

ANSWER KEY

CS200: Programming I

Instructor: Sam Vinitzky

CS200 Midterm Exam

(50 points)

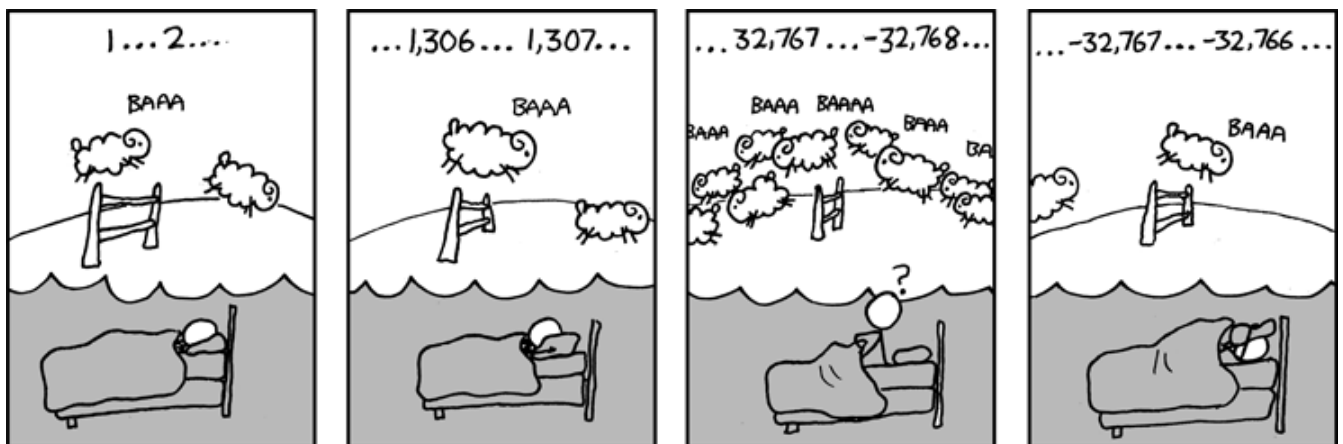
Instructions:

- Do not open this exam until instructed.
- You will have **75 minutes** to complete this exam.
- For questions that ask you to write code, you may write in either pseudocode or Java.¹
- Please write **clearly**. If I cannot read your handwriting, **you will get a 0** until you meet with me in person to read your exam aloud, word by word.²
- **You may not use any electronic devices during the exam.** If you do, Sam will confiscate them, and send embarrassing texts to people in your contacts.³
- Good luck, and have fun!

Name: Answer Key

netID: fakenetID

Student ID number: 123456789



¹If you choose to write in Java, you will not be penalized for small syntax errors (this isn't an issue in pseudocode).

²No, I am not joking.

³Still not joking.

Do not write on this page!

Problem	Score	Possible
1	10	10
2	10	10
3	10	10
4	10	10
5	12	10
Total	52!!!	50

Problem 1: What the heck is going on? [10 points total]

Robert is the proud owner of a local pizzeria. He has trouble remembering things, so he asked his brother Marco to write a program to help with the administration of the restaurant. Unfortunately, Robert is much better at baking pizzas than he is at reading code, so he needs your help deciphering the program! Here is the program that Marco wrote for him:

```
1 public class Pizza{
2     public static int makePizza(int count){
3         count = count + 1;
4         System.out.println(count);
5         return count;
6     }
7
8     public static void main (String[] args){
9         int count = 10;
10        makePizza(count);
11        System.out.println(makePizza(count));
12        // System.out.println(count);
13    }
14 }
```

a) (2 point) When this program is saved as a file, what must the **name of the file** be?

Pizza.java

b) (2 points) When this program is ran, **how many lines** will be printed out?

3

c) (2 points) When line 11 is executed, what will be printed out?

11
11

d) (2 point) Suppose you uncommented line 12, and ran the program. What would line 12 print?

10

e) (2 point) Do lines 4 and 5 do the same thing? If not, what's the difference?

No. Line 4 prints the value of count to the screen, while line 5 returns this value to main.

Problem 2: How many pizzas can I make? [10 points total]

Robert has decided to start using the family's secret pizza recipe at the restaurant. Each pizza that Robert makes using the secret family recipe requires **10 tomatoes**, **1 pound of dough**, and **1.01 pounds of cheese**.

Robert had asked his brother Marco to write a program to determine **how many pizzas** he can make based his supply of each ingredient. However, Marco doesn't think Robert should be using the secret family recipe, so he refused to finish the program! Robert needs your help to finish it.

On the next page, write the code that should go in the corresponding blanks. Your code may be more than one line, if needed.

```
1 public class Ingredients {
2
3     public static int tomatoCheck(int numTomatoes){
4         int numPizzas;
5         // BLANK #1: compute the number of pizzas these tomatoes could make
6         return numPizzas;
7     }
8
9     public static int doughCheck(double poundsOfDough){
10        int numPizzas;
11        // BLANK #2: compute the number of pizzas this amount of dough could make
12        return numPizzas;
13    }
14
15    public static int cheeseCheck(double poundsOfCheese){
16        int numPizzas;
17        // BLANK #3: compute the number of pizzas this amount of cheese could make
18        return numPizzas;
19    }
20
21    public static void main (String[] args){
22        int numTomatoes = 101;
23        double poundsOfDough = 2.4444;
24        double poundsOfCheese = 3.02;
25        int numPizzas;
26
27        // BLANK #4: compute the number of pizzas the above ingredients can make
28
29        System.out.print("You have enough for ingredients for ")
30        System.out.println(numPizzas + " pizzas!");
31    }
32 }
```

Hint: You must assign numPizzas in every blank...

a) (2 points) Write code for blank #1:

```
numPizzas = numTomatoes;
```

b) (2 point) Write code for blank #2:

```
numPizzas = (int) poundsOfCheese;
```

c) (2 points) Write code for blank #3:

```
numPizzas = (int) (poundsOfDough / 1.01);
```

d) (4 points) Write code for blank #4:

Hint: Use the functions to determine which ingredient could make the fewest pizzas...

```
int numPizzasTom = tomatoCheck(101);  
int numPizzasDough = doughCheck(2.4444);  
int numPizzasCheese = cheeseCheck(3.02);
```

```
int minNumber = numPizzasTom;
```

```
if (numPizzasDough < minNumber){  
    minNumber = numPizzasDough;  
}  
if (numPizzasCheese < minNumber){  
    minNumber = numPizzasCheese;  
}  
return minNumber;
```

Problem 3: Who wants this pizza? [10 points total]

Since he began using the secret family recipe, Robert's business has been doing quite well, and he has become very popular around town! However, Robert has a secret – he is afraid of sharp objects. This means he must have his brother Marco slice the pizzas for him! However, Marco is still mad at Robert for using their secret family recipe, so he has been intentionally slicing the pizzas poorly – he slices them into very small pieces, so each pizza has a **large number of slices**.

As if that weren't enough, Robert's main customer base are families that are *extremely* particular. When they receive a pizza, they need the **number of slices** to be evenly divisible by the **number of people in the family**. This is because each person eats as many pieces as possible, and will get mad if they end up having less pizza than someone else in the family.

- a) (5 points) Robert has a pizza that is sliced into 216 pieces. Being an inquisitive man, he is curious as to **how many different sizes of families** he could serve this pizza to.⁴ Write code to determine (and print) this number.

```
total = 0
for i = 1, 2, 3, ..., 216:
    if (216 % i == 0):
        total++
print total
```

⁴For example, a pizza with 12 slices could be served to families of size 1, 2, 3, 4, 6, or 12, which is 6 different sizes.

- b) (5 points) Robert has two pizzas that are sliced into 483 and 395 slices, respectively. Because they were cooked at the same time, **he would like to serve both of these pizzas to the same family**. He wants to get the most bang for his buck, so he would also like to serve them to the **largest** family possible. Write code to determine (and print) the size of the **largest** family to whom Robert could serve **both** of these pizzas.

```
int largest = 0;
for i = 1, 2, 3, ..., 395:
    if (395 % i == 0) and (483 % i == 0):
        largest = i
print i
```

Problem 4: Where did THAT come from? [10 points total]

Although he is a fantastic cook, Robert is not the best at keeping his kitchen clean. Recently, he has noticed a strange⁵ bacteria growing under the stove... Unfortunately, Robert is afraid of germs, and refuses to clean it up until it has shrunk down to be **exactly 1 inch in diameter**. Each day, the bacteria's diameter (measured in **inches**) changes according to the following pattern:

- If the bacteria's diameter is divisible by 3, then its diameter shrinks by a factor of three.
 - If the bacteria's diameter is not divisible by 3, then its diameter increases to be one more than double its current diameter.
- a) (1 points) Suppose that the bacteria's diameter is 4 inches on the first day. On what day would Robert first be able to clean the bacteria? (Don't write code to do it, just figure it out by hand. And show your work!)

Day 1: 4
Day 2: $4*2+1 = 9$
Day 3: 3
Day 4: 1

Can clean on day 4

- b) (4 points) Robert is very unsettled by the bacteria, and is worried it might grow so big that it will over the entire restaurant (which is 300 inches in diameter). Write code to determine (and print) whether or not the bacteria will grow strictly larger than the diameter of the restaurant **before he can clean it up**. Assume the current size of the bacteria is already stored in an integer variable named `diameter`.

```
maxDiameter = diameter
while (diameter != 1):
    if diameter % 3 == 0:
        diameter = diameter / 3
    else:
        diameter = 2*diameter + 1

    if diameter > maxDiameter
        maxDiameter = diameter

if maxDiameter > 300:
    print "Your restaurant is destroyed!"
else
    print "Your restuarant is saved!"
```

⁵The bacteria is a perfect circle.

- c) (5 points) Robert is worried, because he has no idea how long the bacteria has been around. Luckily, he has noticed a pattern – the bacteria’s smell⁶ gets **twice as bad**⁷ every day. Using this information, along with the fact that the bacteria’s current smell ranks at 16,777,216 on the smell-scale, write code to determine (and print) how many days the bacteria has been around. Assume that on the first day, the bacteria ranked at a 1 on the smell-scale.

```
current = 16777216
numDivisions = 0

while current != 1:
    current /= 2
    numDivisions++

print numDivisions
```

⁶Like most things, this is measured on a scale from 1 to 2,147,483,647

⁷Larger numbers smell worse.

Problem 5: Why are there so many bugs? [10 points total]

Because of his terrible sanitation, Robert also has a major issue with **bugs** in his restaurant. He was having trouble keeping track of all of the bugs, so he had his brother Marco write a Java program to do it for him. Programming is hard, and Marco is lazy, so there are still **five** errors in this code.

Identify the 5 errors in the code below by circling them. Then, on the next page, do the following for each error: (1) list the line number of the error, (2) describe the issue, and (3) rewrite the line of code **correctly**, with the error fixed. Each error is worth 2 points total.

```
1 public class Bugs_B_Gone{
2
3     public static int numBugs(){
4         int numBugs = 5;
5         System.out.println(numBugs);
6     }
7
8     public static void main(String[] args){
9
10        int bugs = numBugs();
11        if (bugs <= 10){
12            System.out.println("There are 10 or fewer bugs!");
13        } else (10 < bugs && bugs <100){
14            System.out.println("There are between 10 and 100 bugs!");
15        } else {
16            System.out.println("There are 100 or more bugs!");
17        }
18
19        for (int bugNumber = 1; bugNumber++; bugNumber <= bugs){
20            System.out.println("I am bug numeroer " + 1);
21        }
22
23        int bugNumber = 7;
24        int annoyingNumber = 7; // I hate the number 7!!!
25
26        if (bugNumber = annoyingNumber){
27            System.out.println("Man, that bug's annoying");
28        } else if (bugNumber < annoyingNumber) {
29            System.out.println("Ehh, not that annoying");
30        }
31
32    }
33 }
```

Error #1:

Line number: 5

Issue: missing return statement

Corrected line of code: return numBugs;

Error #2:

Line number: 13

Issue: missing "if"

Corrected line of code: } else if (10 < bugs && bugs < 100){

Error #3:

Line number: 19

Issue: condition and increment order switched

Corrected line of code: for(int bugNumber = 1; bugNumber <= bugs; bugNumber++){

Error #4:

Line number: 20

Issue: variable i doesn't exist

Corrected line of code: System.out.println("I am bug number " + bugNumber);

Error #5:

Line number: 26

Issue: = is assignment, not equality testing

Corrected line of code: if (bugNumber == annoyingNumber){

Error #6:

Line number: 29

Issue: Missing semicolon

Corrected line: System.out.println("Ehh, not that annoying");