1)

```
#include <queue>
#include <iostream>
using namespace std;

int size(queue<int> q) {
    //function returns size but does not modify
    //elements.
    return q.size();
}

int main() {
    queue<int> aQueue;
    int x = 0;

    cout << "Enter queue elements (int). Type -1 to quit\n";

    while (x != -1) {
        cin >> x;
        if (x != -1) {
            aQueue.push(x);
        }
    }

    int sizeQ = aQueue.size();
    cout << sizeQ;
    return 0;
}
```

2)

```
#include <queue>
#include <iostream>
#include <string>
using namespace std;

queue<string> findLarger(queue<string> inputq, int max) {
	queue<string> q;

	while (!inputq.empty()) {
		string element = inputq.front();

		if (element.length() > max) {
			q.push(element);
		}
		inputq.pop();
	}
	return q;
}

int main() {
	const int max = 10;
	queue<string> aQueue;
	aQueue.push("short");
	aQueue.push("longstring1");
	aQueue.push("short2");
```

```
	aQueue.push("longstring2");
	queue<string> aCopy;
	aCopy = aQueue;
	//outputs queue
	cout << endl;
	while (!aCopy.empty()) {
		string el = aCopy.front();
		cout << aCopy.front() << " ";
		aCopy.pop();
	}
	cout << endl;
	//modifies queue by implementing findLArger
	aQueue = findLarger(aQueue, max);
	//Ouputs new queue
	queue<string> anotherCopy;
	anotherCopy = aQueue;
	while (!anotherCopy.empty()) {
		string el = anotherCopy.front();
		cout << anotherCopy.front() << " ";
		anotherCopy.pop();
	}

	return 0;

}
```

3)

```
#include <iostream>
#include <string>
#include <stack>

using namespace std;



stack<string> reverse(string str) {
	stack<string> aStack;
	//stack<string> bStack;
	string word;
	//string sentence;
	bool start = false;
	bool end = false;
	for (int i = 0; i < str.size(); i++) {
		if (str.at(i) != '.'&& str.at(i) != ' ') {
			word.push_back(str.at(i));
		}
		else if (str.at(i) == ' ') {
			aStack.push(word);
			word.clear();
		}
		if (isupper(str.at(i))) {
			str.at(i) = tolower(str.at(i));
			start = true;
		}
		if (str.at(i) == '.') {
			end = true;
```

```
			}
			if (start && end) {
				word.push_back('.');
				start = false;
				end = false;
			}
		}
		return aStack;
}


int main() {
		string str = "You have plenty of courage, I am sure. All you need is
confidence in yourself.";
		stack<string> newSentence = reverse(str);
		while (!newSentence.empty()) {
			cout << newSentence.top() << " ";
			newSentence.pop();
		}

		return 0;
}
```