# Computer Science Answer Key
# UIL State 2015

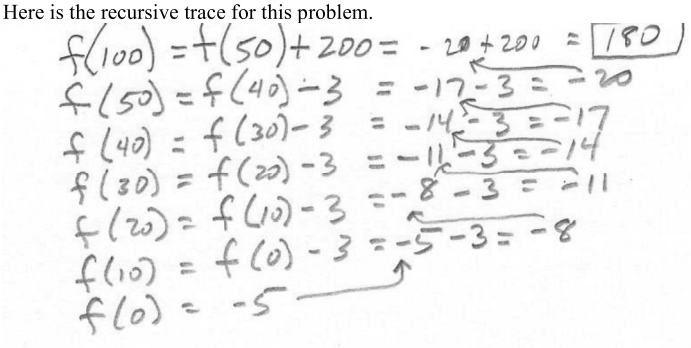
| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1) | C | 11) | A | 21) | A | 31) | C |
| 2) | D | 12) | C | 22) | D | 32) | B |
| 3) | C | 13) | E | 23) | C | 33) | D |
| 4) | C | 14) | D | 24) | B | 34) | A |
| 5) | B | 15) | A | 25) | A | 35) | B |
| 6) | A | 16) | C | 26) | B | 36) | A |
| 7) | A | 17) | A | 27) | B | 37) | C |
| 8) | D | 18) | D | 28) | C | 38) | E |
| 9) | D | 19) | B | 29) | B | 39) | -87 |
| 10) | D | 20) | D | 30) | D | 40) | 21 |

**Note to Graders:**

- All provided code segments are intended to be syntactically correct, unless otherwise stated (e.g. error is an answer). **Ignore any typographical errors.**
- Any necessary Standard Java 2 Packages are assumed to have been imported as needed.
- Assume any undefined (undeclared) variables have been defined as used.

# Explanations:

1. The equation $\mathbf{10000_2X_{16} - 16_8 = 226_{10}}$ translates into the base 10 equation **16X - 14 = 226**, which transforms to **16X = 240**, and finally **X=15**, which is the value **F** in base 16.
2. `6 + 4 * 10 - 8` ➔ `6 + 40 - 8` ➔ `46 - 8` ➔ `38`
3. The three line feeds are created by the `\n`, `%n`, and `\n`. The `\\n` does not create a line feed, but instead outputs "`\n`". Also, unlike the println statement, there is no automatic line feed created by the printf method.
4. The string "moonligh" is 4 characters ***greater*** than the string "ight", since the letter 'm' is four places beyond the letter 'i' in the alphabet.
5. Since P XOR Q is false when both values of P and Q are true, the NOT value is true.
6. The sin ratio of a 30 degree angle (PI/6) is 0.5 (opposite over hypotenuse), which requires the value of b to be 2 since b is the shortest side of 30-60-90 triangle whose hypotenuse is 4. The actual output of Math.sin(Math.PI/6) is 0.49999999999999994, which is virtually 0.5, but using == would output false, thus requiring the subtraction expression as shown to test for virtual equality.
7. '9' += ~2 ➔ '9' += -3 ➔ '6'. The += preserves the char because of autocasting. **Note:** The ~ (complement) operator gives the opposite of a value minus 1, therefore the opposite of 2 is -2, minus 1 is negative 3. Conversely, ~-3 produces 2 (opposite of -3 is 3, minus 1 is 2).
8. The color blue is output 5 times (-5 to -1), black once(0), red 8 times(1 through 8), and green twice(9 and 10).
9. The output sequence is: 1 4 13 40 121 364.
10. The output values are: 0.0 0.0 1.0 1.0 2.0
11. The code loops three times using the strings "`1 2 3 4 `", "` 6 `", and "` 8 9 10`", taking the first non-space element of each string, resulting in an output of "168"
12. Since the ASCII values of "ABCDE" are 65-69, these characters are accumulated in the string, but in reverse order.
13. In order for this code to work properly, a parentheses must be placed into the expression `out.println(`**`(x|y)`**`==z?y<<x:z>>x);` since **==** has a higher order of precedence than **|**, and bitwise OR cannot take a boolean value as an operand along with an integer.
14. Since the increment happens before the output and after the check, the last few outputs of this code are: 126, 127, -128, with the byte value wrapping around to the minimum value to end the loop.
15. Although the removeAll method only removes one item, the return value is still true since something was removed, and the contents of List1 is the same, less the 5.
16. The initial principal amount is 1000.0, compounded annually at 10% for 2 years, which after the first year is 1100.0 (1000+100), and after two years 1210.0 (1100+110).
17. The correct adjustment increases the number of loop iterations by 12, and divides by 12 the interest compounded.
18. The complete output of the code is: 1 2 2 4 **6** 8 **3 6 9 12 15 18** 4 8 **12** 16 20 **24** 28 32, showing 9 multiples of 3.
19. If A and B are both false, their NOT values are both true, making the XOR value false, and the NOT XOR value true, which when ANDed with a true value of C causes the entire expression to be true. The triple 111 will also make this expression true.
20. Since this class uses the compareTo method, an undefined method of the Comparable interface, "implements Comparable" is required in the class header.
21. In this case, the required parameter type for the compareTo method is Object. Once inside the method, the object reference can be changed to something more specific.
22. Line 10 contains a logic error, or bug. The code should be **`teamScore-=scores[0];`** since the lowest team member's score is in position zero after the Arrays.sort statement, which arranges the scores in ascending order.
23. Team five wins over team six because it has a fourth member. Team seven wins the 5-way tie by virtue of the greater 4th score. Teams one and two tie, but have a fourth member, thus ranking higher than teams three and four, who also tied. However, since the Collections.sort did not change the relative positions of the tied teams, when the list is reversed, team two is listed before one, and four before three.
24. The equation setup is: $1024*10/5 = N\log_2 N/24$, which resolves to 4096*12/24, with an N value of 4096.
25. `33&25 ==> 100001&11001 ==> 000001 ==> 1`
26. The 19 values of y in the loop process are: `-2 0 2 4 6 -1 1 3 5 0 2 4 6 1 3 5 2 4 6`
27. The "front, back" sequence for the first five MergeSortHelper calls is: 0 5, 0 2, 0 1, 0 0, 1 1, at which time the first merge occurs for the first two elements in the list. MergeSortHelper 6 uses the front/back values 2 2, at which time the next merge occurs between the first list that was merged (containing elements in positions 0 and 1), and the single element at position 2, creating the output "merge: 0 2".
28. There are eleven different front/back splits in this process: 0 5, 0 2, 0 1, 0 0, 1 1, 2 2, 3 5, 3 4, 3 3, 4 4, and 5 5.
29. Since the binary search is a O(log N) process, the number of searches is the log base 2 exponent number that creates a value that equals or just exceeds 2000, which 11. 2^11 is 2048, therefore it will take at most eleven search steps to find the **page** on which the word exists, or that the word is not found.
30. The original expression is postfix form. Choice I is the infix equivalent, and choice IV is the prefix equivalent.
31. Here is the recursive trace for this problem.




32. Since the values 9 and 10.0 are added mathematically, that result is 19.0, which is then concatenated with "11", resulting in "19.011".
33. The complete output order is: 0 1 2 3 5 7 7.
34. $\bar{A} \oplus B$ is the correct expression, with the bar over the A indicating NOT, and the circle-plus operator for XOR.
35. The output values for each pair are: `3 2==> CS, 1 7==> CS, 5 3==> STATE, 4 -3==> CS, 4 5==> UIL, -6 5==> STATE.`
36. The split pattern "`[aeiou]+`" means split on any sequence of those letters, wherever they occur in the string, in whatever order. Since the first split occurs at the `'e'`, there is an empty string at the front, which means position 1 contains "xtr".

37. $\overline{\overline{A} * B} * (\overline{B} + \overline{A})$ simplifies to $(A + \overline{B}) * (\overline{B} + \overline{A})$ using DeMorgan's law and the Double Negative law, then to $A\overline{B} + \overline{B} + \overline{A}\overline{B}$ (choice E) using FOIL and the complement law for AND, and finally to $\overline{B}$ using the absorption law.
38. Since both pointer expressions are referencing the value 5 at the end of the list, the sum of 5 and 5 is 10. **Note**: the position of p does not change in either expression.
39. Using the two's complement conversion process, 10101001 converts back to 01010111, which is the value 87, hence the original bit string is -87.
40. There are 21 different paths of length 3: WXZW, WXZX, WYYY, WYZX, WYYZ, WYZW, XZXZ, XZWY, XZWX, YYYY, YYYZ, YYZX, YZXZ, YZWX, YZWY, YYZW, ZXZX, ZXZW, ZWYZ, ZWYY, and ZWXZ. These can be found using the "brute force" method by careful examination, or using the matrix multiplication process as shown below.

Using the matrix multiplication method produces the following matrices:
Adjacency matrix M1
(7 paths of length 1)

```
  WXYZ
W 0110
X 0001
Y 0011
Z 1100
```

M1*M1 = M2 (12 paths of length 2)

```
  WXYZ
W 0012
X 1100
Y 1111
Z 0111
```

M1*M2 = M3 (21 paths of length 3)

```
  WXYZ
W 2211
X 0111
Y 1222
Z 1112
```