

CS 241 - Week 3 Tutorial

Assembly Language Programming and C++ Review

Fall 2018

1 Assembly Language Programming

Problem 1 - I/O and loops in MIPS

Adapt your solution to problem 4 from last tutorial to read in characters from stdin until EOF is encountered and print out their uppercase versions to stdout. If the character is not a lower-case letter, simply print out the character unchanged.

Problem 2 - Using the stack in MIPS

Write a MIPS program which reads in characters from stdin until EOF is encountered, then prints the same characters out backwards to stdout. Use the stack to store the characters.

Problem 3 - Functions and recursion in MIPS

Rewrite your solution to Problem 1 from last tutorial using a recursive function instead of a loop.

2 C++ Review

You can use several different languages in CS241 for assignments, namely C++, Racket, and Scala. For the most part we'll avoid talking about specific languages whenever possible in this course, but since a little more than half of you typically elect to do the assignments in C++, here's a bit of C++ review.

2.1 Code Style

There are a number of flaws in the following code snippet. How can this piece of code be improved? Think about both stylistic improvements and performance improvements.

```

#include <iostream>
#include <vector>
#include <set>
#include <map>
#include <string>
#include <algorithm>
using namespace std;

bool foo(vector<string> v) {
    if (v.size() > 16) {
        return true;
    } else {
        return false;
    }
}

int bar(map<string, map<vector<string>, int> > m, string w) {
    return m[w].size();
}

bool baz(string fruit) {
    return fruit == "apple" || fruit == "pear" || fruit == "mango" ||
           fruit == "coconut" || fruit == "kiwi" || fruit == "pepper"
}

string temp;
bool qux(pair<vector<string>, int> p) {
    int count = 0;
    for (int i = 0; i < p.second; ++i) {
        if (p.first[i] == temp) count++;
    }
    if (count > p.second/2) {
        return true;
    } else {
        return false;
    }
}

int main() {
    vector<string> fruits;
    map<string, map<vector<string>, int> > fruitMap;
}

```

```

while (true) {
    string fruit;
    cin >> fruit;

    fruitMap[fruit][fruits] = fruits.size();

    int mode;
    if (fruit == "apple") {
        mode = 0;
    } else if (fruit == "banana") {
        mode = 1;
    } else if (fruit == "tangelo") {
        mode = 2;
    } else {
        throw 143;
    }

    fruits.push_back(fruit);
if (foo(fruits)) {
    cout << "Many_fruits" << endl;
}

int val = bar(fruitMap, fruit);

    bool flag1 = false, flag2 = false;
    if (val > 12345) {
        flag1 = true;
    } else if (fruits.size() > 8 && fruits.size() < 12) {
        flag1 = true;
    } else if (mode == 1) {
        flag2 = true;
    }

    if (flag2 || flag1 && baz(fruit)) {
        break;
    }
}

for (map<string, map<vector<string>, int> >::iterator it = fruitMap.b
    it != fruitMap.end(); ++it) {
    temp = (*it).first;
}

```

```
        cout << count_if((*it).second.begin(), (*it).second.end())  
    }  
}
```