Find the length of the largest strictly monotonic segment of an array (increasing or decreasing)

example

1 2 5 7 6 6 6 3 2 10 11 12 16 29 29

1 2 5 7 - 4
7 6 - 2
6 3 2 - 3
- 7
2 10 11 12 16 29 - 6

1 2 5 7 - 4
7 6 6 6 3 2 - 6
2 10 11 12 16 29 29

answer 6

```
void searchForGroup(int arr[], int length) {
 int foundLength = 0;
 int currentLength = 1;
 int otherLength = 1;
 int flag = 0;            // not encountered increasing/decreasing
     for (int i = 0; i < length - 1; i++) {
           if (arr[i] < arr[i + 1]) {       // encountered increase
              if (flag == -1) {              // just finished neg string, started new increase
                 if (foundLength < otherLength) { //resets new counter for increase
                    foundLength = otherLength;
                 }
                 currentLength = 1;
              }
              currentLength++;
              flag = 1;
           }
           if (arr[i] > arr[i + 1]) {             // encountered decrease
              if (flag == 1) {                     // previous numbers higher, new decrease encountered
                 if (foundLength < currentLength) {    // reset counter for new decrease
                    foundLength = currentLength;
                 }
                 otherLength = 1;
              }
              otherLength++;
              flag = -1;
           }
           if (arr[i] == arr[i+1]) {     // Extra error catch for "strict" string where if numbers are increasing or decreasing before two #s that are the same, it resets string count
              if ( flag == 1) {
                 if (foundLength < currentLength) {    // reset counter for new decrease
                       foundLength = currentLength;
                 }
                 currentLength = 1;
              }
              if (flag == -1) {
                 if (foundLength < otherLength) {
                    foundLength = otherLength;
                 }
                 otherLength = 1;
              }
           }
           // std::cout << "current: " << currentLength << " other: " << otherLength << " found: " << foundLength << " arr[i]: " << arr[i] << " arr[i + 1]: " << arr[i+1] << std::endl;
     }
     if (foundLength < currentLength) {
        foundLength = currentLength;
     }
     else if (foundLength < otherLength) {
        foundLength = otherLength;
     }
     std::cout << "current: " << currentLength << " other: " << otherLength << " found: " << foundLength << std::endl;
    std::cout << "String of numbers: " << foundLength << std::endl;
}

}
```