Fundamentals of Programming
Extra Credit (Argyle): More Loops

Objectives. The objective of this extra credit is to:

1) Give you additional practice with writing while, for, and do-while loops.
2) Give you the opportunity to deconstruct hard problems into combinations of subproblems that can be solved independently. Problem deconstruction is perhaps the most powerful problem-solving ability, and you should master it.

As extra credit, this homework is worth 1% of your final grade if you choose to complete it. That makes it approximately equal to one homework or one question on an exam. **It cannot be submitted late.**

All homework is checked for evidence of plagiarism. Especially this homework. Academic dishonesty carries a tremendous penalty, and offenders will be caught. Copying homework will result in a minimum of your failure of this class, and can include expulsion from Lehigh, and worse.

Assignment. Write a program that displays an argyle pattern in text. See examples below. Query the user for the width and height of the display window (in characters), the size of the argyle diamonds (in characters; they are diamonds with identical side lengths), the width of the argyle stripe, two different characters to fill the argyle diamonds with, and one character to fill the stripe with. While the characters provided as fill can be anything, the dimensions provided must all be positive integers and **the argyle stripe width must be odd and not larger than half of the diamond size**.

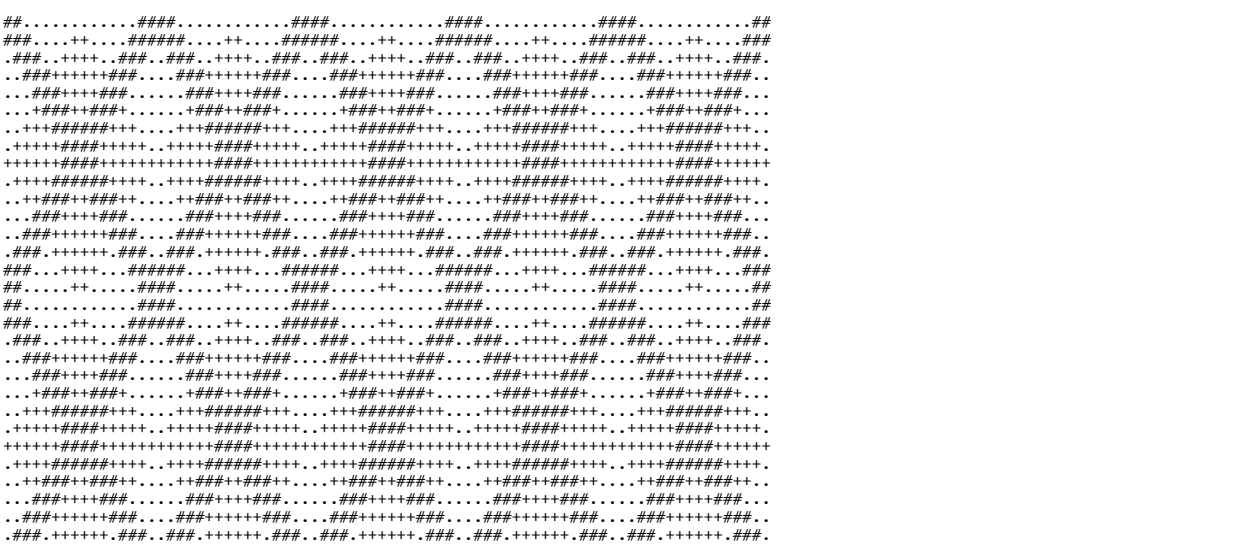
Any inputs that do not satisfy the criteria above must be queried again until the user provides an acceptable input. All char inputs are automatically acceptable, so you do not need to re-query for chars. You can get a char by doing the following:

```
Scanner myScanner = new Scanner(System.in);
String temp = myScanner.next();
char result = temp.charAt(0);
```

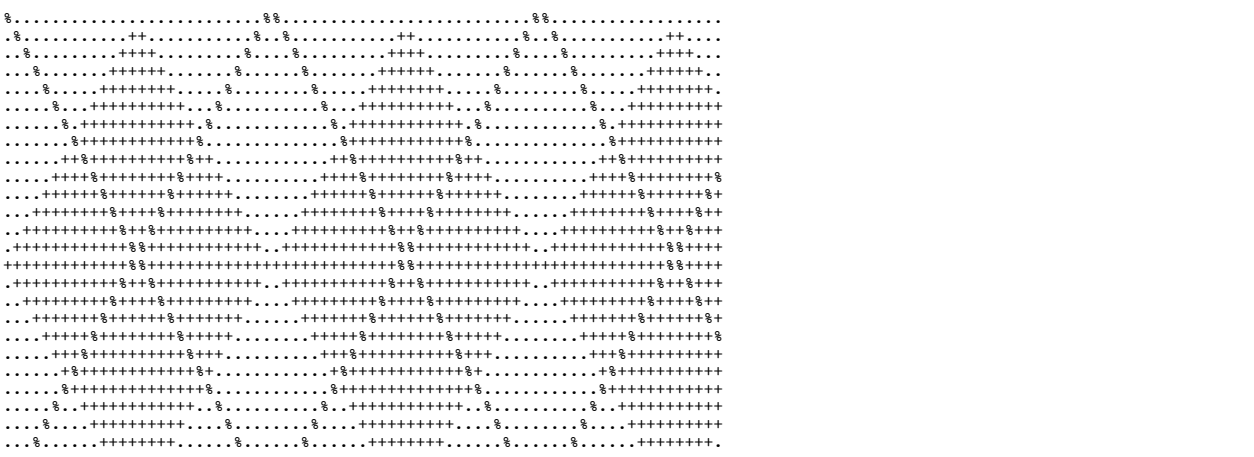
The String method (`.charAt(i)`) gets the character in the string at integer i. If i is bigger than the length of the string, then you'll get a runtime error.

Examples:

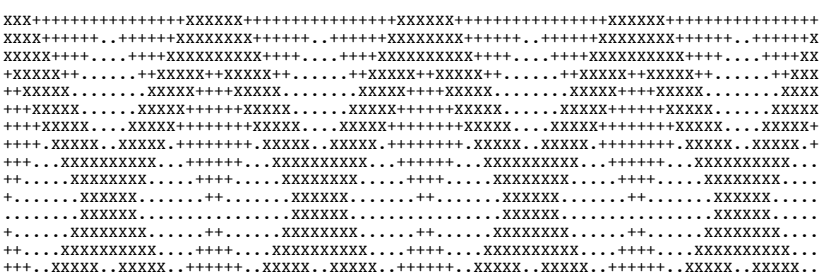
```
please enter a positive integer for the width of Viewing window in characters.  80
please enter a positive integer for the height of Viewing window in characters.  30
please enter a positive integer for the width of the argyle diamonds.  8
please enter a positive odd integer for the width of the argyle center stripe.  3
please enter a first character for the pattern fill.  .
please enter a second character for the pattern fill.  +
please enter a third character for the stripe fill.  #
```



```
please enter a positive integer for the width of Viewing window in characters.  75
please enter a positive integer for the height of Viewing window in characters.  25
please enter a positive integer for the width of the argyle diamonds.  14
please enter a positive odd integer for the width of the argyle center stripe.  1
please enter a first character for the pattern fill.  .
please enter a second character for the pattern fill.  +
please enter a third character for the stripe fill.  %
```



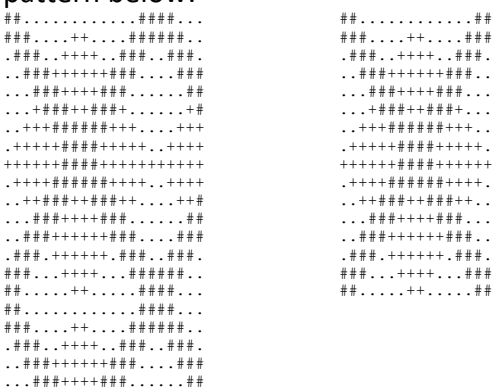
```
please enter a positive integer for the width of Viewing window in characters.  85
please enter a positive integer for the height of Viewing window in characters.  15
please enter a positive integer for the width of the argyle diamonds.  11
please enter a positive odd integer for the width of the argyle center stripe.  5
please enter a first character for the pattern fill.  +
please enter a second character for the pattern fill.  .
please enter a third character for the stripe fill.  X
```



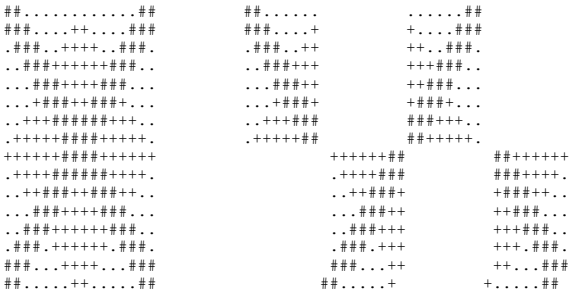
As practice for the exam, implement your solution to this homework using while loops, then again using for loops, then finally using do-while loops to print the argyle pattern.

Hints for doing this homework:

The argyle pattern is a product of a number of independent patterns, including the diamonds and the stripes. Even though there are multiple patterns, they all repeat. The argyle pattern itself is just a diamond in a square with an X through it. You can see the repeating part of the pattern below:



The repeating part of the pattern is itself four simpler patterns:



So rather than trying to write the code finished code from front to back, which is exceptionally difficult and impractical, first try writing the code that will make the simplest of patterns, like the four patterns in the above right. Try generating those patterns with varying stripe thicknesses. Then trying to generate all four of those patterns together to make a single diamond. Then try to put multiple diamonds together.

Useful but cryptic hint: The modulus operator is your friend for repeating patterns.

Store the program in **Argyle.java**. Submit the file on coursesite. Fully correct responses will receive full credit for 1 uncompleted homework, or one extra homework if all homeworks are submitted at the end of the semester.