Name: Vince Allen D. Cristal	**Date:** February 4, 2026
School: LSPU - San Pablo City Campus	**Course:** On-The-Job Training

1. Inventory Tracker

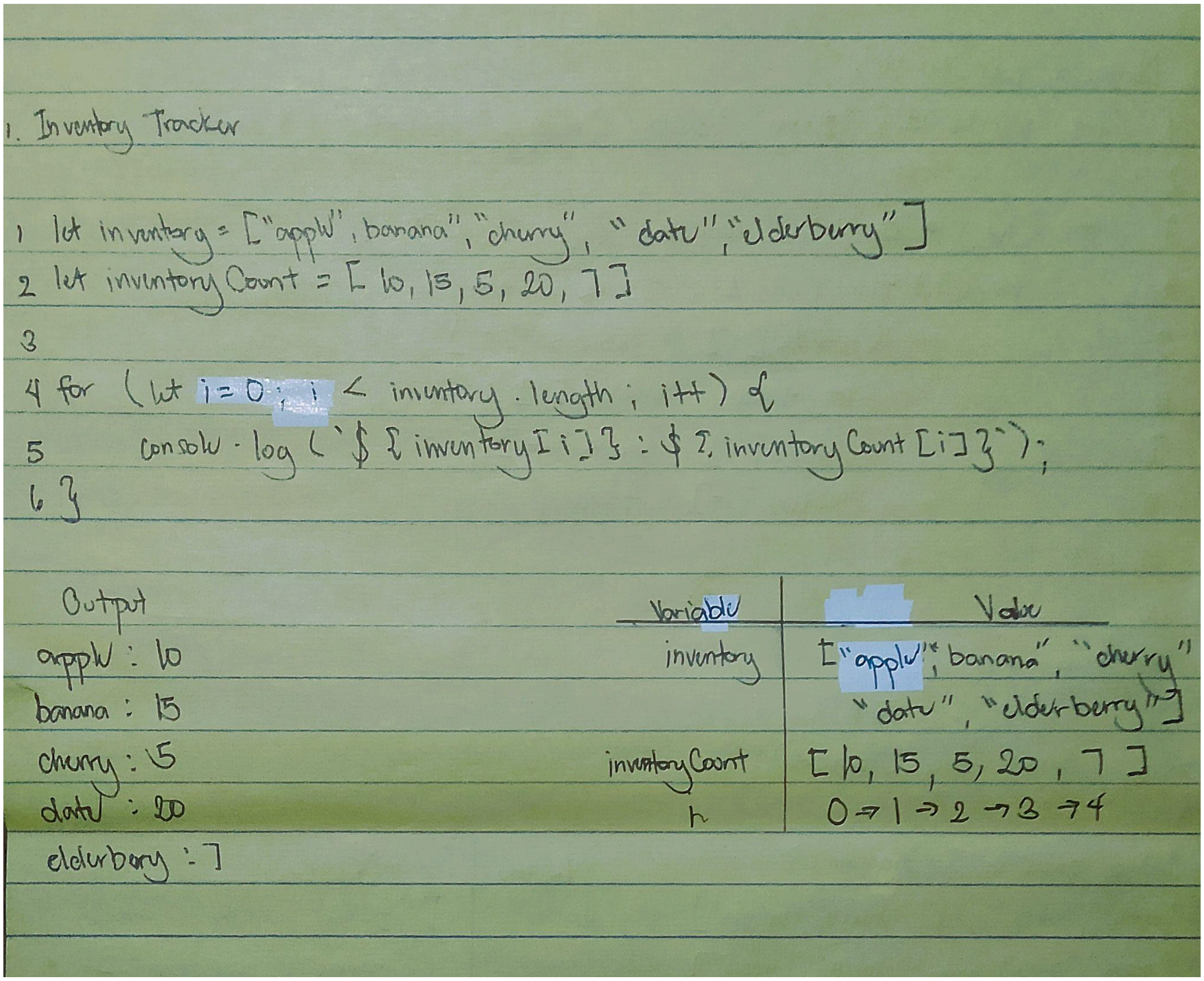
Code:

```
let inventory = ["apple", "banana", "cherry", "date", "elderberry"];
let inventoryCount = [10, 15, 5, 20, 7];

for (let i = 0; i < inventory.length; i++) {
        console.log(`${inventory[i]}: ${inventoryCount[i]}`);
}
```

Output:

```
apple: 10
banana: 15
cherry: 5
date: 20
elderberry: 7
```

Screenshot of T-diagram:



1. Inventory Tracker

```
let inventory = ["apple", banana", "cherry", "date", "elderberry"]
let inventoryCount = [ 10, 15, 5, 20, 7 ]

for (let i = 0; i < inventory.length; i++) {
    console.log(`${inventory[i]}: ${inventoryCount[i]}`);
}
```

Output

```
apple : 10
banana : 15
cherry : 5
date : 20
elderbery : 7
```

Variable	Value
inventory	["apple", banana", "cherry" "date", "elderberry"]
inventoryCount	[10, 15, 5, 20, 7]
h	0 → 1 → 2 → 3 → 4

2. Alphabetical Sorting

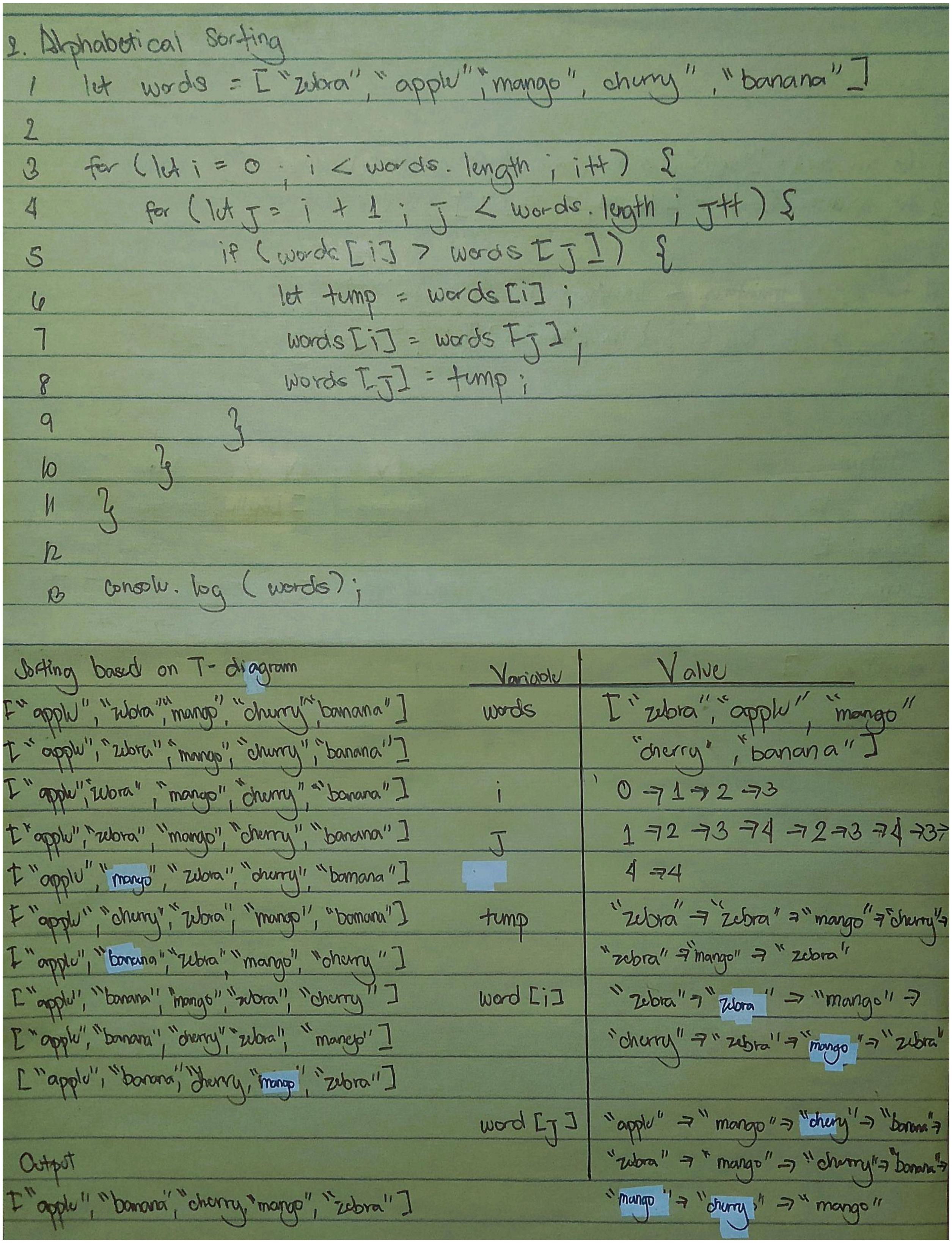
Code:

```
let words = ["zebra", "apple", "mango", "cherry", "banana"];

for (let i = 0; i < words.length; i++) {
  for (let j = i + 1; j < words.length; j++) {
    if (words[i] > words[j]) {
      let temp = words[i];
      words[i] = words[j];
      words[j] = temp;
    }
  }
}

console.log(words);
```

Output:

```
["apple", "banana", "cherry", "mango", "zebra"]
```

Screenshot of T-diagram:



2. Alphabetical Sorting

```
let words = ["zebra", "apple", "mango", cherry", "banana"]

for (let i = 0 ; i < words.length ; i++) {
    for (let j = i + 1 ; j < words.length ; j++) {
        if (words[i] > words[j]) {
            let temp = words[i] ;
            words[i] = words[j] ;
            words[j] = temp ;
        }
    }
}

console.log (words);
```

Sorting based on T-diagram

["apple", "zebra", "mango", "cherry", "banana"]
["apple", "zebra", "mango", "cherry", "banana"]
["apple", "zebra", "mango", "cherry", "banana"]
["apple", "zebra", "mango", "cherry", "banana"]
["apple", "mango", "zebra", "cherry", "banana"]
["apple", "cherry", "zebra", "mango", "banana"]
["apple", "banana", "zebra", "mango", "cherry"]
["apple", "banana", "mango", "zebra", "cherry"]
["apple", "banana", "cherry", "zebra", "mango"]
["apple", "banana, cherry, "mango", "zebra"]

Variable	Value
words	["zebra", "apple", "mango" "cherry", "banana"]
i	0 → 1 → 2 → 3
j	1 → 2 → 3 → 4 → 2 → 3 → 4 → 3 → 4 → 4
temp	"zebra" → "zebra" → "mango" → "cherry" → "zebra" → "mango" → "zebra"
word [i]	"zebra" → "zebra" → "mango" → "cherry" → "zebra" → "mango" → "zebra"
word [j]	"apple" → "mango" → "chery" → "banana" → "zebra" → "mango" → "cherry" → "banana" → "mango" → "cherry" → "mango"

Output

["apple", "banana", "cherry, "mango", "zebra"]

3. Unique Array Builder

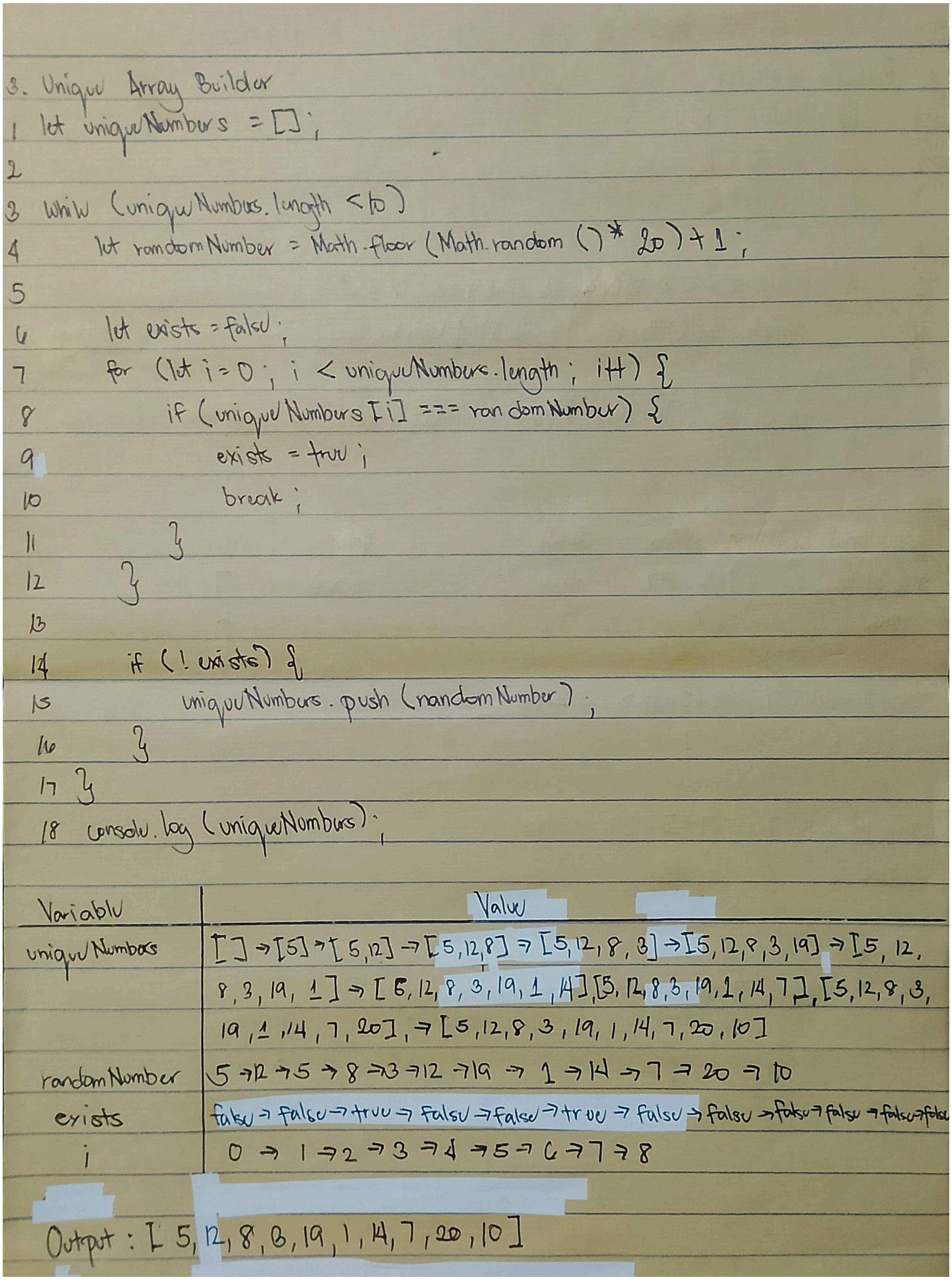
Code:

```
let uniqueNumbers = [];

while (uniqueNumbers.length < 10) {
  let randomNumber = Math.floor(Math.random() * 20) + 1;

  // Check if the randomNumber already exists in the array
  let exists = false;
  for (let i = 0; i < uniqueNumbers.length; i++) {
    if (uniqueNumbers[i] === randomNumber) {
      exists = true;
      break;
    }
  }

  // If it doesn't exist, add it to the array
  if (!exists) {
    uniqueNumbers.push(randomNumber);
  }
}

console.log(uniqueNumbers);
```

Output:

```
[5, 12, 8, 3, 19, 1, 14, 7, 20, 10]
```

Screenshot of T-diagram:



3. Unique Array Builder

```
let uniqueNumbers = [];

while (uniqueNumbers.length < 10)
      let randomNumber = Math.floor (Math.random ()* 20) + 1;

      let exists = false;
      for (let i = 0; i < uniqueNumbers.length; i++) {
            if (uniqueNumbers [i] === randomNumber) {
                  exists = true;
                  break;
            }
      }

      if (! exists) {
            uniqueNumbers.push (randomNumber);
      }
}
console.log (uniqueNumbers);
```

Variable	Value
uniqueNumbers	[ ] → [5] → [5, 12] → [5, 12, 8] → [5, 12, 8, 3] → [5, 12, 8, 3, 19] → [5, 12, 8, 3, 19, 1] → [5, 12, 8, 3, 19, 1, 14], [5, 12, 8, 3, 19, 1, 14, 7], [5, 12, 8, 3, 19, 1, 14, 7, 20], → [5, 12, 8, 3, 19, 1, 14, 7, 20, 10]
randomNumber	5 → 12 → 5 → 8 → 3 → 12 → 19 → 1 → 14 → 7 → 20 → 10
exists	false → false → true → false → false → true → false → false → false → false → false → fals
i	0 → 1 → 2 → 3 → 4 → 5 → 6 → 7 → 8

Output : [5, 12, 8, 3, 19, 1, 14, 7, 20, 10]

4. Triangle Checker

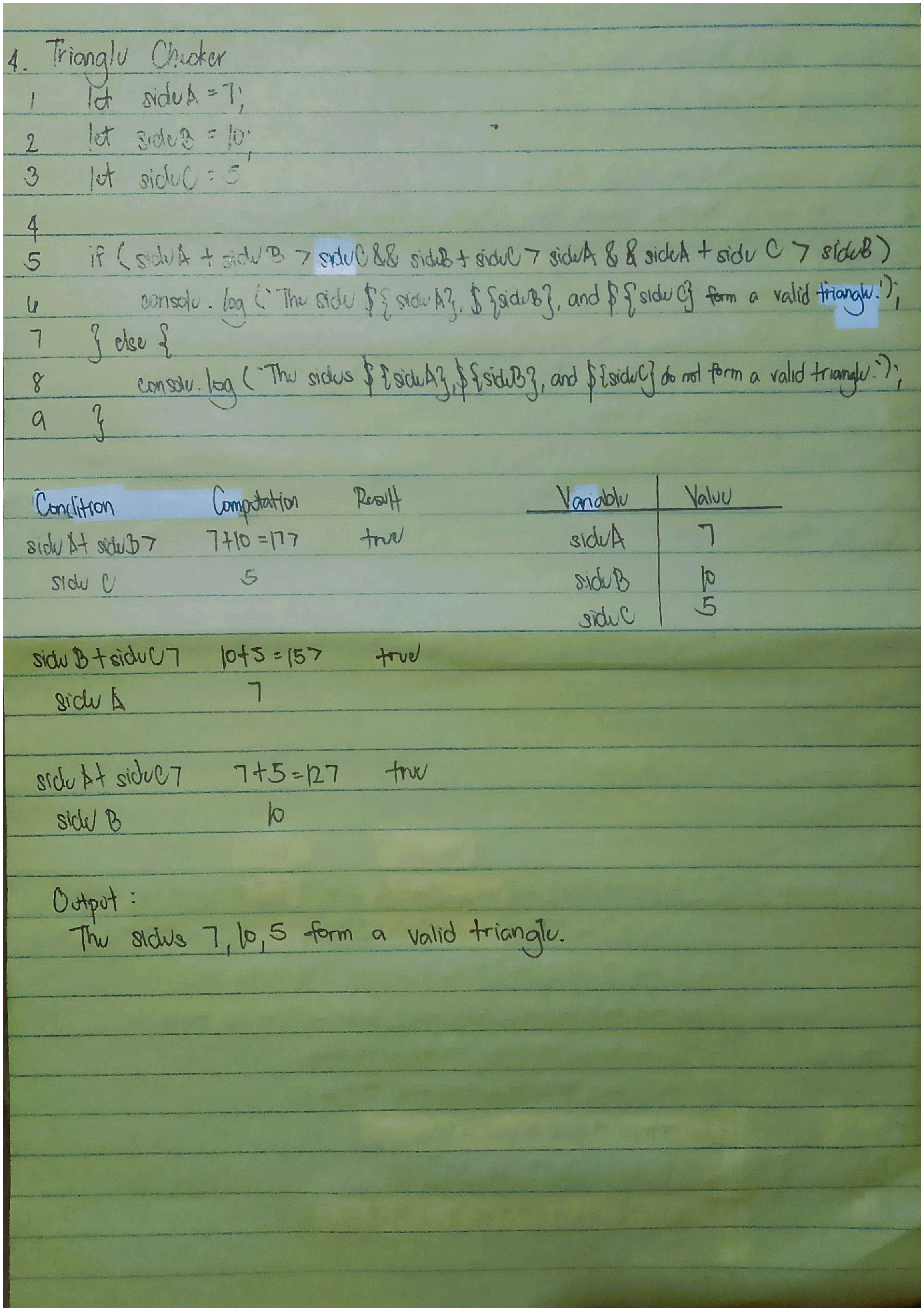
Code:

```
let sideA = 7;
let sideB = 10;
let sideC = 5;

if (sideA + sideB > sideC && sideB + sideC > sideA && sideA + sideC > sideB) {
  console.log(`The sides ${sideA}, ${sideB}, and ${sideC} form a valid triangle.`);
} else {
  console.log(`The sides ${sideA}, ${sideB}, and ${sideC} do not form a valid triangle.`);
}
```

Output:

```
The sides 7, 10, and 5 form a valid triangle.
```

Screenshot of T-diagram:



4. Triangle Checker

```
let sideA = 7;
let sideB = 10;
let sideC = 5

if (sideA + sideB > sideC && sideB + sideC > sideA && sideA + side C > sideB)
      console.log(`The side ${sideA}, ${sideB}, and ${sideC} form a valid triangle.`);
} else {
      console.log(`The sides ${sideA}, ${sideB}, and ${sideC} do not form a valid triangle.`);
}
```

Condition	Computation	Result
sideA + sideB > side C	7+10 = 17 > 5	true
sideB + sideC > side A	10+5 = 15 > 7	true
sideA + sideC > side B	7+5 = 12 > 10	true

Variable	Value
sideA	7
sideB	10
sideC	5

Output:

The sides 7, 10, 5 form a valid triangle.