Lab Report 3 - Bugs and Commands (Week 5)

Part 1

Failure-inducing input:

```
@Test
public void testReverseInPlace() {
int[] input1 = {1,2,3,4,5};
ArrayExamples.reverseInPlace(input1);
assertArrayEquals(new int[]{5,4,3,2,1}, input1);
}
```

Output:

```
There was 1 failure:
1) testReverseInPlace(ArrayTests)
arrays first differed at element [3]; expected:<2> but was:<4>
```

Okay input:

```
@Test
public void testReverseInPlace() {
int[] input1 = {1};
ArrayExamples.reverseInPlace(input1);
assertArrayEquals(new int[]{1}, input1);
}
```

Output:

```
JUnit version 4.13.2
...
Time: 0.015

OK (3 tests)
```

Before-code:

```
static void reverseInPlace(int[] arr) {
    for(int i = 0; i < arr.length; i += 1) {
        arr[i] = arr[arr.length - i - 1];
    }
}
```

After-code:

```
static void reverseInPlace(int[] arr) {
    for(int i = 0; i < arr.length / 2; i += 1) {
      int temp = arr[i];
      arr[i] = arr[arr.length - i - 1];
      arr[arr.length - i - 1] = temp;
    }
}
```

The before-code overwrites the first half of the array with the second half of the array, so the array is reversed, but the second half of the reversed array is lost. The after-code swaps the first and last elements, then the second and second-to-last elements, and so on, so the array is reversed without losing any elements.

Part 2

`grep`

4 interesting command-line options:

`-i` - ignore case

`grep -i "study" technical/biomed/1468-6708-3-1.txt`

```
events [ 10 ] . In this paper we study whether BMI at
          Study design: The Cardiovascular Health
          Study
          The Cardiovascular Health Study (CHS) is a
          population-based longitudinal study of 5,888 adults aged
        from EVGFP) in the first seven years of the study, adjusted
        CHS Cardiovascular Health Study
```

This matches all lines that contain "study", regardless of case. It is useful when searching for all instances such as "Study", which is not matched by the default case-sensitive search.

`grep -i "Clinical" technical/biomed/1468-6708-3-1.txt`

```
        decreased mortality. Clinical trials powered to detect
        whom risk factors, subclinical disease, and morbidity are
          baseline. Clinical covariates include hypertension,
        significantly greater than zero. A clinical trial of a
          Implications for clinical trials
          YHL, but not YOL. Clinical trials of weight modification
          found for underweight older adults. Clinical trials whose
```

```
             outcome measure. Both YOL and YHL would be clinically
           YHL as the outcome measure in clinical trials involving
```

This matches all lines that contain "Clinical", regardless of case. It is useful when searching for all instances such as "clinical", which is not matched by the default case-sensitive search.

`-n` - print line numbers

```
grep -n "months" technical/biomed/1468-6708-3-1.txt
```

```
57:           brief telephone interview 6 months after each scheduled
85:           months. EVGFP is a simple but well-known measure, which
100:           months, YHL has a reasonably continuous distribution. A
```

This functions as a regular search, except that the line numbers of the lines that match the search term are also shown. This is useful for identifying the line numbers of the lines that match the search term.

```
grep -n "risk factors" technical/biomed/1468-6708-3-1.txt
```

```
27:         In older adults, risk factors may have a greater effect
37:         whom risk factors, subclinical disease, and morbidity are
383:           effects of obesity on risk factors for future health. A
```

This functions as a regular search, except that the line numbers of the lines that match the search term are also shown. This is useful for identifying the line numbers of the lines that match the search term.

`-v` - invert match

```
grep -v "study" technical/biomed/1468-6708-3-1.txt | wc -l
```

```
429
```

This shows the lines that do not match the search term, instead of the lines that do match the search term. This is useful for eliminating lines that match the search term.

```
grep -v "health" technical/biomed/1468-6708-3-1.txt | wc -l
```

```
388
```

This shows the lines that do not match the search term, instead of the lines that do match the search term. This is useful for eliminating lines that match the search term.

`-c` - print count

```
grep -c "study" technical/biomed/1468-6708-3-1.txt
```

```
3
```

This shows the number of lines that match the search term. This is useful when determining the frequency of the search term, instead of the lines themselves.

```
grep -c "health" technical/biomed/1468-6708-3-1.txt
```

```
44
```

This shows the number of lines that match the search term. This is useful when determining the frequency of the search term, instead of the lines themselves.

Information source: Github Copilot

Github Copilot was used to suggest 4 command-line options for `grep`. The specific examples and explanations where written by me, referencing `man grep`.