

PKG 07: LOOPS

Extended ANSWER to the PKG 06 CHALLENGE:

Your short sentence: the FEw tHE pRouD

Original String: the FEw tHE pRouD

To upper case : THE FEW THE PROUD

To lower case : the few the proud

3 e/E found!

The last character is: D

All first to upper: The Few The Proud

```

import java.util.Scanner;

public class CharacterStringPractice {

    public static void main(String[] args) {

        String sOriginal;
        Scanner input = new Scanner(System.in);
        System.out.print("Your short sentence: ");
        sOriginal = input.nextLine();

        String sUpper = sOriginal.toUpperCase(); // not reused
        String sLower = sOriginal.toLowerCase(); // to be reused

        System.out.println("\nOriginal String: " + sOriginal);
        System.out.println("To upper case : " + sUpper);
        System.out.println("To lower case : " + sLower);

        //-----
        int totalE = 0;
        int i = 0;
        while (i < sLower.length()) {
            if (sLower.charAt(i) == 'e') {
                totalE++;
            }
            i++;
        }
        // vs. for (int i = 1; i < sLower.length(); i++)

        if (totalE == 0) {
            System.out.println("No e/E found!");
        } else {
            System.out.println(totalE + " e/E found!");
        }
        // Also see: equalsIgnoreCase, compareToIgnoreCase, logical operator && ||

        //-----
        char lastChar = sOriginal.charAt(sOriginal.length() - 1);
        System.out.println("The last character is: " + lastChar);

        //-----
        // First character of all words to upper. Handling very simple strings.
        char cPrevious = 0, cCurrent = 0;
        String sUpperFirstChars = sLower.substring(0, 1).toUpperCase();

        for (int j = 1; j < sLower.length(); j++) {
            cCurrent = sLower.charAt(j);

            if (cPrevious == ' ') { // Unicode Space 32
                cCurrent = Character.toUpperCase(cCurrent);
            }

            sUpperFirstChars = sUpperFirstChars.concat(String.valueOf(cCurrent));
            //sUpperFirstChars += String.valueOf(cCurrent);
            cPrevious = cCurrent;
        }

        System.out.println("All first to upper: " + sUpperFirstChars);
    }
}

```

64 LOOPS, Daniel Liang

65 while LOOP, pretest loop

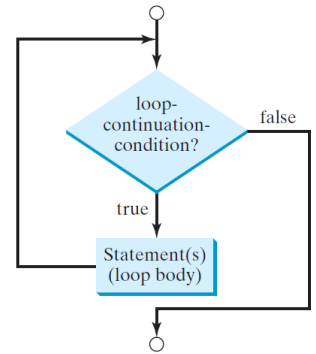
- 66
- 67 - Executes statements repeatedly while the condition is true.
 - 68 - An *infinite loop* is a loop which runs forever.
- 69

70

```
71  int count = 0;
72  while (count < 5) {
73      System.out.println(count + ". CSC210!");
74      count++;
75  }
```

76

```
0. CSC210!
1. CSC210!
2. CSC210!
3. CSC210!
4. CSC210!
```



77 do-while LOOP, posttest loop

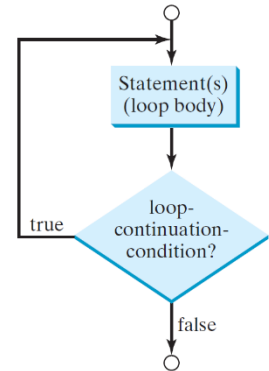
- 78
- 79 - A do-while loop is the same as a while loop except that it
 - 80 executes the loop body first then checks the loop continuation
 - 81 condition.
- 82

83

```
84  int count = 0;
85  do {
86      System.out.println(count + ". CSC210!");
87      count++;
88  } while (count < 5); // semicolon
```

89

```
0. CSC210!
1. CSC210!
2. CSC210!
3. CSC210!
4. CSC210!
```



90 for LOOP, pretest loop

- 91
- 92 - Control structure is between a pair of parentheses.
 - 93 - (*initial-action; loop-continuation-condition; action-after-each-iteration*)
 - 94 - Control variable controls number of iterations.
- 95

```
96  for (int count = 0; count < 5; count++) {
97      System.out.println(count + ". CSC210!");
98      // count++ happens here
99  }
```

100

```
101 // System.out.println(count); // ERROR: Cannot find symbol
```

102

103 -----

```
104  int i;
105  for (i = 0; i < 5; i++ ) {
106      System.out.println(i + ". CSC210!");
107  }
```

108

```
109  System.out.println(i); // 5
```

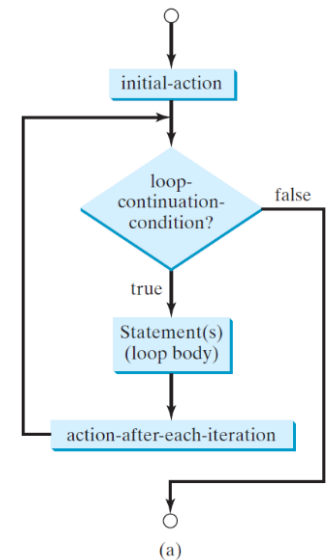
110

111 -----

```
112  for (int x = 0, y = 5; x < 5 && y > 0; x++, y--) {}
```

113

```
0. CSC210!
1. CSC210!
2. CSC210!
3. CSC210!
4. CSC210!
```



114 Nested LOOPS

- 115
- 116 - Nested loops consist of an *outer loop* and one or more *inner loops*. Each time the
 - 117 outer loop is repeated, the inner loops are reentered, and started anew.
- 118

```
119  for (int i = 0; i < 5; i++) {
120      System.out.print("Row " + i + ": ");
121      for (int j = 0; j < 5; j++) {
122          System.out.print(j + " ");
123      }
124      System.out.println("");
125  }
```

```
Row 0: 0 1 2 3 4
Row 1: 0 1 2 3 4
Row 2: 0 1 2 3 4
Row 3: 0 1 2 3 4
Row 4: 0 1 2 3 4
```

126 LOOPS, break and continue

```
127
128 break
129
130     for (int i = 0; i < 5; i++) {
131         System.out.print("Row " + i + ": ");
132
133         for (int j = 0; j < 5; j++) {
134             System.out.print(j + " ");
135
136             if (j == 3) {
137                 System.out.print("Breaking...");
138                 break;
139             }
140
141         }
142
143         System.out.println("");
144     }
145     System.out.println("Out of nested loops!");
146
```

Row 0:	0	1	2	3	Breaking...
Row 1:	0	1	2	3	Breaking...
Row 2:	0	1	2	3	Breaking...
Row 3:	0	1	2	3	Breaking...
Row 4:	0	1	2	3	Breaking...
Out of nested loops!					

147
148 break and label

Row 0:	0	1	2	3	Breaking... Out of nested loops!
--------	---	---	---	---	----------------------------------

```
149
150     breakNestedLoops:
151
152     for (int i = 0; i < 5; i++) {
153         System.out.print("Row " + i + ": ");
154
155         for (int j = 0; j < 5; j++) {
156             System.out.print(j + " ");
157
158             if (j == 3) {
159                 System.out.print("Breaking...");
160                 break breakNestedLoops;
161             }
162
163         }
164
165         System.out.println("");
166     }
167
168     System.out.println("Out of nested loops!");
169
```

170
171 continue

```
172
173     for (int i = 0; i < 5; i++) {
174         System.out.print("Row " + i + ": ");
175
176         if (i == 1 || i == 3) {
177             System.out.print("Continuing...");
178             continue;
179         }
180
181         for (int j = 0; j < 5; j++) {
182             System.out.print(j + " ");
183         }
184
185         System.out.println("");
186     }
187
188     System.out.println("Out of nested loops!");
```

Row 0:	0	1	2	3	4
Row 1:	Continuing...				
Row 2:	0	1	2	3	4
Row 3:	Continuing...				
Row 4:	0	1	2	3	4
Out of nested loops!					

190

191 Challenge 1:

192

193 - Convert the *while* loop on line #26 to *do-while* and *for* loops.194 - Convert the *for* loop on line #49 to *do-while* and *while* loops.

195

196

197

198

199

200

201 Challenge 2:

202

203 - Convert the *for* loop to *while* and *do-while* loops.

204

```
14 public static void main(String[] args) {
15     int orgNum = 5;
16     int curNum = orgNum;
17
18     for (int i = 0; i < 20; i++) {
19
20         if (i == 10 || i == 15) {
21             break;
22         }
23         curNum += 5;
24         System.out.println("This run: " + i
25                             + "\tOriginal Number: " + orgNum
26                             + " \tCurrent Number: " + curNum + "");
27
28     }
29
30 }
31 }
```

205

206

207

208

209

210

211

212

213 Challenge 3:

214

215 - Please write a program which behaves like in the below sample run:

216

217 Please enter an integer for dividend: 17

218 |-----|

219 MODULAR MATH ---

220 |-----|

221 Idx. Dividend Modulo 5

222 0 17 2

223 1 14 4

224 2 11 1

225 3 8 3

226 4 5 0

227 |-----|

228

229

230

231

232

```

234
235 /**
236  * Count the number of operations in
237  * the control structure of the for-loop
238  *
239  * * * ORDER:
240  * * 1. i = 0    happens ONCE
241  * * 2. i < n    happens every time
242  * * 3. EXE      for-loop body statements execute only when step 2 is satisfied
243  * * 4. i++      happens only when step 3 happens
244  *
245  * for (int i = 0;          i < n;          i++ )
246  * i      assignment      comparison      assignment & addition
247  * 0      1              1              1              1
248  * 1      0              1              1              1
249  * 2      0              1              1              1
250  * 3      0              1              1              1
251  * ...
252  * n      0              1              0              0
253  *
254  * Total: 1 + (n + 1) + 2n = 3n + 2
255  */
256
257 // Example
258 // Count all operations: =, numeric, return, compare...
259 // int i, n = 10, sum = 0;
260 // for (i = 0; i <= 2n; i++) {
261 //     sum += i + 3;
262 // }
263
264 public class OpsInForLoopPractice {
265
266     public static void main(String[] args) {
267
268         int i, n = 10, sum = 0;          // A, 2 assignments, 2 ops, once
269
270         // B: When for-loop starts, i = 0. 1 assignment, 1 ops, once only
271         // C: i <= 2*n. 1 multiplication and 1 comparison, 2 ops, once each loop
272         // D: sum = sum + i + 3. 1 assignment and 2 additions, 3 ops, once each loop
273         // E: i = i + 1. 1 addition and 1 assignment, 2 ops, once each loop
274         int opTotal = 1;                  // B. Counter not included.
275
276         // E
277         for (i = 0; i <= 2*n; i++, opTotal += 2) { // E
278
279             sum += i + 3;                  // D
280             opTotal += 2 + 3;              // C and D. Counter not included.
281             System.out.println("i: " + i + "\tTotal OPS: " + opTotal);
282
283         }
284
285         // F: When i = 2n + 1, +1 multiplication and +1 comparison.
286         opTotal += 2 + 2;                  // A and F. Counter not included.
287         System.out.println("i: " + i + "\tTotal OPS: " + opTotal);
288
289         // 1 + (2n + 2)*2 + (2n + 1)*2 = 8n + 7
290         //             (2n + 1)*3 = 6n + 3
291         //             +2 = 2
292         //             = 14n + 12
293         // if n == 10, sum = 152
294     }
295 }

```

i: 0	Total OPS: 6
i: 1	Total OPS: 13
i: 2	Total OPS: 20
i: 3	Total OPS: 27
i: 4	Total OPS: 34
i: 5	Total OPS: 41
i: 6	Total OPS: 48
i: 7	Total OPS: 55
i: 8	Total OPS: 62
i: 9	Total OPS: 69
i: 10	Total OPS: 76
i: 11	Total OPS: 83
i: 12	Total OPS: 90
i: 13	Total OPS: 97
i: 14	Total OPS: 104
i: 15	Total OPS: 111
i: 16	Total OPS: 118
i: 17	Total OPS: 125
i: 18	Total OPS: 132
i: 19	Total OPS: 139
i: 20	Total OPS: 146
i: 21	Total OPS: 152

```
297
298 1. How many times will the following code print "Welcome to Java"?
299 int count = 0;
300 while (count < 10) {
301     System.out.println("Welcome to Java");
302     count++;
303 }
304 a. 8          b. 9          c. 10          d. 11          e. 0
305
306 2. Analyze the following code.
307 int count = 0;
308 while (count < 100) {
309     // Point A
310     System.out.println("Welcome to Java!");
311     count++;
312     // Point B
313 }
314 // Point C
315 a. count < 100 is always true at Point A
316 b. count < 100 is always true at Point B
317 c. count < 100 is always false at Point B
318 d. count < 100 is always true at Point C
319 e. count < 100 is always false at Point C
320
321 3. How many times will the following code print "Welcome to Java"?
322 int count = 0;
323 while (count++ < 10) {
324     System.out.println("Welcome to Java");
325 }
326 a. 8          b. 9          c. 10          d. 11          e. 0
327
328 4. What is the output of the following code?
329 int x = 0;
330 while (x < 4) {
331     x = x + 1;
332 }
333 System.out.println("x is " + x);
334 a. x is 0    b. x is 1    c. x is 2    d. x is 3    e. x is 4
335
336 5. What will be displayed when the following code is executed?
337 int number = 6;
338 while (number > 0) {
339     number -= 3;
340     System.out.print(number + " ");
341 }
342 a. 6 3 0      b. 6 3      c. 3 0      d. 3 0 -3      e. 0 -3
343
344 6. How many times will the following code print "Welcome to Java"?
345 int count = 0;
346 do {
347     System.out.println("Welcome to Java");
348     count++;
349 } while (count < 10);
350 a. 8          b. 9          c. 10          d. 11          e. 0
351
352 7. How many times will the following code print "Welcome to Java"?
353 int count = 0;
354 do {
355     System.out.println("Welcome to Java");
356 } while (count++ < 10);
357
358 a. 8          b. 9          c. 10          d. 11          e. 0
```

```

359 8. How many times will the following code print "Welcome to Java"?
360 int count = 0;
361 do {
362     System.out.println("Welcome to Java");
363 } while (++count < 10);
364 a. 8          b. 9          c. 10          d. 11          e. 0
365
366 9. What is the value in count after the following loop is executed?
367 int count = 0;
368 do {
369     System.out.println("Welcome to Java");
370 } while (count++ < 9);
371 System.out.println(count);
372 a. 8          b. 9          c. 10          d. 11          e. 0
373
374 10. Analyze the following statement:
375 double sum = 0;
376 for (double d = 0; d < 10;) {
377     d += 0.1;
378     sum += sum + d;
379 }
380 a. The program has a compile error because the adjustment is missing in the for loop.
381 b. The program has a compile error because the control variable in the for loop cannot
382 be of the double type.
383 c. The program runs in an infinite loop because d < 10 would always be true.
384 d. The program compiles and runs fine.
385
386 11. Which of the following loops prints "Welcome to Java" 10 times?
387 A:
388 for (int count = 1; count <= 10; count++) {
389     System.out.println("Welcome to Java");
390 }
391
392 B:
393 for (int count = 0; count < 10; count++) {
394     System.out.println("Welcome to Java");
395 }
396
397 C:
398 for (int count = 1; count < 10; count++) {
399     System.out.println("Welcome to Java");
400 }
401
402 D:
403 for (int count = 0; count <= 10; count++) {
404     System.out.println("Welcome to Java");
405 }
406 a. BD          b. ABC          c. AC          d. BC          e. AB
407
408 12. Which of the following loops correctly computes  $1/2 + 2/3 + 3/4 + \dots + 99/100$ ?
409 A:
410 double sum = 0;
411 for (int i = 1; i <= 99; i++) {
412     sum = i / (i + 1);
413 }
414 System.out.println("Sum is " + sum);
415
416 B:
417 double sum = 0;
418 for (int i = 1; i < 99; i++) {
419     sum += i / (i + 1);
420 }
421 System.out.println("Sum is " + sum);

```

```

422 C:
423 double sum = 0;
424 for (int i = 1; i <= 99; i++) {
425     sum += 1.0 * i / (i + 1);
426 }
427 System.out.println("Sum is " + sum);
428
429 D:
430 double sum = 0;
431 for (int i = 1; i <= 99; i++) {
432     sum += i / (i + 1.0);
433 }
434 System.out.println("Sum is " + sum);
435
436 E:
437 double sum = 0;
438 for (int i = 1; i < 99; i++) {
439     sum += i / (i + 1.0);
440 }
441 System.out.println("Sum is " + sum);
442
443 a. BCD      b. ABCD      c. B      d. CDE      e. CD
444
445 13. The following loop displays _____.
446 for (int i = 1; i <= 10; i++) {
447     System.out.print(i + " ");
448     i++;
449 }
450 a. 1 2 3 4 5 6 7 8 9      b. 1 2 3 4 5 6 7 8 9 10
451 c. 1 2 3 4 5              d. 1 3 5 7 9              e. 2 4 6 8 10
452
453 14. Do the following two statements in (I) and (II) result in the same value in sum?
454 (I):
455 for (int i = 0; i < 10; ++i) {
456     sum += i;
457 }
458
459 (II):
460 for (int i = 0; i < 10; i++) {
461     sum += i;
462 }
463 a. Yes      b. No
464
465 15. What is the output for y?
466 int y = 0;
467 for (int i = 0; i < 10; ++i) {
468     y += i;
469 }
470 System.out.println(y);
471
472 a. 10      b. 11      c. 12      d. 13      e. 45
473
474 16. What is i after the following for loop?
475 int y = 0;
476 for (int i = 0; i < 10; ++i) {
477     y += i;
478 }
479 a. 9      b. 10      c. 11      d. undefined
480
481 17. Is the following loop correct?
482 for ( ; ; );
483 a. Yes      b. No
484

```


485 18. Analyze the following fragment:

```
486 double sum = 0;
487 double d = 0;
488 while (d != 10.0) {
489     d += 0.1;
490     sum += sum + d;
491 }
```

- 492
- 493 a. The program does not compile because sum and d are declared double, but assigned
 - 494 with integer value 0.
 - 495 b. The program never stops because d is always 0.1 inside the loop.
 - 496 c. The program may not stop because of the phenomenon referred to as numerical
 - 497 inaccuracy for operating with floating-point numbers.
 - 498 d. After the loop, sum is $0 + 0.1 + 0.2 + 0.3 + \dots + 1.9$
- 499

500 19. Analyze the following code:

```
501 public class Test {
502     public static void main (String args[]) {
503         int i = 0;
504         for (i = 0; i < 10; i++);
505         System.out.println(i + 4);
506     }
507 }
```

- 508
- 509 a. The program has a compile error because of the semicolon (;) on the for loop line.
 - 510 b. The program compiles despite the semicolon (;) on the for loop line, & displays 4.
 - 511 c. The program compiles despite the semicolon (;) on the for loop line, & displays 14.
 - 512 d. The for loop in this program is same as for (i = 0; i < 10; i++) { };
- 513 System.out.println(i + 4);
- 514

515 20. How many times is the println statement executed?

```
516 for (int i = 0; i < 10; i++)
517     for (int j = 0; j < i; j++)
518         System.out.println(i * j)
```

- 519
- 520 a. 100 b. 20 c. 10 d. 45
- 521

522 21. Given the following four patterns,

523

524 Pattern A	Pattern B	Pattern C	Pattern D
525 1	1 2 3 4 5 6	1	1 2 3 4 5 6
526 1 2	1 2 3 4 5	2 1	1 2 3 4 5
527 1 2 3	1 2 3 4	3 2 1	1 2 3 4
528 1 2 3 4	1 2 3	4 3 2 1	1 2 3
529 1 2 3 4 5	1 2	5 4 3 2 1	1 2
530 1 2 3 4 5 6	1	6 5 4 3 2 1	1

531

532 Which of the pattern is produced by the following code?

```
533 for (int i = 1; i <= 6; i++) {
534     for (int j = 6; j >= 1; j--)
535         System.out.print(j <= i ? j + " " : " " + " ");
536     System.out.println();
537 }
```

- 538
- 539 a. Pattern A b. Pattern B c. Pattern C d. Pattern D
- 540

541 22. How many times is the println statement executed?

```
542 for (int i = 0; i < 10; i++)
543     for (int j = 0; j < 10; j++)
544         System.out.println(i * j);
```

- 545
- 546 a. 100 b. 20 c. 10 d. 45
- 547

548 23. To add $0.01 + 0.02 + \dots + 1.00$, what order should you use to add the numbers to
549 get better accuracy?
550
551 a. add 0.01, 0.02, ..., 1.00 in this order to a sum variable whose initial value is 0.
552 b. add 1.00, 0.99, 0.98, ..., 0.02, 0.01 in this order to a sum variable whose initial
553 value is 0.
554

555 24. Analyze the following code.
556 `double sum = 0;`
557 `for (double d = 0; d < 10; sum += sum + d) {`
558 `d += 0.1;`
559 `}`
560

561 A. The program has a syntax error because the adjustment statement is incorrect in the
562 for loop.
563 B. The program has a syntax error because the control variable in the for loop cannot
564 be of the double type.
565 C. The program compiles but does not stop because d would always be less than 10.
566 D. The program compiles and runs fine.
567

568 25. What is y after the following for loop statement is executed?
569 `int y = 0;`
570 `for (int i = 0; i < 10; ++i) {`
571 `y += 1;`
572 `}`
573 A. 9 B. 10 C. 11 D. 12
574

575 26. Will the following program terminate?
576 `int balance = 10;`
577
578 `while (true) {`
579 `if (balance < 9)`
580 `break;`
581 `balance = balance - 9;`
582 `}`
583 a. Yes b. No
584

585 27. What is sum after the following loop terminates?
586 `int sum = 0;`
587 `int item = 0;`
588 `do {`
589 `item++;`
590 `sum += item;`
591 `if (sum > 4)`
592 `break;`
593 `}`
594 `while (item < 5);`
595 a. 5 b. 6 c. 7 d. 8 e. 9
596

597 28. What is the output after the following loop terminates?
598 `int number = 25;`
599 `int i;`
600
601 `boolean isPrime = true;`
602 `for (i = 2; i < number && isPrime; i++) {`
603 `if (number % i == 0) {`
604 `isPrime = false;`
605 `}`
606 `}`
607 `System.out.println("i is " + i + " isPrime is " + isPrime);`
608
609 a. i is 5 isPrime is true b. i is 5 isPrime is false
610 c. i is 6 isPrime is true d. i is 6 isPrime is false

```

611 29. What is the output after the following loop terminates?
612 int number = 25;
613 int i;
614
615 boolean isPrime = true;
616 for (i = 2; i < number; i++) {
617     if (number % i == 0) {
618         isPrime = false;
619         break;
620     }
621 }
622
623 System.out.println("i is " + i + " isPrime is " + isPrime);
624
625 a. i is 5 isPrime is true           b. i is 5 isPrime is false
626 c. i is 6 isPrime is true           d. i is 6 isPrime is false
627
628 30. What is sum after the following loop terminates?
629 int sum = 0;
630 int item = 0;
631 do {
632     item++;
633     if (sum >= 4)
634         continue;
635     sum += item;
636 }
637 while (item < 5);
638
639 a. 6           b. 7           c. 8           d. 9           e. 10
640
641 31. Will the following program terminate?
642 int balance = 10;
643
644 while (true) {
645     if (balance < 9)
646         continue;
647     balance = balance - 9;
648 }
649
650 a. Yes           b. No
651
652 32. What balance after the following code is executed?
653
654 int balance = 10;
655
656 while (balance >= 1) {
657     if (balance < 9)
658         continue;
659     balance = balance - 9;
660 }
661
662 A. -1           B. 0           C. 1           D. 2           E. The loop does not end
663
664 33. What is the value of balance after the following code is executed?
665 int balance = 10;
666
667 while (balance >= 1) {
668     if (balance < 9)
669         break;
670     balance = balance - 9;
671 }
672
673 A. -1           B. 0           C. 1           D. 2

```

```

674 34. What is the number of iterations in the following loop?
675     for (int i = 1; i < n; i++) {
676         // iteration
677     }
678 a. 2*n      b. n      c. n - 1      d. n + 1
679
680 35. What is the number of iterations in the following loop?
681     for (int i = 1; i <= n; i++) {
682         // iteration
683     }
684
685 a. 2*n      b. n      c. n - 1      d. n + 1
686
687 36. Suppose the input for number is 9. What is the output from running the following
688 program?
689
690 import java.util.Scanner;
691
692 public class Test {
693     public static void main(String[] args) {
694         Scanner input = new Scanner(System.in);
695         System.out.print("Enter an integer: ");
696         int number = input.nextInt();
697
698         int i;
699
700         boolean isPrime = true;
701         for (i = 2; i < number && isPrime; i++) {
702             if (number % i == 0) {
703                 isPrime = false;
704             }
705         }
706
707         System.out.println("i is " + i);
708
709         if (isPrime)
710             System.out.println(number + " is prime");
711         else
712             System.out.println(number + " is not prime");
713     }
714 }
715
716 a. i is 3 followed by 9 is prime
717 b. i is 3 followed by 9 is not prime
718 c. i is 4 followed by 9 is prime
719 d. i is 4 followed by 9 is not prime
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736

```