace \( B \) with \( j \leq #L \Rightarrow t' \leq t + #L - j \land j' \leq #L \land i'=i \)

Again, easy.

(b) Write a program to find, in a given list of positive naturals, the number of segments whose sum is a given natural.