/*
* uninitialized.cpp
*
* @author Cynthia Lee
* @version 1920-Aut
* Just prints out some uninitialized memory. It's like opening a present, you
* never know what you're going to get! #Surprise #NotAVeryGoodPresentThough
*/

#include <iostream>
#include "console.h"
#include "simpio.h"
using namespace std;

const int ARRAY_SIZE = 9;
const double B_GRADE = 0.8;

int guessContentsOfStackVariable();
int guessContentsOfHeapArray();
void checkGuess(int guess, int answer);

/*
* Let's print some uninitialized memory!
* Professional coder on a closed course. Do not try this at home.
*/
int main() {
    cout << "Guess numbers between -2.147 billion and +2.147 billion (approx)";
    cout << endl << endl;
    int nCorrect = 0;
    nCorrect += guessContentsOfStackVariable();
    nCorrect += guessContentsOfHeapArray();
    if (((double)nCorrect)/(ARRAY_SIZE + 1) >= B_GRADE) {
        cout << "Wow!!! Great guessing!!" << endl;
    } else {
        cout << "You made " << nCorrect << " correct guesses. " << "Better luck next time." << endl;
    }
    return 0;
}

/*
* Quick demo to show that STACK memory can be uninitialized and if we print
* the contents of a local variable we see a "random"/garbage value.
* @return 1 if the user correctly guessed the uninitialized int value, else 0.
*/
int guessContentsOfStackVariable() {
    int x;
    int guess = getInteger("Guess the value of x: ");
    if (x == guess) {
        cout << "You guessed correctly!" << endl << endl;
        return 1;
    } else {
        cout << "Sorry, the correct answer was " << x << endl << endl;
        return 0;
    }
}

/*
* Quick demo to show that HEAP memory can be uninitialized and if we print
* the contents of a heap array of type int we see "random"/garbage values.
* @return the number of correctly guessed uninitialized int values by the
* user (between 0 and ARRAY_SIZE, inclusive).
*/
int guessContentsOfHeapArray() {

/* show that heap memory can be uninitialized */
int* arr = new int[ARRAY_SIZE];
int nCorrect = 0;
for (int i = 0; i < ARRAY_SIZE; i++) {
    cout << "Now guess what number is in arr[" << i << "]: ";
    int guess = getInteger();
    if (guess == arr[i]) {
        cout << "You guessed correctly!" << endl << endl;
        nCorrect++;
    } else {
        cout << "Sorry, the correct answer was " << arr[i] << endl << endl;
    }
}
delete [] arr;
return nCorrect;