Problem collection.

Here is a collection of problems you can use to practice.

0.
Given f[0..100) of int, construct a program to determine whether the 2nd half of f is an exact copy of the 1st half of f.

1.
Determine the integer square root of 1023. This is the largest integer i, where i *i is less than or equal to 1023.

2.
Compute the sum of the even elements in g[20..529).

3.
Construct a program to find the largest index i in f[0..1000) where f.i = 2. We guarantee such a value exists.

4.
Determine whether the word "at" is present in the array g[0..100000) of char.

5.
Construct a program to compute the product of the odd elements in f[100..200) of int.

6.
Given f[0..N] where N is at least 1. We know that f.0 is even and f.N is odd. Find an index i where f.i is even and f.(i+1) is odd.

7.
Given f[0..N) and g[0..N) of int, construct a program to determine whether the arrays f and g are exact copies of each other.

8.
Given f[0..10) and g[0..10) of int, construct a program to compute the sum of the products of the values in f and g which have the same index.

9.
Given f[0..20) of char, which represents a word, and g[0..2000) of char which represents a text, construct a program to determine whether f appears at the centre of g.

10.
Given the function f which is defined as follows.

(0) f.0 = 0
(1) f.1 = 1
(2) f.(n+2) = f.n + f.(n+1) + 1 , n greater than or equal to 0.

Construct a program to calculate f.N for some natural N.

11.
Given the function f which is defined as follows.

(0) f.0 = 0
(1) f.1 = 1
(2) f.(n+2) = f.n + f.(n+1) + 4 , n greater than or equal to 0.

Construct a program to calculate f.N for some natural N.

12.
Given X : Int ; N : Nat . Construct a program to establish the following postcondition.

Post: $r = X^N$

13.
Construct a program to count the number of values which are between 10 and 15 inclusive in the array g[40..92) of int.

14.
Construct a program to calculate the product of the natural numbers from 100 to 199 inclusive.

15.
The array f[0..1000) of int is almost in ascending order. This means that f.i is less than or equal to f.(i +1) for most indices i. However, in some places that relationship doesn't hold. Find the leftmost place that it doesn't hold.

16.
Given f[0..100) and g[202..302) both of char, construct a program to determine whether g contains the reverse of f.

17.
Construct a program which determines whether a given natural number N is a prime number or not.

18.
Construct a program to determine the length of the longest segment in f[0..N) of Int, which contains only negative values.

19.
Construct a program to determine the length of the longest segment in f[0..1000) of Int, which contains only zeros.

20.
Construct a program to determine the length of the longest segment in f[0..N) of Int, which is ascending.

21.
Construct a program to determine the length of the longest segment in f[0..N) of Int, which contains at most one negative value.

22.
Construct a program to determine the length of the longest segment in f[0..N) of Int, which contains at most 2 even values.

23.
Given the function f which is defined as follows.

(0) f.0 = 7
(1) f.1 = 9
(2) f.(n+2) = f.n + f.(n+1) + 1 , n greater than or equal to 0.

Construct a program to calculate f.N for some natural N.

24.
Given the function f which is defined as follows.

(0) f.0 = 2
(1) f.1 = 12
(2) f.(2*n) = 2*f.n + f.(n+1) + 6 , 0 < n
(3) f.(2*n+1) = f.n + 7 , 0 < n

Construct a program to calculate f.N for some natural N.

25.
Given f[10..20] of int where f.10 = 6 and f.20 = 21, Construct a program to find an index i in f where even.(f.i) and odd.(f.(i+1))

26.
Given the following definition of a function f, construct an algorithm to compute f.N for some natural number N.

f.0 = 0
f.1 = 1
f.(2*n) = f.n + f.(n+1) , 0 < n
f.(2n + 1) = f.n + 13*f.(n+1) + 5 , 0 < n

27.
Given f[0..N] of Integer, where 0 < N. Please construct an efficient program to fulfil the following specification.

Pre: { prime.(f.0) $\wedge$ not.prime.(f.N) }

Post: { prime.(f.i) $\wedge$ not.prime.(f.(i+1)) }

28.
Given f[0..N] of Boolean, where 0 < N. Please construct an efficient program to fulfil the following specification.

Pre: { f.0 $\wedge$ not.(f.N) }

Post: { f.i $\wedge$ not.(f.(i+1)) }