1. D → while (<inside of this boolean condition>)
2. B → traditional for loop is for using index based and counter.
3. A → do-while loop is known that guarantes the at least once entrance and execution
4. C → for-each mechanism is to loop array without referring to element index. Because it runs element-wise
5. A → <<break>> this keyword is used to end loop iteration. Continue is skipping the current execution.
6. A → this keyword is used after a loop. İt means that it was completed then keep going.
7. B → for (traditional) has three segments (first;second;third)
8. C → it depends on the first index and increments or decrements of iterate
9. D → actually for-each statements start with 1 index so we cannot change it instead we can use new variable for fix this problem.
10. A → after a single iteration through the loop only and only is do-while loop
11. B → This code does not compile. Because condition of while loop should be boolean value. But we gave the int value so it gives us compilation error.
12. A → when we look at the code it starts with size minus one. So it means that starting with 1 and iterate it till it equals 0.
13. D→ In this for loop there is no restriction of loop. So it will execute as unstopped
14. A → This code run 2 times. Because first it is empty and has 0 lenght. Then when it entrance to first iteration it will be 1 and 2 respectively. So outcome of code will be aa
15. D → when we run this piece of code it will outcome that is infinite args… . Because there is no restriction inside of code
16. B → the result will be 2. When app starts to run count will be 0 reason why it takes 0 automatically when it is defined as static. Stop lenght is 4. Stops[0] is Washington and its length is 10. So it provides while but if condition.inside of if there is an incrementation operations. Then count will be 1. And in second iteration it will monreo and provides if condition and will take the break so end the end of output count should be 2.
17. C → the variable count is defined inside of loop. Since it is not a global variable it cannot use out of loop
18. D → All of the above. We can define for loop as `for (;;)`
19. C → we cannot use for-each loop for infinite loop
20. A → output of code pieces is can, cup,
21. B → it outputs helium once. Because it is prop of do-while loop that does at least once inside of do.
22. D → since for-each starts with index of 1 then none of the above can output the same result
23.
24.
25. C → The loops completes with no output. Because this conditions cannot be provided
26.
27. B → inflate – done. Because do Works at least once. And provides if condition.
28. C → first step length is 0 then it will be 2 times 2 every iteration and never be 3 . so there is infinite loop in this code
29. B → initialization expression, boolean conditional, update statement
30. B → 4. This is there are 2 loops. So iteratively it will be 4.
31. A → normally i decreases one by one . but when it enters the while loop then it decreases 3 by 3.
32. D → none of the above
33. C → this code does not compile. Because break keywords should be inside of loop. But at this code it is out of loop.

34. C → There is a syntax error Because after i++ it should be comma
35. D → Node of the above: Because length of array is 3. And if we try to select index of length it throws out of bound exception
36. B → tie is assigned to null then it goes to while loop once
37. C → there is an error because of missing loop statement
38. C → the result of this code is 4
39. C → this code cannot compile. Because while loop conditions should be boolean but in this code we gives a string
40. A → output of this code is 2. Because once it enters to do condition then it welcomes by second do condition and inside of second condition it will be 1 after while loop it will be 2
41. C → break t easly breaks all loops
42. B → output of this code is “Downtown Day – Uptown Night”
43. Output of this code 2x2 = 4 lines like a matrix
44. A → First alpha runs.second beta runs.then then according to beta condition delta runs.
45. B → alpha, beta, delta, gamma, beta
46. C→ because c is run 6 times, A: 5 times, B: 5 times, C: 6 times, D: 5 times
47. D → None of the above : Because tie assigments is not inside of loop
48. C → this code cannot compile since we cannot use keywords as a label.
49. D → this code is an infinite loop. Because it will be allways true
50. B → when we write the int i = 0, j = 0; then this code can be compiled