

Check if the array is *STRICTLY* increasing or *STRICTLY* decreasing or non-monotonic check for monotonicity in 1 pass through the array



example

1 5 17 19 23

answer "increasing"

56 52 37 19 3

answer "decreasing"

1 5 67 19 23

answer "non-monotonic"

```
void searchArray(int arr[], int length) {
    int arrayCheck = 0; // define char for end cout whether incr, decr, or
    neither
    for (int i = 0; i < length - 1; i++) {
        if (arr[i + 1] < arr[i]) { // first check if decreasing
            arrayCheck = 2;
        }
        if (arrayCheck == 2) {
            if (arr[i + 1] == arr[i]) { // catch if changes
                arrayCheck = 3;
                break;
            }
            else if (arr[i + 1] > arr[i]) { // catch if changes
                arrayCheck = 3;
                break;
            }
        }
        if (arr[i + 1] > arr[i]) { // check if increasing
            arrayCheck = 1;
        }
        if (arrayCheck == 1) {
            if (arr[i + 1] == arr[i]) { // catch if changes
                arrayCheck = 3;
                break;
            }
            else if (arr[i + 1] < arr[i]) { // catch if changes
                arrayCheck = 3;
                break;
            }
        }
        if (arr[i + 1] == arr[i]) { // else just monotonic if first two elements
            arrayCheck = 3;
            break;
        }
    }
    if (arrayCheck == 1) {
        std::cout << "increasing" << std::endl;
    }
    if (arrayCheck == 2) {
        std::cout << "decreasing" << std::endl;
    }
    if (arrayCheck == 3) {
        std::cout << "non-monotonic" << std::endl;
    }
}
```

```
void searchArray(int arr[], int length) {
    int counter = 0;
    for (int i = 0; i < length - 1; i++) {
        if (arr[i + 1] < arr[i]) { // first check if decreasing
            counter--;
        }
        if (arr[i + 1] > arr[i]) { // check if increasing
            counter++;
        }
    }
    if (counter == length - 1) {
        std::cout << "increasing" << std::endl;
    }
    else if (counter == -(length - 1)) {
        std::cout << "decreasing" << std::endl;
    }
    else {
        std::cout << "non-monotonic" << std::endl;
    }
}
```

//inefficient for finding nonmonotony because has to check all of the elements to the end