

1. Program analysis

Step	Description	AC	a0	a1	a2
1	Loads a0 into AC register				
2	Compliments AC				
3	Loads AC into memory				
4	Increment a0 and skip if zero				
5	Branch Unconditionally to address 20				
6	Load the value pointed by a1 into AC				
7	Add the value of a1 to the value of a2				
8	Load AC into the location pointed to by a3				
9	Load a1 into AC				
10	Increment AC by 1				
11	Store AC into a1				
12	Increment AC by 1				
13	Store AC into a2				
14	Increment AC by 1				
15	Store AC into a3				
16	Branch Unconditionally to step 4				

What this program does is builds a list of fibonacci sequence providing that the initial values in a0 and a1 are [1,1] or [0,1]

2.

```
for(int n = 0; n < Counter; n++){  
    Value at Z <- value at X + value at Y  
    X <- X + 1  
    Y <- Y + 1  
    Z <- Z + 1  
}
```

3. This program calculates values of fib

4. Its practical to use indirect addressing because it cycles through a list of values that get generated and pointer arithmetic is easy for list generation.