Solution @07-08-23

1. Merge Sorted Array

```
// Java program to merge two sorted arrays
import java.util.*;
import java.lang.*;
import java.io.*;
class MergeTwoSorted{
    // function to merge arrays
    public static void mergeArrays(int[] nums1, int[] nums2, int m,int n, int[]
nums3){
        // copy nums1[] elements to nums3[]
        for(int i=0;i<m;i++)
            nums3[i]=nums1[i];
        // apply insertion sort algorithm to insert nums2[] elements to nums1[]
        for(int i=0;i<n;i++) {
            int temp=nums2[i];
            int j=m-1;
            for(;j>=0;j--){
                //move elments one position ahead that are greater than current
value
                if(nums3[j]>temp){
                    nums3[j+1]=nums3[j];
                }
                else
                    break;
            }
            m=m+1;
            //put Current element at its correct position.
            nums3[j+1]=temp;
        }
    }
    // driver code
    public static void main (String[] args){
        int[] nums1 = {1, 3, 5, 7};
        int m = nums1.length;
        int[] nums2 = {2, 4, 6, 8};
        int n = nums2.length;
        int[] nums3 = new int[m+n];
        // calling function to merge two sorted arrays
        mergeArrays(nums1, nums2, m, n, nums3);
        // printing the resultant sorted array
        System.out.println("Array after merging");
        for (int i=0; i < m+n; i++)
            System.out.print(nums3[i] + " ");
    }
}
```

2.Reverse Interger

```
class Solution {
    public int reverse(int x) {
        if (x == 0){ //if the number is zero then reverse will be zero
            return 0;
        }
        else{
            int flag = 1;
            if (x < 0){
                flag = -1;
                x = -x;
            }
            int result = 0;
            while(x > 0){
                int remaining_digit = x % 10;
                int newresult = result * 10 + remaining_digit;
                if (result != (newresult - remaining_digit) / 10){
                    result = 0;
                    break;
                }
                result = newresult;
                x = x / 10;
            }
            result = result * flag;
            return result;
        }
    }
}
```