Winter 2012	**CPE 101:** Fundamentals of Computer Science I	Alexander Dekhtyar

Homework 2: Midterm 1 Preparation

Due: Friday, January 27, in-class
(We will use this homework on Thursday, January 26 during midterm review, so have it ready by then)

Problem 1. Expressions

Rewrite each C expression using parentheses to show the order of operations.
(e.g. `a+b-c` is `(a+b)-c`.)

(a) `7 + x + 4 / y` ___________________________________

(b) `a + - 13 - - x * 2` ____________________________________

(c) `c * f - 2 == 3 + 2` ____________________________________

(d) `a == 4 && b == 3 * -34` ____________________________________

(e) `a + b - - c == -c - -b` ____________________________________

(f) `3 + - 2 || a == c && b` ____________________________________

(g) `t * ! a == b && r || a && ! b` ____________________________________

(h) `1 + 2 * 3 % 4 || 5 && ! 6` ____________________________________

(i) `a / b / c * d == e - ! f + 3` ____________________________________

(j) `w <= ! 3 + r ||! r && ! w` ____________________________________

Problem 2. Comparisons and logical expressions

Consider the following assignments (all variables are `int` and declared):

```
x = 4;
y = x-1;
z = (x-1)*(y-1);
```

For each expression below, specify what it evaluates to.

```
(a) x >= y                              evaluates to _________
(b) y + x == z                          evaluates to _________
(c) y + x >= z                          evaluates to _________
(d) (x > y) && (z > x)                  evaluates to _________
(e) (x >= y) && (y+x == z)              evaluates to _________
(f) (x >= y) || (y+x == z)              evaluates to _________
(g) !(x>=y)                             evaluates to _________
(h) !( (x >= y) && (y+x == z))          evaluates to _________
(i) !(x >= y) || !(y+x == z)            evaluates to _________
(j) (x == 1) || (x == 3) || !(x == 4)   evaluates to _________
(k) !(!( x == y))                       evaluates to _________
```

Problem 3. Simple functions

In the code fragments below all variables are declared as `int`.

(1) Consider the following code:

```
int boo(int x, int y);

int main() {
  printf("%d\n", boo(5,6));
  printf("%d\n", boo(6,5));
  printf("%d\n", boo(12,0));
  printf("%d\n", boo(-8,-17));
  return(0);
}

int boo(int x, int y){
  if (x>y) {
    return x+1;
  }
  else {
    return x+y;
  }
}
```

What will this program print?

(2) Consider the following functions:

```
int bells(int x) {
   if (x) {
      if (!(x+1)) {
         return 0;
      }
      else {
         return x+1;
      }
   }
   return x-1;
}

int whistles(int x, int y) {
  if (x && y) {
      return x+y;
  }
  else {
    return x*y+1;
  }
}
```

What will the following expressions evaluate to?

(i) `whistles(2,4)` _____

(ii) `whistles(0, 17)` _____

(iii) `bells(15)` _____

(iv) `bells(bells(3))` _____

(v) `bells(whistles(1,2))` _____

(vi) `whistles(bells(1),bells(2))` _____

(vii) `bells(whistles(bells(1),bells(3)))` _____

(ix) `bells(bells(whistles(0,5)))` _____

(x) `whistles(whistles(1,3), whistles(bells(1),bells(3)))` _____

Problem 3. Conditional Statements

(a) Write a function `daysRemaining()` that takes as input two integers: one representing the calendar month (1—12) and one, representing the current date in the month. The function shall return the number of days until the first of the next month.

E.g., `daysRemaining(1,1)` returns 31, `daysRemaining(1,31)` returns 1; `daysRemaining(9,10)` returns 20, and so on.

No variable guards necessary, assume correct values for input parameters.

(b) Rooms 203 – 216 in building 14 are located in the first faculty quad; rooms 217 – 229 — in the second quad, rooms 232– 238 — in the CSL, and rooms 246 – 257 are located in the classroom wing. Write a C function `checkRoom()`, which takes as input the room number, and prints out, `"first quad"`, `"second quad"`, `"CSL"`, `"classroom wing"` depending on the room number. If no location can be found, print `"not sure"`. (no need to `#define` constants here).

(c) Write two functions: `double getMedian()` and `double getMean()`. Each function takes as input three `double` values. `getMedian()` returns the median of the three input numbers, `getMean()` returns the mean of the three input numbers.

(d) Write a function `int meanOrMedian()` which takes as input three `double` values. The function outputs -1 if the mean of the three numbers is greater than the median. It output 1 if the median is greater than the mean. It outputs 0 if the mean is equal to the median. You can use previously developed functions (see part (c) of the problem) in your code.