

CSE1322L Assignment 5 - Spring 2023

Concept Summary:

Recursion

For this assignment¹ you will design a recursive method and the main program that calls the recursive method.

1. This assignment has been adapted from the Roger Frank collection.

Description

Take a piece of paper and fold it in half. Unfold it and there is one fold, which I'll signify with a "v". If you fold the paper twice (always keeping the folds in the same direction and the creases parallel), it will have three folds when you unfold it, and they will be in a "^ v v" pattern. Fold a piece of paper three times, and you will see "^ ^ v v ^ v v" as the sequence of folds.

Folding a piece of already folded paper sounds a lot like recursion, and indeed it is. In case you can't see the algorithm, here it is:

- If you fold a paper once, the fold pattern is "v".
- For additional folds, it is the pattern you have only reversed and flipped over, then a "v" fold, and then the pattern you already have tacked on the end.

Assignment:

- Write a helper method reverse which takes in a string and returns the string backwards. For example, if given "Hello", it will return "olleH"
 - Hints (you don't have to use these, but they may be helpful):
 - You can convert a string to an array of characters using (Java) `string.toCharArray()` or (C#) `string.ToCharArray()`;
 - Once you have an array of characters, you can reverse them pretty easily.
 - You can convert an array of characters back to a string with (Java) `String x=new String(charArray)` or (C#) `string x = new string(charArray)`;
 - You may use a loop or recursion for this method.
- Write a helper method which converts all ^ to v and all v to ^ in a string. It should take in and return a string.
 - Hint: You may find (Java) `string.replace()` or (C#) `String.Replace()` useful.
- Write a static method, paperFold, which is a recursive routine that returns a string representing the fold pattern for a paper folded n times. The driver program will call the paperFold method

As in many recursive solutions, expect the paperFold method to be extremely simple (and of course recursive).

Main method in Java:

```
public static void main(String[] args) {  
    for(int i=1;i<10;i++) {  
        String fold_string=paperfold(i);  
        System.out.println("For "+i+" folds we get: "+fold_string+"\n");  
    }  
}
```

Main method in C#:

```
public static void Main (string[] args) {  
    for(int i=1;i<10;i++)  
    {  
        string fold_string=paperfold(i);  
        Console.WriteLine("For "+i+" folds we get:"+fold_string+"\n");  
    }  
}
```

Sample Output:

For 1 folds we get: v

For 2 folds we get: ^vv

For 3 folds we get: ^^vv^vv

For 4 folds we get: ^^v^^vvv^^vv^vv

For 5 folds we get: ^^v^^vv^^^vv^vvv^^v^^vvv^^vv^vv

For 6 folds we get:

^^v^^vv^^^vv^vv^^^v^^vvv^^vv^vvv^^v^^vv^^^vv^vvv^^vv^vv

For 7 folds we get:

^^v^^vv^^^vv^vv^^^v^^vvv^^vv^vv^^^v^^vv^^^vv^vvv^^v^^vvv^^vv^vvv
^^v^^vv^^^vv^vv^^^v^^vvv^^vv^vvv^^v^^vv^^^vv^vvv^^v^^vvv^^vv^vv

[illegible][illegible]

You'll submit one class on gradescope as Assignment 5.

<https://ccse.kennesaw.edu/fye/submissionguidelines.php>

<https://ccse.kennesaw.edu/fye/courseschedules.php>