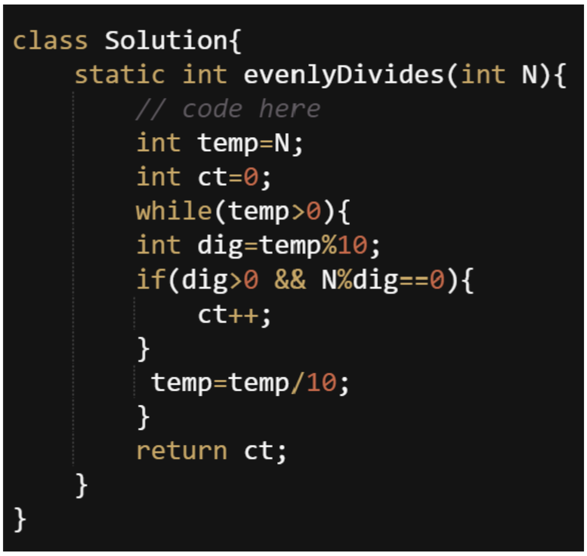
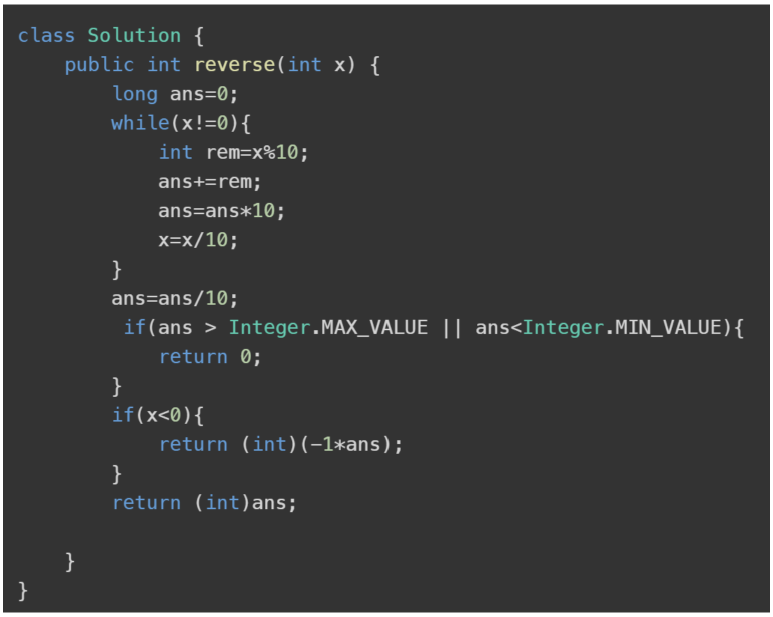
DIGIT TRAVERSALS

Evenly divides:- https://www.geeksforgeeks.org/problems/count-digits5716/1



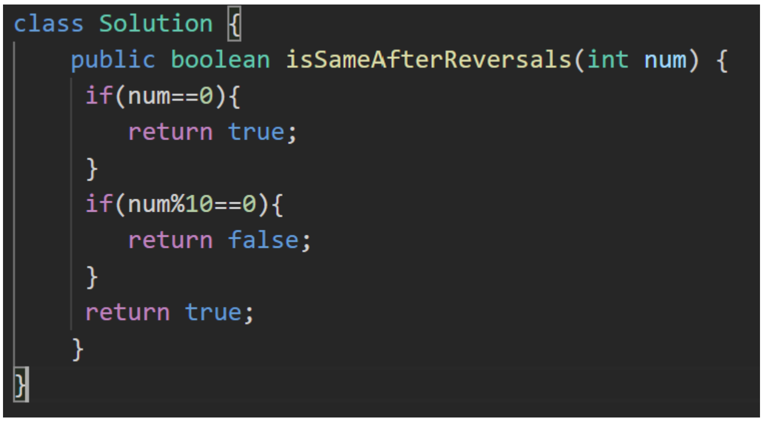
```
class Solution{
    static int evenlyDivides(int N){
        // code here
        int temp=N;
        int ct=0;
        while(temp>0){
        int dig=temp%10;
        if(dig>0 && N%dig==0){
            ct++;
        }
         temp=temp/10;
        }
        return ct;
    }
}
```

Reverse Integer :- https://leetcode.com/problems/reverse-integer/description/



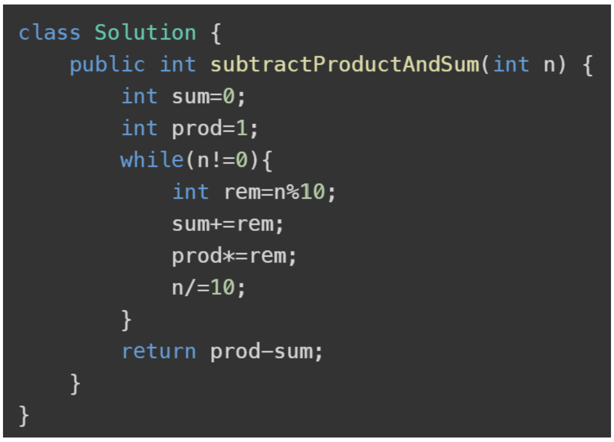
```
class Solution {
    public int reverse(int x) {
        long ans=0;
        while(x!=0){
            int rem=x%10;
            ans+=rem;
            ans=ans*10;
            x=x/10;
        }
        ans=ans/10;
         if(ans > Integer.MAX_VALUE || ans<Integer.MIN_VALUE){
            return 0;
        }
        if(x<0){
            return (int)(-1*ans);
        }
        return (int)ans;

    }
}
```

Double Reversal:- https://leetcode.com/problems/a-number-after-a-double-reversal/description/



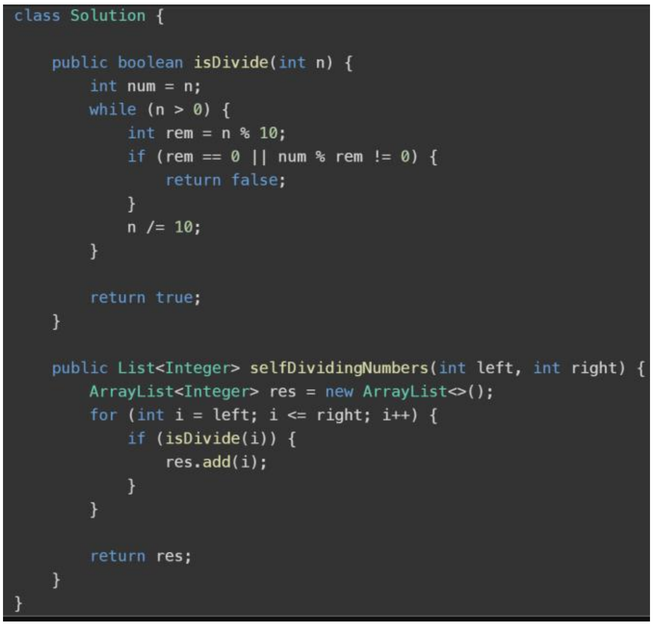
```
class Solution {
    public boolean isSameAfterReversals(int num) {
     if(num==0){
        return true;
     }
     if(num%10==0){
        return false;
     }
     return true;
    }
}
```

Subtract prod and sum :- https://leetcode.com/problems/subtract-the-product-and-sum-of-digits-of-an-integer/description/



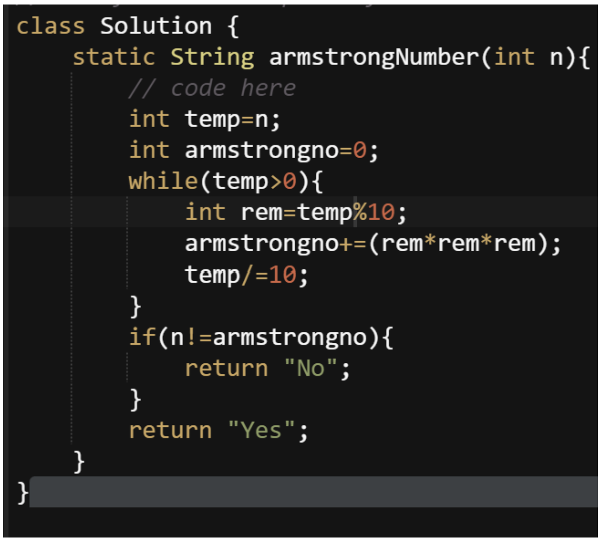
```java
class Solution {
    public int subtractProductAndSum(int n) {
        int sum=0;
        int prod=1;
        while(n!=0){
            int rem=n%10;
            sum+=rem;
            prod*=rem;
            n/=10;
        }
        return prod-sum;
    }
}
```

Self Dividing Numbers:- https://leetcode.com/problems/self-dividing-numbers/description/



```java
class Solution {

    public boolean isDivide(int n) {
        int num = n;
        while (n > 0) {
            int rem = n % 10;
            if (rem == 0 || num % rem != 0) {
                return false;
            }
            n /= 10;
        }

        return true;
    }

    public List<Integer> selfDividingNumbers(int left, int right) {
        ArrayList<Integer> res = new ArrayList<>();
        for (int i = left; i <= right; i++) {
            if (isDivide(i)) {
                res.add(i);
            }
        }

        return res;
    }
}
```

Armstrong No:- https://www.geeksforgeeks.org/problems/armstrong-numbers2727/1



```java
class Solution {
    static String armstrongNumber(int n){
        // code here
        int temp=n;
        int armstrongno=0;
        while(temp>0){
            int rem=temp%10;
            armstrongno+=(rem*rem*rem);
            temp/=10;
        }
        if(n!=armstrongno){
            return "No";
        }
        return "Yes";
    }
}
```