

CMSI 182

INTRODUCTION TO COMPUTER SCIENCE

Fall 2007

Assignment 115

This is the payoff — after slogging through how we think about algorithms, let's finally feed them into a computer, so that they can perform these computations for us.

Not for Submission

If you haven't done so already, read Chapter 5 in the Brookshear book.

For Submission

Convert the pseudocode algorithms from last week into working JavaScript programs. The templates given below are intended for use with the JavaScript scratch page that can be found on the course Web site, so that you can enter the values you want in the input fields then click *Run* to see the result:

1. The full program for converting any number from 0 to 255 into 8-bit binary would look like this; fill in the *convertTo8Bit* function:

```
// Shortcuts to the input fields.
var form = document.getElementById("scratch");
var input1 = form.input1;
var input2 = form.input2;
```

```
function convertTo8Bit(number) {
    // Fill this out.
}
```

```
var answer = convertTo8Bit(input1.value);
alert(answer);
```

As you may be able to infer from the code, you can provide the number to convert by typing it into *Input 1* on the Web page. If everything works, you should get an alert box that contains the 8-bit representation (or an error message if the value in *Input 1* was out of range).

2. The full program for “simulating” the three-door game would look like this; fill in the *simulateDoorGame* function:

```
// Shortcuts to the input fields.
var form = document.getElementById("scratch");
var input1 = form.input1;
var input2 = form.input2;
```

```
function simulateDoorGame(trialCount,
    switchDoors) {
    // Fill this out.
}
```

```
var answer = simulateDoorGame(input1.value,
    input2.value);
alert(answer);
```

You can provide the number of “game rounds” to play in *Input 1*. To indicate that you would like to switch doors, type “true” into *Input 2*, or “false” otherwise. If everything works, you should get an alert box that contains the number of times that the “player” chose the winning door.

Submit your programs in some electronic form, so that I can try them out myself.

Extra Credit

You will receive extra credit if you also submit JavaScript versions of the following algorithms, based on the pseudocode given in Tuesday's pseudocode examples handout:

- *makeChange(amount, denominationList)* — The version that can handle any system of coin denominations, expressed in *denominationList*.
- *russianPeasantMultiply(factor1, factor2)* — The version on the right, which adds up the final product right away.