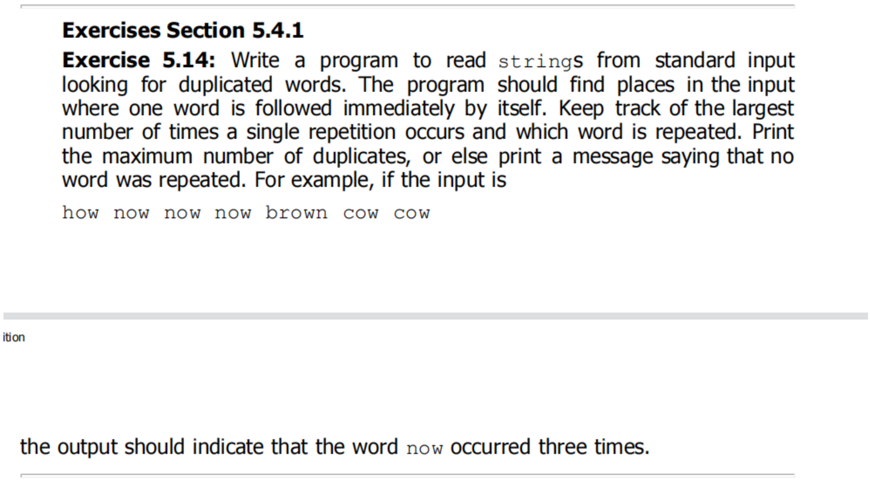
【C++】Day14

Class	C++
Date	@December 2, 2021
Material	
Series Number	
Summary	

【Ch5】Iterative Statements

5.4 Iteratitve Statements

5.4.2 Traditional for Statement



Exercises Section 5.4.1

Exercise 5.14: Write a program to read `string`s from standard input looking for duplicated words. The program should find places in the input where one word is followed immediately by itself. Keep track of the largest number of times a single repetition occurs and which word is repeated. Print the maximum number of duplicates, or else print a message saying that no word was repeated. For example, if the input is

```
how now now now brown cow cow
```

ition

the output should indicate that the word `now` occurred three times.

```
string str;
string prevStr;
cin >> prevStr;
int maxCount = 1;
int count = 0;
string maxStr;
maxStr = prevStr;
while(cin >> str) {
  if(str == prevStr) {
    ++count;
    if(count > maxCount) {
      maxCount = count;
    maxStr = str;
    }
  } else
    count = 1;
  prevStr = str;
}
if(maxCount > 1)
  cout << maxStr << " occurs " << maxCount;
```

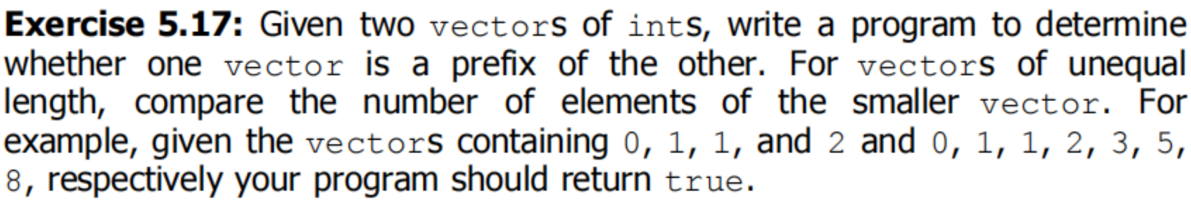
Multiple Definitions in the for Header

As in any other declaration, init-statement can define several objects. However, init-statement may be only a single declaration statement. Therefore, all the varaibles must have the same base type.

As one example, we might write a loop to duplicate the elements of a vector on the end as follows:

```
//remember the size of v and stop when we get to the original last element
for(decltype(v.size()) i = 0, sz = v.size(); i != sz; ++i)
  v.push_back(v[i]);
```

In this loop we define both the index, i, and the loop control, sz, in init-statement



Exercise 5.17: Given two `vector`s of `int`s, write a program to determine whether one `vector` is a prefix of the other. For `vector`s of unequal length, compare the number of elements of the smaller `vector`. For example, given the `vector`s containing `0`, `1`, `1`, and `2` and `0`, `1`, `1`, `2`, `3`, `5`, `8`, respectively your program should return `true`.

```
vector<int> prefix(3, 0);
  vector<int> vec = {0, 0, 0, -1};
  int tempInt;

  unsigned int prefSize = prefix.size();
  unsigned int vecSize = vec.size();
  if(vecSize >= prefSize) {
    int i = 0;
    while(i < prefSize) {
      if(prefix[i] != vec[i]) {
        cout << "Not a prefix" << endl;
        return 0;
      }
      ++i;
    }
  }
  cout << "Is prefix";
```