

Find the length of the longest constant section of an array

example

1 4 4 4 8 8 6 7 9 9 9 9 0

1 4 4 4 8 8 6 7 9 9 9 9 9

93210576412

length-1 length

HW: there is an error, hint: changed length -

1,

answer 4

```
void searchForGroup(int arr[], int length) {
    int foundLength = 0;
    int currentLength = 1;
    for (int i = 0; i < length - 1; i++) {
        if (arr[i] == arr[i + 1]) { //checking if two numbers next to each other are the same
            currentLength++;
        }
        else {
            if (foundLength < currentLength) {
                foundLength = currentLength;
            }
            currentLength = 1; // reset for next possible length of numbers
        }
    }
    std::cout << "Longest string of numbers is: " << foundLength << std::endl;
}
```



```
void searchForGroup(int arr[], int length) {
    int foundLength = 0; // final length of string of numbers
    int currentLength = 1; // current length of string of numbers
    for (int i = 0; i < length - 1; i++) {
        if (arr[i] == arr[i + 1]) { //checking if two numbers next to each other are the same
            currentLength++;
        }
        else {
            if (foundLength < currentLength) {
                foundLength = currentLength;
            }
            currentLength = 1; // reset for next possible length of numbers
        }
    }
    if (foundLength < currentLength) {
        foundLength = currentLength;
    } // added second time to catch string at the end because it has no number to compare last
    std::cout << "Longest string of numbers is: " << foundLength << std::endl;
}
// error is that it doesn't count longest string of numbers if its at the end
```