

1) True or False? Mark each blank clearly with a **T** or **F**

- \_\_\_\_\_ (a) In Java, class names are case sensitive, but reserved words are not.
- \_\_\_\_\_ (b) Every program, except applets, must have a main method.
- \_\_\_\_\_ (c) A constructor must always have the same name as its class.
- \_\_\_\_\_ (d) Syntax errors cause runtime exceptions.
- \_\_\_\_\_ (e) An object's data elements are called instance variables or fields.
- \_\_\_\_\_ (f) Only the number and types of parameters must match when passing to a method.
- \_\_\_\_\_ (g) A constructor always has a void return type.
- \_\_\_\_\_ (h) When a class is extended, new fields can be added.
- \_\_\_\_\_ (i) A subclass inherits all of the constructors, methods, and fields of the superclass.
- \_\_\_\_\_ (j) You always need access to a class' source code to use it.

2) Circle and label the 10 compiler errors in the following method.

- a. variable already declared
- b. cannot find symbol in java.lang.Math
- c. possible loss of precision
- d. found int, required boolean
- e. ';' expected
- f. String concatenation error
- g. variable might not have been initialized
- h. return value missing
- i. '{' missing
- k. missing declaration

```
public int wholsIt (int a, b)
{
    int r = (int)(Math.random * a);

    if (r = b)
        System.out.println("a = " a);

    double [] dd;

    for(int count = 0; count < a; count++)
        dd[count] = count * 2.0;
}

int x = 9.5 * a;

int r = a * b

return;
}
```

3) Convert the decimal number (base 10)  $341_{10}$  to both binary (base 2) and octal (base 8). Circle your answers.

4) Convert the hexadecimal number (base 16)  $E58_{16}$  to both decimal (base 10) and binary (base 2). Circle your answers.

5) Given 2 positive integer values, return the larger value that is in the range  $10 \dots 20$ , inclusive, or return 0 if neither is in that range.

For example:

`max1020(11,19)` should return 19.

`max1020(20,7)` should return 20.

`max1020(9,21)` should return 0.

`max1020(19,21)` should return 19.

```
public int max1020(int a, int b)
```

6) The sum  $1 + (1/2) + (1/3) + (1/4) + (1/5) + (1/6) + \dots + (1/n)$  increases without bound. That is, there is no limit to how big this sum can be, as long as  $n$  is big enough. What's the least value for  $n$  that will give us a sum of at least 2? Well,  $1 + (1/2) + (1/3) = 11/6 < 2$ , and  $1 + (1/2) + (1/3) + (1/4) = 25/12 > 2$ , so  $n = 4$  is the least value necessary to achieve a sum of at least 2.

Write a flowchart that will take as input an **int** value named **targetSum**. The flowchart should determine the least **n** value necessary to achieve the **targetSum**, and return this value.

For example:

**targetSum = 2** should return **4** because  $1 + (1/2) + (1/3) + (1/4) = 25/12 > 2$  is the first sum greater than 2.

**targetSum = 3** should return **11** because  $1 + (1/2) + (1/3) + \dots + (1/10) + (1/11) > 3$  is the first sum greater than 3.

**targetSum = 5** should return **83** because  $1 + (1/2) + (1/3) + \dots + (1/82) + (1/83) > 5$  is the first sum greater than 5.

In writing this flowchart, you may assume that the **targetSum** is greater than or equal to 1. (You will need a separate sheet of paper.)

7) Two positive integers are said to be **relatively prime** if they share no common factors except for 1. Write a method that takes in two positive **ints** in the parameter list, and determines if these two **ints** are relatively prime. The method should return **true** if the numbers are relatively prime, and **false** if the numbers are not relatively prime.

For example:

**relativelyPrime(7,15)** should return **true** because 7 and 15 share no common factor except 1.

**relativelyPrime(14,49)** should return **false** because 14 and 49 share a common factor of 7.

**relativelyPrime(20,100)** should return **false** because 20 and 100 share a common factor of 2 (and others).

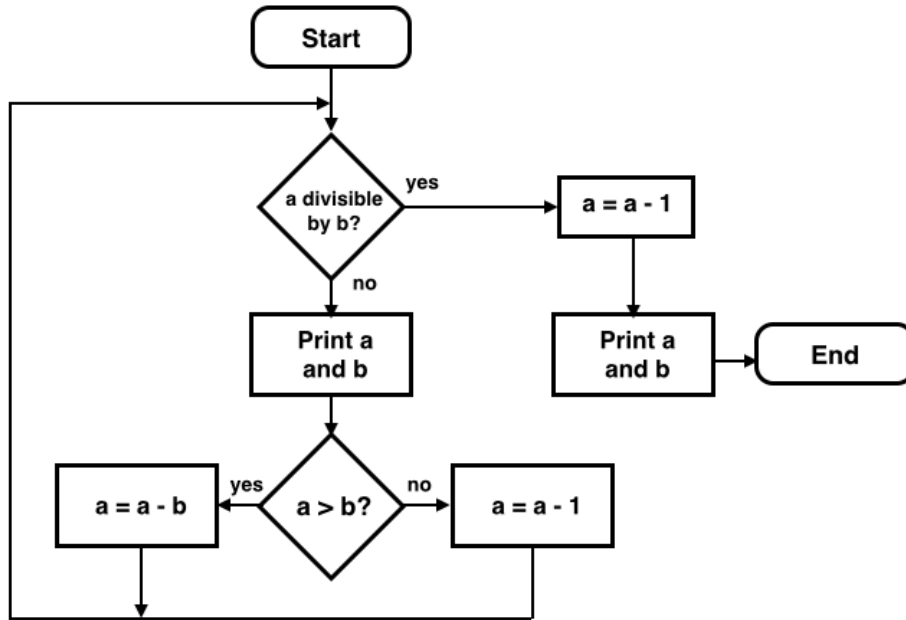
**relativelyPrime(8,21)** should return **true** because 8 and 21 share no common factor except 1.

You may assume that the **ints** passed to this method are positive.

```
public boolean relativelyPrime (int x, int y)
```

8) The following flowchart takes two positive integers as input, **a** and **b**, and prints out the results to the screen. Write a method **secret()** that takes **a** and **b** as parameters and performs the actions of the flowchart. A sample output run is:

a = 20	b = 3
a = 17	b = 3
a = 14	b = 3
a = 11	b = 3
a = 8	b = 3
a = 5	b = 3
a = 2	b = 3
a = 1	b = 3
a = 0	b = 3



```
public void secret (int a, int b)
```