

```
public class DuplicateNumver {  
    public int solution(int X){  
        int max=Integer.MIN_VALUE;  
        StringBuilder builder=new StringBuilder(String.valueOf(X));  
        for(int i=1;i<builder.length();i++){  
            if(builder.charAt(i-1)==builder.charAt(i)){  
                StringBuilder b=new StringBuilder(builder.toString());  
                b.deleteCharAt(i);  
                int a=Integer.valueOf(b.toString());  
                if(a>max){  
                    max=a;  
                }  
            }  
        }  
        return max;  
    }  
}
```

```

import java.util.Stack;

public class ImageDirectory {

    int sum;
    int numberOfimg;

    public int solution(String S) {
        sum = 0;
        numberOfimg=0;
        String[] ss = S.split("\n");
        helper(ss);
        return sum==0?numberOfimg:sum;
    }

    public void helper(String[] ss) {
        HashSet<String> set = new HashSet();
        StringBuilder b = new StringBuilder();
        Stack<Integer> st = new Stack();
        int index = 0;
        boolean[] visit = new boolean[ss.length];
        int num = 0;

        while (index < ss.length) {
            st.push(index++);

            while (!st.empty()) {
                int s = st.peek();
                if (!visit[s]) {
                    visit[s] = true;
                    if (!ss[s].contains(".")) {
                        num = num + 1 + ss[s].trim().length();
                        b.append('/');
                        b.append(ss[s].trim());
                    }
                }
            }
            if (check(ss[s]) && set.add(b.toString())) {
                numberOfimg++;
                sum = sum + num;
            }

            if (index < ss.length && isNextLev(ss[index], ss[s])) {
                st.push(index++);
            } else {
                int s1 = st.pop();
                if (!ss[s1].contains(".")) {
                    b.delete(b.length()-ss[s1].trim().length()-1, b.length());
                    num = num - 1 - ss[s1].trim().length();
                }
            }
        }
    }
}

```

```

public boolean isNextLev(String s1, String s) {
    int a = 0;
    int b = 0;
    for (int i = 0; i < s.length(); i++) {
        if (s.charAt(i) != ' ') {
            a = i;
            break;
        }
    }
    for (int j = 0; j < s1.length(); j++) {
        if (s1.charAt(j) != ' ') {
            b = j;
            break;
        }
    }
    return b - a == 1;
}

```

```

public boolean check(String s) {
    String s2 = s.trim();
    int a = 0;
    for (int i = 0; i < s2.length(); i++) {
        if (s2.charAt(i) == '.') {
            a = i;
            break;
        }
    }
    String s1 = s2.substring(a);
    return s1.equals(".jpeg") || s1.equals(".gif") || s1.equals(".png");
}

```