

Binary Search Algorithm – part 1

1. The Python code is given for the *binary search* algorithm. Convert the code to correct Pearson pseudocode, then draw the flowchart diagram for the algorithm. Use the back side of the page for more space. Recall that the `//` operator in Python is integer division.

```

1 def binary_search(arr, target)
2     low = 0
3     high = len(arr) - 1
4     while low <= high:
5         mid = (low + high) // 2
6         if arr[mid] == target:
7             return mid
8         if arr[mid] < target:
9             low = mid + 1
10        else:
11            high = mid - 1
12    return -1

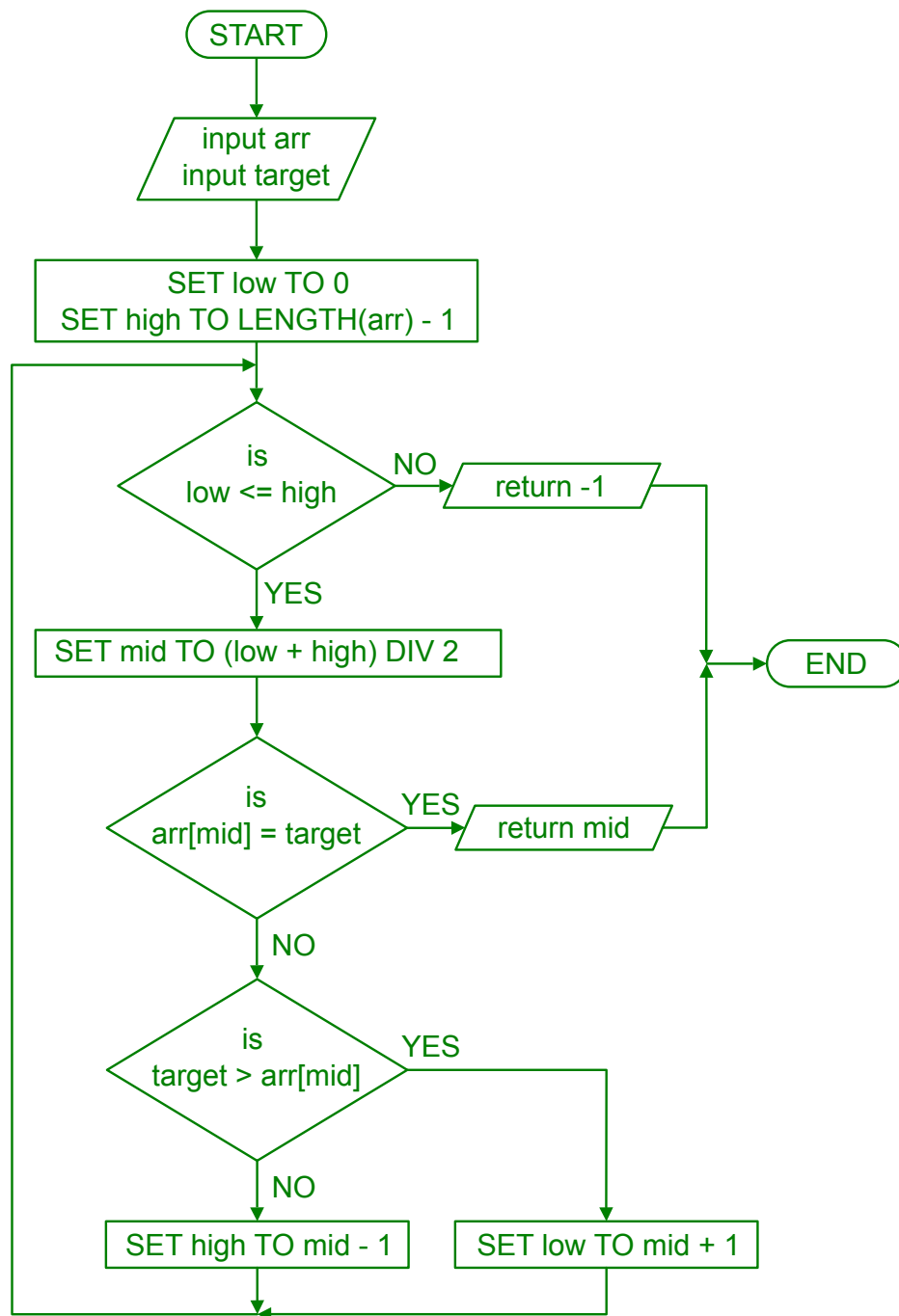
```

```

FUNCTION binary_search(arr, target)
BEGIN FUNCTION
SET low TO 0
SET high TO LENGTH(arr) - 1
WHILE low <= high DO
    SET mid TO (low + high) DIV 2
    IF arr[mid] = target THEN
        RETURN mid
    END IF
    IF arr[mid] < target THEN
        SET low TO mid + 1
    ELSE
        SET high TO mid - 1
    END IF
END WHILE
RETURN -1
END FUNCTION

```

- Ensure each student remembers:
 - neatness and proper indentation
 - BEGIN FUNCTION and matching END FUNCTION
 - DO for WHILE . . DO statements
 - THEN for IF . . THEN statements
 - END WHILE
 - END IF (two of them!)
 - SET *variable TO expression*, not just *variable = expression*
 - Pearson pseudocode for integer division is DIV
- Flowchart diagram on page 2.

Binary Search Algorithm – part 1

- The correct flowchart symbols must be used
- The contents of any flowchart block may be any understandable code: Pearson pseudocode, Python, Java, or just any pseudocode – as long as the intent is clear and unambiguous.
- There must be one and only one START block and one and only one END block
- Arrow heads with the correct direction of flow are required on every connection