Add one to number

Problem: https://www.interviewbit.com/problems/add-one-to-number/

Given a non-negative number represented as an array of digits, add 1 to the number (increment the number represented by the digits). The digits are stored such that the most significant digit is first element of array.

Example:
If the vector has `[1, 2, 3]`
the returned vector should be `[1, 2, 4]`
as `123 + 1 = 124`.

Wrong Approach:

To convert the list elements to string then convert to integer, add 1, and then assign the values to different list as output.

```
1.  public class Solution {
2.      public ArrayList<Integer> plusOne(ArrayList<Integer> A) {
3.
4.          int i;
5.          String array_as_string="";
6.          ArrayList<Integer> output = new ArrayList<>();
7.          boolean check_for_zeros_at_start = false;
8.
9.          for(i=0;i<A.size();i++)
10.         {
11.             if(A.get(i)>0)
12.             check_for_zeros_at_start = true;
13.
14.
15.
16.             if(check_for_zeros_at_start)
17.             array_as_string+=A.get(i);
18.         }
19.
20.         if(array_as_string.length()==0)
21.         {
22.             output.add(1);
23.             return output;
24.         }
25.
26.         long num = Long.parseLong(array_as_string);
27.         num+=1;
28.
```

```
        long dum_num = num;
        int length = A.size()-1;

        while(dum_num>0)
        {
            int remainder = (int)(dum_num%10);
            output.add(remainder);
            dum_num/=10;
        }

        Collections.reverse(output); //reversing the list
        return output;
    }

}
```

Right Approach: To add one to number represented by digits, follow the below steps :

- Parse the given array from end like we do in school addition.
- If the last elements 9, make it 0 and carry = 1.
- For the next iteration check carry and if it adds to 10, do same as step 2.
- After adding carry, make carry = 0 for the next iteration.
- If the vectors add and increase the vector size, append 1 in the beginning.

```
public class Solution {
    public ArrayList<Integer> plusOne(ArrayList<Integer> A) {

        int length = A.size();
        int carry;
        int i;

        A.set(length-1,A.get(length-1)+1);
        carry = A.get(length-1)/10;
        A.set(length-1,A.get(length-1)%10);

        for(i=length-2;i>=0;i--)
        {
            if(carry==1)
            {
                A.set(i,A.get(i)+1);
                carry = A.get(i)/10;
                A.set(i,A.get(i)%10);
            }

            else
            break;
```

```
        }
//To handle the carry at the index 0 element
//for eg: 999
        if(carry==1)
        A.add(0,1); //Inserts value at the beginning

        for(i=0; i<length; i++) //To remove the 0's before the MSB.
        {
            if(A.get(i)==0)
            {
            A.remove(i);

//When the element is removed the size decreases and the next element is located
  on current index now.
//Inorder to not skip the next element we decrement our index.
            i--;
            }
            else
            break;
        }

        if(A.size()==0) //If the given list contains only 0's we return 1.
        {
             A.add(0,1);
        }
        return A;

    }

}
```