**QUESTION SET 2**

Solve the problems below.

Quickly scan the problems and solve the easiest ones first.

Then go back and keep sweeping this list until all problems are completed.

Take a **fullscreen desktop screenshot of your submission results**. The screenshot must show that you completed the problem 100% and Visual Studio. Name your .cpp file as “main\_[PROBLEM SET]\_[PROBLEM NUMBER].cpp”.

For example the solution to the first problem would be named: “main\_2\_1.cpp”.

More questions will be added to the bottom of this list. Please check the github link often.

NOTE: you MUST set the submission language to C (not C++) unless absolutely required by the problem (for example, OOP)

1. [~~https://www.hackerrank.com/challenges/pointer-in-c/problem~~](https://www.hackerrank.com/challenges/pointer-in-c/problem)
2. [~~https://www.hackerrank.com/challenges/bitwise-operators-in-c/problem~~](https://www.hackerrank.com/challenges/bitwise-operators-in-c/problem)
3. [~~https://www.hackerrank.com/challenges/printing-tokens-/problem~~](https://www.hackerrank.com/challenges/printing-tokens-/problem)
4. [~~https://www.hackerrank.com/challenges/frequency-of-digits-1/problem~~](https://www.hackerrank.com/challenges/frequency-of-digits-1/problem)
5. <https://www.hackerrank.com/challenges/dynamic-array-in-c/problem>
6. [~~https://www.hackerrank.com/challenges/recursion-in-c/problem~~](https://www.hackerrank.com/challenges/recursion-in-c/problem)
7. [~~https://www.hackerrank.com/challenges/variadic-functions-in-c/problem~~](https://www.hackerrank.com/challenges/variadic-functions-in-c/problem)
8. [~~https://www.hackerrank.com/challenges/c-tutorial-pointer/problem~~](https://www.hackerrank.com/challenges/c-tutorial-pointer/problem)
9. [~~https://www.hackerrank.com/challenges/variable-sized-arrays/problem~~](https://www.hackerrank.com/challenges/variable-sized-arrays/problem)
10. [~~https://www.hackerrank.com/challenges/c-tutorial-strings/problem~~](https://www.hackerrank.com/challenges/c-tutorial-strings/problem)
11. [~~https://www.hackerrank.com/challenges/c-tutorial-struct/problem~~](https://www.hackerrank.com/challenges/c-tutorial-struct/problem)
12. [~~https://www.hackerrank.com/challenges/c-tutorial-class/problem~~](https://www.hackerrank.com/challenges/c-tutorial-class/problem)
13. [~~https://www.hackerrank.com/challenges/classes-objects/problem~~](https://www.hackerrank.com/challenges/classes-objects/problem)
14. [~~https://www.hackerrank.com/challenges/box-it/problem~~](https://www.hackerrank.com/challenges/box-it/problem)
15. <https://www.hackerrank.com/challenges/inherited-code/problem>
16. [~~https://www.hackerrank.com/challenges/virtual-functions/problem~~](https://www.hackerrank.com/challenges/virtual-functions/problem)
17. <https://www.hackerrank.com/challenges/abstract-classes-polymorphism/problem>
18. [~~https://www.hackerrank.com/challenges/vector-sort/problem~~](https://www.hackerrank.com/challenges/vector-sort/problem)
19. [~~https://www.hackerrank.com/challenges/cpp-sets/problem~~](https://www.hackerrank.com/challenges/cpp-sets/problem)
20. [~~https://www.hackerrank.com/challenges/cpp-maps/problem~~](https://www.hackerrank.com/challenges/cpp-maps/problem)
21. [~~https://www.hackerrank.com/challenges/inheritance-introduction/problem~~](https://www.hackerrank.com/challenges/inheritance-introduction/problem)
22. <https://www.hackerrank.com/challenges/cpp-exception-handling/problem>
23. [~~https://www.hackerrank.com/challenges/accessing-inherited-functions/problem~~](https://www.hackerrank.com/challenges/accessing-inherited-functions/problem)
24. [~~https://www.hackerrank.com/challenges/c-class-templates/problem~~](https://www.hackerrank.com/challenges/c-class-templates/problem)
25. [~~https://www.hackerrank.com/challenges/preprocessor-solution/problem~~](https://www.hackerrank.com/challenges/preprocessor-solution/problem)
26. <https://www.hackerrank.com/challenges/operator-overloading/problem>
27. [~~https://www.hackerrank.com/challenges/overload-operators/problem~~](https://www.hackerrank.com/challenges/overload-operators/problem)
28. <https://www.hackerrank.com/challenges/cpp-class-template-specialization/problem>
29. [~~https://www.hackerrank.com/challenges/2d-array/problem~~](https://www.hackerrank.com/challenges/2d-array/problem)
30. <https://www.hackerrank.com/challenges/dynamic-array/problem>
31. [~~https://www.hackerrank.com/challenges/array-left-rotation/problem~~](https://www.hackerrank.com/challenges/array-left-rotation/problem)
32. [~~https://leetcode.com/problems/squares-of-a-sorted-array/~~](https://leetcode.com/problems/squares-of-a-sorted-array/)
33. <https://leetcode.com/problems/array-partition-i/>
34. [~~https://leetcode.com/problems/peak-index-in-a-mountain-array/~~](https://leetcode.com/problems/peak-index-in-a-mountain-array/)
35. [~~https://leetcode.com/problems/fibonacci-number/~~](https://leetcode.com/problems/fibonacci-number/)
36. <https://leetcode.com/problems/transpose-matrix/>
37. <https://leetcode.com/problems/letter-case-permutation/>
38. <https://leetcode.com/problems/hamming-distance/>
39. [~~https://leetcode.com/problems/add-digits/~~](https://leetcode.com/problems/add-digits/)
40. [~~https://leetcode.com/problems/roman-to-integer/~~](https://leetcode.com/problems/roman-to-integer/)
41. [~~https://leetcode.com/problems/rotate-string/~~](https://leetcode.com/problems/rotate-string/)
42. [~~https://www.codingame.com/training/easy/lumen~~](https://www.codingame.com/training/easy/lumen)
43. [~~https://www.codingame.com/ide/puzzle/ascii-art~~](https://www.codingame.com/ide/puzzle/ascii-art)
44. [~~https://www.codingame.com/ide/puzzle/chuck-norris~~](https://www.codingame.com/ide/puzzle/chuck-norris)
45. [~~https://www.codingame.com/ide/puzzle/horse-racing-duals~~](https://www.codingame.com/ide/puzzle/horse-racing-duals)
46. <https://www.hackerrank.com/challenges/balanced-brackets/problem>
47. [~~https://www.hackerrank.com/challenges/plus-minus/problem~~](https://www.hackerrank.com/challenges/plus-minus/problem)
48. [~~https://www.hackerrank.com/challenges/birthday-cake-candles/problem~~](https://www.hackerrank.com/challenges/birthday-cake-candles/problem)
49. [~~https://www.hackerrank.com/challenges/divisible-sum-pairs/problem~~](https://www.hackerrank.com/challenges/divisible-sum-pairs/problem)
50. [~~https://www.hackerrank.com/challenges/bon-appetit/problem~~](https://www.hackerrank.com/challenges/bon-appetit/problem)
51. <https://www.hackerrank.com/challenges/circular-array-rotation/problem>
52. [~~https://www.hackerrank.com/challenges/small-triangles-large-triangles/problem~~](https://www.hackerrank.com/challenges/small-triangles-large-triangles/problem)
53. <https://www.hackerrank.com/challenges/frequency-of-digits-1/problem>
54. [~~https://www.hackerrank.com/challenges/time-conversion/problem~~](https://www.hackerrank.com/challenges/time-conversion/problem)

* 15. <https://www.hackerrank.com/challenges/inherited-code/problem>
  + Something about ‘throw’ and ‘catch’ but we’ve never used them. Exception is what VS says when you pass an incompatible value and crash so it can be fixed isn’t it
* 17. <https://www.hackerrank.com/challenges/abstract-classes-polymorphism/problem>
  + Don’t understand how the Node struct works
* 22. <https://www.hackerrank.com/challenges/cpp-exception-handling/problem>
  + Something about ‘throw’ and ‘catch’ but we’ve never used them. Exception is what VS says when you pass an incompatible value and crash so it can be fixed isn’t it
* 26. <https://www.hackerrank.com/challenges/operator-overloading/problem>
  + My attempt times out. The solution found in the discussion board seems similar but modifies values of one of the objects being added and passes a pointer to the modified object. that seems aberrant for ‘+’ and reckless to change your input if you intend to output the change
* 28. <https://www.hackerrank.com/challenges/cpp-class-template-specialization/problem>
  + Rather vague on non class templates but the confounding part is how enum is supposed to work
* 30. <https://www.hackerrank.com/challenges/dynamic-array/problem>
  + I fail to understand what the queries are doing but it mentions ‘^’ as a bitwise operator
* 33. <https://leetcode.com/problems/array-partition-i/>
  + I fail to understand what formula this is applying. I think the example is 4 because 1+3 but I don’t know why 1 paired with 2 and 3 paired with 4
* 36. <https://leetcode.com/problems/transpose-matrix/>
  + RTE heap buffer overflow, why is returnColumnSizes double pointer when it’s a single value. Both return sizes are a pointer more than A sizes
* 37.<https://leetcode.com/problems/letter-case-permutation/>
  + RTE global buffer overflow-don’t understand
* 38.<https://leetcode.com/problems/hamming-distance/>
  + Don’t know how to do it. All examples found use bitwise operators
* 46. <https://www.hackerrank.com/challenges/balanced-brackets/problem>
  + Don’t know why results won’t pass even though malloc
* 51. <https://www.hackerrank.com/challenges/circular-array-rotation/problem>
  + Don’t know why results won’t pass even though malloc
* 53. <https://www.hackerrank.com/challenges/frequency-of-digits-1/problem>
  + Duplicate of problem 4