Wenlan Tian

COP 3530

9/16/2014

Exercise2

Section 1087, MAEB211

Part I. Street parade

#include <string>

#include <iostream>

#include <fstream>

#include <vector>

using namespace std;

vector <int> temp;

int n = 0;

int arr[0];

void input(string fileName){

cin >> fileName;

ifstream inStream;

string tempString;

int m = 1;

int i = 0;

inStream.open(fileName.c\_str());

if(inStream.fail()){

cout << fileName << "cannot be opened" << endl;

return;

}

inStream >> n;

while (inStream >> m && m!=0){

arr[i]=m;

i++;

}

}

//check if current array element is the minimun from the current to the last

bool min(int x){

for (int i = x+1; i < n; i++){

if (arr[x] > arr[i])

return false;

}

return true;

}

//compare the order of the vector

//if it is ordered from max to min, then yes; otherwise, no

bool output(){

cout << "output: " << flush;

for (int i = 0; i < temp.size();i++){

if (temp[i] < temp[i+1]){

cout << "no" << endl;

return false;

}

}

cout << "yes" << endl;

return true;

}

int main(){

//push the input index of love mobiles into an array

string inFileName;

input(inFileName);

//check each element, if it is not the minimun in the array, push it onto a vector

for (int i =0; i < n; i++){

if(!min(i))

temp.push\_back(arr[i]);

}

//compare the order of the vector

output();

return 0;

}

Part II. Write neat pseudo code for an algorithm, which will evaluate an arbitrary prefix expression (+-&\*/)

1. change prefix to postfix expression by reversing the order of all the values and operators
2. simply implementing using a stack for postfix expression that (1) every time, when encounter a value, push the value to the stack; (2) every time, when encounter a operator, pop the stack twice, and perform the operation using the values as the operands, and push the result back on to the stack
3. return the top of the stack which is the result for the original prefix expression.