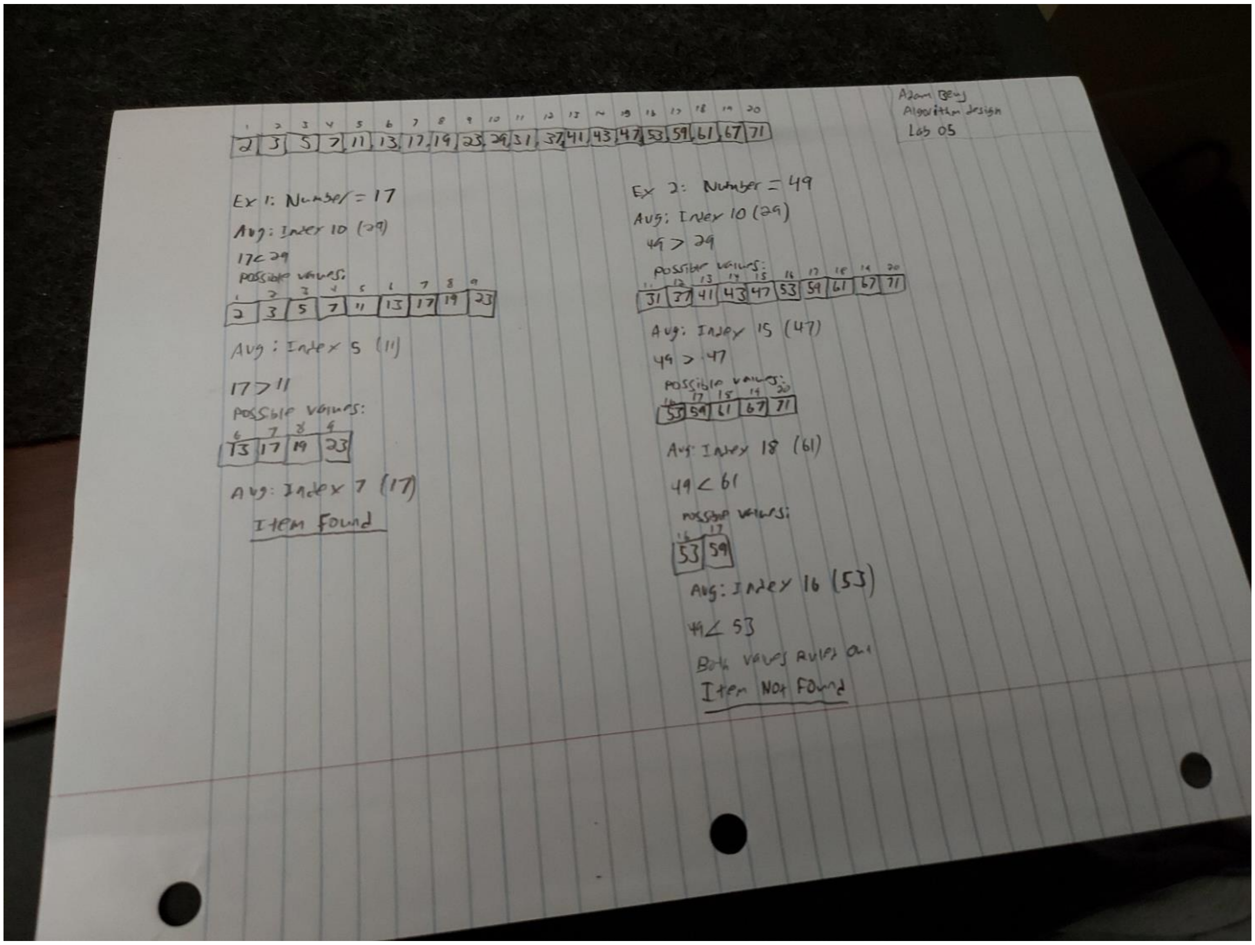
Step 1 (by hand)



Adam Berg
Algorithm design
Lab 05

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
2	3	5	7	11	13	17	19	23	29	31	37	41	43	47	53	59	61	67	71

Ex 1: Number = 17

Avg: Index 10 (29)

17 < 29

possible values:

1	2	3	4	5	6	7	8	9
2	3	5	7	11	13	17	19	23

Avg: Index 5 (11)

17 > 11

possible values:

6	7	8	9
13	17	19	23

Avg: Index 7 (17)

Item Found

Ex 2: Number = 49

Avg: Index 10 (29)

49 > 29

possible values:

11	12	13	14	15	16	17	18	19	20
31	37	41	43	47	53	59	61	67	71

Avg: Index 15 (47)

49 > 47

possible values:

16	17	18	19	20
53	59	61	67	71

Avg: Index 18 (61)

49 < 61

possible values:

16	17
53	59

Avg: Index 16 (53)

49 < 53

Both values ruled out

Item Not Found

Step 2 (approach)

In this pseudocode, I will attempt to find a given number out of a list of the first 20 prime numbers. I will first divide the list in half by finding the average index, then comparing that number with the given number. If the given number is bigger than that of the average index, then I will know to rule out the average index and all numbers below it. I will then repeat this process, continuing to divide the list in half until I find the correct number in the list.

Step 3 (my pseudocode)

```
Start = 0

End = length(list)
```

```
While start <= end
    Mid = (start + end) MOD 2
    If list[mid] = target_number
        RETURN true
    Elseif list[mid] < target_number
        Start = mid + 1
    Else
        End = mid – 1
Return False
```

<u>Step 4 (Copilot AI generated pseudocode)</u>

```
Initialize prime_numbers as [2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71].
Input target_number.
Set left = 0 and right = length(prime_numbers) - 1.
While left <= right:
  Calculate mid = (left + right) // 2.
  If prime_numbers[mid] == target_number:
    - Output "Number found at index " + mid.
    - Exit loop.
  Else if prime_numbers[mid] < target_number:
    - Set left = mid + 1.
  Else:
    - Set right = mid - 1.
If target_number not found:
  - Output "Number is not in the list."
```

Step 5 (analysis)

Both solutions will likely work well, and there are only minor differences. The AI solution was better at being specific and using actual variable names, and it listed the prime numbers at the top, whereas mine did not. I used MOD instead of //, and upon asking AI about that, it said both are acceptable. Overall, mine was shorter and simpler, but it also did not have all of the information that may be needed to write the code. My solution can be improved by adding this information to the pseudocode so it will be easier to translate. I also need to subtract 1 to the end to account for how lists work in programming languages (starting at 0). Copilots solution can be improved by returning instead of exiting, and being less wordy, allowing the programmer to give his or her own messages.

Step 6 (updated pseudocode)

```
Initialize list as [2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71].

Start = 0

End = length(list) - 1

While start <= end

    Mid = (start + end) MOD 2

    If list[mid] = target_number

        RETURN target found

    Elseif list[mid] < target_number

        Start = mid + 1

    Else

        End = mid – 1

Return target not found
```