

P4: Joseph's Technicolor Coats

(25 points)

Due: Friday, July 5 @ 11:59pm

Part 0: Introduction

Brrrrring brrriiiiing the phone rings – its Joseph, the local tailor! Joseph is well known throughout the community for making over-the-top, eccentric coats. While most coats you see are only one or two colors, Joseph enjoys making coats that are MANY colors. For example, he once made a coat that was red and yellow and green and brown and scarlet and black and ochre and peach and ruby and olive and many other colors!

Joseph is getting older, and his mind isn't as sharp as it used to be. He is no longer able to come up with patterns for his coats on his own, so he needs your help! He has asked that you use your newfound knowledge of loops to write a series of fancy computer programs that will generate complex color patterns for him.

Joseph's computers are slow, but they **can** run conditionals and loops! *They cannot run anything we haven't covered in class...* This includes any of the following: String parsing, built-in Java libraries and classes, arrays, ArrayLists, regular expression, etc... If you have questions about what you are/are not allowed to use, please ask on Piazza.

To grade this assignment, we will be **running** your program and checking that the correct text is printed out¹. We will be checking whether your output **exactly** matches the specifications, including spacing and capitalization, so make sure to read this document carefully!

Each program you turn in should include a comment at the top with (1) your full name, (2) your student ID number, (3) your netID, and (4) the name of anyone you discussed the homework with (excluding Sam and Alex).

Hint from Sam: Each of these problems are doable using a **single** “for” loop, possibly containing one more more “if/else” statements (as necessary). You will **not** need to use multiple nested loops. Also, you do not need to write any functions outside of `main`, but you are welcome to if it would help you.

As always: **start early, ask questions, and have fun!**

¹This is unlike P2 and P3, where we didn't care what you wrote in `main`, and used alternative behind-the-scenes tests to check each function

Part 1: Basic Patterns [8 points total]

As mentioned, Joseph needs your help generating some patterns for his coats. Each program you write will print out a **single** pattern, which will be formatted as a **numbered list of colors**, indicating the order in which Joseph should use them. They should be formatted like this:

- 1) red
- 2) yellow
- 3) green
- 4) red
- 5) blue

Each line should be **number** followed by a “)” and a **space**, and then a **single color** (in all lower case). The example pattern would indicate the Joseph that the first stitch should be red, then the second should be yellow, then the third should be green, etc...

a) A very simple pattern (3 points)

Before Joseph can trust you with the recipe for his **super secret signature pattern**, he needs to test your ability at generating simpler patterns.

The first pattern Joseph wants is **very simple**: red and yellow alternating. Write a program called `VerySimplePattern.java` that **prints out 71 lines** of this pattern when you run it.²

For reference, the first few lines of this pattern are:

- 1) red
- 2) yellow
- 3) red
- 4) yellow
- 5) red

Hint: In order to generate a pattern where every **even** number is yellow, and every **odd** number is red, you may want a conditional statement that tests whether the number is even or odd, and print accordingly...³

²For the record, your code should **not** contain 71 separate print statements...

³Notice that the first 71 lines of this pattern will **not** complete a cycle – i.e. the last color printed should not be yellow...

b) A less simple pattern (5 points)

The next pattern Joseph wants is a little **less simple**: red, yellow, green, and brown, repeated in that order. Write a program called `LessSimplePattern.java` that **prints out the first 93 lines of this pattern** when ran.⁴ Again, you will need to write code in your `main` function.

Additionally, Joseph wants to keep track of **how many** strings of each color he needs. After the pattern is printed out, please also print out the total number of each color that are used.⁵ Follow the formatting used in the example below.

For example, if I were to only print the first **11** lines of the pattern, I would print:

```
1) red
2) yellow
3) green
4) brown
5) red
6) yellow
7) green
8) brown
9) red
10) yellow
11) green
```

```
total red: 3
total yellow: 3
total green: 3
total brown: 2
```

Remember that your program needs to print out the first 93 lines, not the first 11. This was just an example to show the formatting.

Hint: In order to generate this pattern, you will need to test some property of the number to determine which color to print... Look at the example above, and remember “%” ...

⁴Notice that the first 93 lines of this pattern will **not** complete a cycle – i.e. the last color printed should not be brown...

⁵You should not need to **manually** count the totals – your code can do this for you with the magic of **variables**...

Part 2: Substitution Patterns [15 points total]

Those patterns were fine, but Joseph’s specialty is making complicated patterns that involve **substitution**. For example, he might use a simple red/yellow/red/yellow pattern as a base, but replace every 3rd color with *silver*. Then we’d get something like red/yellow/silver/yellow/red/silver...

a) Simple substitution pattern (3 points)

Joseph wants to make sure you are up to the task of producing his super secret signature pattern by having you write a program to generate the relatively **simple substitution pattern** described above. That is, use the original red/yellow alternating pattern as the “base pattern”, **but** if a number is divisible by 3, put “silver” instead of the color we were supposed to use according to the base pattern.

Write a program called `SimpleSubstitutionPattern.java` that prints out the first **121 lines** of this pattern.

For reference, the first 10 lines of this pattern would be:

- 1) red
- 2) yellow
- 3) silver
- 4) yellow
- 5) red
- 6) silver
- 7) red
- 8) yellow
- 9) silver
- 10) yellow

Hint: You may find that the easiest way to write this code is by copying over your code from `VerySimplePattern.java` and adding additional conditional statements...

b) Fancy substitution pattern (5 points)

Joseph thinks you’re ready for a **fancy substitution pattern** which is *very* similar to his super secret signature pattern. This pattern is generated using the following rules:

- The base pattern is **red/yellow/green/brown repeating**, as in part (1b), **but...**
- If the number is **divisible by 3**, then use **silver**.
- If the number is **divisible by 5**, then use **gold**.
- If the number is **divisible by 15** (i.e., divisible by **both 3 and 5**) then use **ruby**.

Write a program called `FancySubstitutionPattern.java` that prints out the first **251 lines** of this pattern.

For reference, the first 17 lines would be:

- 1) red
- 2) yellow
- 3) silver
- 4) brown
- 5) gold
- 6) silver
- 7) green
- 8) brown
- 9) silver
- 10) gold
- 11) green
- 12) silver
- 13) red
- 14) yellow
- 15) ruby
- 16) brown
- 17) red

Hint: You may find that the easiest way to write this code is by copying over your code from `LessSimplePattern.java` and adding additional conditional statements...

c) Joseph's super secret signature pattern (6 points)

Since you have demonstrated exceptional ability at printing out lists of colors, Joseph believes you are finally ready for his world-renowned **super secret signature pattern**. This pattern is a highly contested trade secret, so make sure you do not share it with anyone!!!

The pattern is generated as follows:

- The base pattern is **red/yellow/green/brown/olive/chocolate/peach repeating**, **but...**
- If the number is **divisible by 3**, then use **silver**.

- If the number is **divisible by 5**, then use **gold**.
- If the number is **divisible by 15** (i.e., divisible by **both 3 and 5**) then use **ruby**.
- If the number **ends in a 1** use **violet**.
- If the number **ends in a 1** *and* is **divisible by 3**, use **mauve**.
- If the number **ends in a 1** *and* is **divisible by 5**, use **purple**.
- If the number **ends in a 1** *and* is **divisible by 15**, use **pink**.

Write a program called `SuperSecretSignaturePattern.java` that prints out the first **369 lines** of this pattern.

For legal reasons, I am not allowed to give examples of this pattern, or Joseph could sue me for copyright infringement...

Part ∞: Feedback Form [3 points]

Fill out this [feedback form](#) **after you submit** this assignment. Completion of this will count towards your grade, but your responses themselves will not affect your grade in any way (so be honest!).

What to turn in

On [Canvas](#), turn in a zip folder named `<your_net_id>_P4.zip` containing the files:

- `VerySimplePattern.java` [3 points]
- `LessSimplePattern.java` [5 points]
- `SimpleSubstitutionPattern.java` [3 points]
- `FancySubstitutionPattern.java` [5 points]
- `SuperSecretSignaturePattern.java` [6 points]

Also complete the [feedback form](#). [3 points]
