University of Arizona, Department of Computer Science

CSc 340 Foundations of Computer Systems — Homework 1

Christian Collberg

Due Thursday, January 25

1. Consider a computer with a register, an input dial used to input data into the computer, a device that can multiply two numbers, a device that can add two numbers. There is an operation dial that is used to select either of the devices (either the adder: 0 or the multiplier: 1) and the result is stored back in the register. For the above defined system write all the steps involved in the evaluation of the expression

$$(11 * 9) + 13$$

giving the values stored in the registers at the end of each step. Assume the initial value of the register is 0.

2. Write the decimal values (base 10) for the following numbers and also convert them into binary digits (base 2):

   (a) ABC_{16}
   (b) 666_9
   (c) 1110001_3
   (d) 1234_8
   (e) 777_8

3. Convert the following from Octal (base 8) to Hexadecimal (base 16)

   (a) 7173_8
   (b) 345_8
   (c) 7214_8

4. Convert the following from Hexadecimal (base 16) to Octal (base 8)

   (a) DEC_{16}
   (b) 397_{16}
   (c) $B447_{16}$

5. If a computer is built such that a register has 36 bits,

   (a) How many values can a register hold?
   (b) What is the highest value that can be stored in it?

6. Convert the following fractions from decimal to binary

   (a) 9.14_{10}
   (b) 0.68_{10}
   (c) 56.995625_{10}

7. A character string is typically represented as a *null-terminated* array of 8-bit bytes. That is, every string (even the empty string) ends with a *null* (0) byte.

   Write a static Java method `compare` that takes two null-terminated byte arrays `A` and `B` as input and returns -1, 0, or 1 as output. `compare(A,B)` should return 1 if `A>B`, 0 if `A=B`, and -1 if `A<B`.

   Use the following template:

```
class cmp {
   static int compare(byte A[],byte B[]) {
      ...
   }

   public static void main (String[] args) {
     byte s1[] = {0};                    // ""
     byte s2[] = {72,69,76,76,79,0};     // "HELLO"
     byte s3[] = {72,69,76,76,0};        // "HELL"
     byte s4[] = {72,69,76,76,78,79,0}; // "HELLNO"

     System.out.println("compare('','')=" + compare(s1,s1));
     System.out.println("compare('HELLO','HELLO')=" + compare(s2,s2));
     System.out.println("compare('','HELLO')=" + compare(s1,s2));
     System.out.println("compare('HELLO','')=" + compare(s2,s1));
     System.out.println("compare('HELL','HELLO')=" + compare(s3,s2));
     System.out.println("compare('HELLO,'HELL')=" + compare(s2,s3));
     System.out.println("compare('HELLO,'HELLNO')=" + compare(s2,s4));
     System.out.println("compare('HELLNO,'HELLO')=" + compare(s4,s2));
   }
}
```

   Comparing two strings A and B is done character by character. The first position in which the two strings differ determines the result of the comparison. The main program above would yield the following output:

```
compare('','')=0
compare('HELLO','HELLO')=0
compare('','HELLO')=-1
compare('HELLO','')=1
compare('HELL','HELLO')=-1
compare('HELLO,'HELL')=1
compare('HELLO,'HELLNO')=1
compare('HELLNO,'HELLO')=-1
```