**PPS Assignment 0**

Name: Aryan Saxena

Student ID: 40233170

**Question 1**

**A.**

1. Input an Array of size n
2. Input the target integer
3. TRAVERSE through the Array from (i = 0 to n)
4. IF Array [i] is equal to the target and i is not the last value in the Array
5. Array [i] = Array [i] + Array [i+1]
6. Array [i+1] = Array [i]-Array [i+1];
7. Array [i] = Array [i] – Array [i+1];
8. Increase (i by 1)
9. Print the Array

**B. Time Complexity**: O(n)

**C. Space Complexity:**  O(n)

**Question 2**

A.

1. Take INPUT in String using (sc.nextLine())
2. Create an Array with the same length and split each Character
3. Create an Array List<Character> and String answer
4. TRAVERSE(Outer loop) through the Array from (i=0 to Array.length)
5. Make a count = 0;
6. TRAVERSE(Inner-Loop) through the Array again from (j = 0 to Array.length)
7. IF (Array [i] == Array [j] && i != j)
8. Count++
9. Array[j] = 0;
10. Outside the Traverse (2)
11. IF Count>0
12. Then add Array[i] to the ArrayList
13. Array[i] = 0;
14. TRAVERSE through the Array from (i=0 to Array.length)
15. If (Array[i] == Vowel)
16. Then add the Array[i] to the ArrayList
17. Array[i] = 0;
18. TRAVERSE through the Array from (I = 0 to Array.length)
19. If (Array[i] != 0)
20. answer = answer + Array[i]
21. TRAVERSE through the ArrayList from (I = 0 to ArrayList.size)
22. If (ArrayList(i) != null)
23. answer = answer + ArrayList(i)
24. Print answer

**B. Time Complexity**: O(n2)

**C. Space Complexity:** O(n)

**Question 3**

1. Take INPUT as Integer Array
2. TRAVERSE through the Array using FOR (i = 0 to Array.length)
3. Take reverse of Array[i] in a variable REV
4. IF (Array[i] != REV) then break
5. IF (Array[i] != 1 or 0 or 8) then break
6. TRAVERSE through the Array using FOR(j=0 to Array.length)
7. IF (i != j)
8. Array[j] in REV
9. IF (Array[j] != REV) then break
10. IF (Array[j] != 1 or 0 or 8) then break
11. Initialize Integer I1,I2,I3,I4 with value 0
12. TEMP = Array[i] – Array[j];
13. IF (TEMP == 10)
14. DO ( IF ( |i - j| > | I1 – I2 |)
15. DO( I1 = i, I2 = j)
16. PRINT ( I1 Array[I1] I2 Array[I1] )
17. TEMP2 = 0
18. TRAVERSE through the Array using FOR (I = 0 to (Array.Length-1))
19. I3 = i , I4 = i + 1
20. PRINT I3 Array[I3] I4 Array[I4]
21. Tetradic Numbers are those numbers that only contain 0, 1 or 8 and are palindromes.

My code will input the numbers to an Integer Array, and then will traverse through every element of the Integer Array. It will take the reverse of the element, if the reverse is not equal to the element itself then the code will break out of the loop. Otherwise, If the element is not 0,1 or 8, then it would break.

Further, the code will traverse the Array. If index i is not equal to index j, Array element with index j is checked for both reverse and 0,1 and 8, if they pass the loop would run otherwise it will break. We created 4 indexes i1,i2,i3,i4. A temporary variable will contain Array[i] – Array[j], if the temporary variable will be equal to 10 then it will check if the absolute value of (i – j) > (I1 – I2), if that passes i1 to I and i2 to j. Then the code will print index i1 and Array[i1], index i2 and Array[i2]. Temporary2 variable will be initialized with 0,. Again traverse through the Array using for till the end. Then I3 = i, I4 = i +1. And print index i3 and Array[i3], index i4 and Array [i4].

1. **Time Complexity :** O(n2)

Since, two nested for loops running and each FOR loop iterates from 1 to N, the overall complexity of the code will be O(n2)

1. **Stack Growth :** O(n)

At any given point, at most the space consumed by the code will be (n) due to N elements.