

```

public class Solution {

    public static void main(String[] args) {
        System.out.println(new Solution().run());
    }

    private static final String NUMBER =
"7316717653133062491922511967442657474235534919493496983520312774506326239578
3180169848018694788518438586156078911294949545950173795833195285320880551112
5406987471585238630507156932909632952274430435576689664895044524452316173185
6403098711121722383113622298934233803081353362766142828064444866452387493035
8907296290491560440772390713810515859307960866701724271218839987979087922749
2190169972088809377665727333001053367881220235421809751254540594752243525849
0771167055601360483958644670632441572215539753697817977846174064955149290862
5693219784686224828397224137565705605749026140797296865241453510047482166370
4844031998900088952434506585412275886668811642717147992444292823086346567481
3919123162824586178664583591245665294765456828489128831426076900422421902267
1055626321111109370544217506941658960408071984038509624554443629812309878799
2724428490918884580156166097919133875499200524063689912560717606058861164671
0940507754100225698315520005593572972571636269561882670428252483600823257530
420752963450";

        private static final int ADJACENT = 13;

        public String run() {
            long maxProd = -1;
            for (int i = 0; i + ADJACENT <= NUMBER.length(); i++) {
                long prod = 1;
                for (int j = 0; j < ADJACENT; j++)
                    prod *= NUMBER.charAt(i + j) - '0';
                maxProd = Math.max(prod, maxProd);
            }
            return Long.toString(maxProd);
        }
    }
}

```