

# It's Only X \* Y

Problem

Submissions

Leaderboard

Discussions

Author : niko.ac.id

Di dalam sebuah program, time complexity perkalian brute force jauh lebih buruk dibanding dengan divide and conquer dengan angka besar. Dari algoritma yang diambil dari ppt dibawah buat kan kodenya, beserta print 3 rekursif utama secara urut berdasarkan 3 rekursif hint dibawah!

- 3 Urutan rekursif Hint = (a,c,s),(b+a,c+d,s),(b,d,s)
- Hint = Lihat PPT ASA DnC
- Hint = Kode jawaban berdasarkan **pseudo code dibawah** tapi **perlu diubah** dikit doang (NEW)
- Hint = String n diambil yang terpendek
- Hint = Sudah di test pake Java + Python aman

```
function Kali2(input X, Y : LongInteger, n : integer) → LongInteger
{ Mengalikan X dan Y, masing-masing panjangnya n digit dengan algoritma
  Divide and Conquer.
  Masukan: X dan Y
  Keluaran: hasil perkalian X dan Y
}
Deklarasi
  a, b, c, d : LongInteger
  s : integer

Algoritma:
  if n = 1 then
    return X * Y      { perkalian biasa }
  else
    s ← n div 2      { bagidua pada posisi s }
    a ← X div 10s
    b ← X mod 10s
    c ← Y div 10s
    d ← Y mod 10s
    return Kali2(a, c, s)*102s + Kali2(b, c, s)*10s +
           Kali2(a, d, s)*10s + Kali2(b, d, s)
  endif
```

## Input Format

- baris pertama input angka pertama
- baris kedua input angka kedua

## Constraints

- $-10^{15} \leq x \leq 10^{15}$

## Output Format

- ### Sample Input 0

### Sample Output 0

### Sample Input 1

### Sample Output 1

More



```
1 import java.math.BigInteger;
2 import java.util.Scanner;
3
4 public class Main {
5
6     // fungsi untuk mengalikan dua bilangan dengan algoritma Divide and Conquer
7     public static BigInteger Kali2(BigInteger X, BigInteger Y, int n) {
8         if (n == 1) {
9             return X.multiply(Y);
10        }
11        else {
```

```

12         int s = n / 2;
13         BigInteger a = X.divide(BigInteger.TEN.pow(s)); // bagi X menjadi a dan b
14         BigInteger b = X.mod(BigInteger.TEN.pow(s));
15         BigInteger c = Y.divide(BigInteger.TEN.pow(s)); // bagi Y menjadi c dan d
16         BigInteger d = Y.mod(BigInteger.TEN.pow(s));
17         BigInteger p = Kali2(a, c, s); // hitung a*c
18         BigInteger q = Kali2(b, d, s);
19         BigInteger r = Kali2(a.add(b), c.add(d), s);
20         System.out.println(p + " " + r + " " + q); // print semua nilai rekursif
21         BigInteger temp = r.subtract(p).subtract(q);
22         BigInteger temp2 = temp.multiply(BigInteger.TEN.pow(s));
23         return p.multiply(BigInteger.TEN.pow(2 * s)).add(temp2).add(q); // hitung hasil
    perkalian
24     }
25 }
26
27 public static void main(String[] args) {
28     Scanner input = new Scanner(System.in);
29     String X = input.next();
30     String Y = input.next();
31     int n = Math.min(X.length(), Y.length());
32     X = String.format("%" + n + "s", X).replace(' ', '0');
33     Y = String.format("%" + n + "s", Y).replace(' ', '0');
34     BigInteger x = new BigInteger(X);
35     BigInteger y = new BigInteger(Y);
36     BigInteger hasil = Kali2(x, y, n);
37     System.out.println(hasil); // print hasil perkalian
38     input.close();
39 }
40 }

```

Line: 1 Col: 1

 [Upload Code as File](#) ☐ [Test against custom input](#)

[Run Code](#)

[Submit Code](#)