

题目描述：

给定字符串 A 、 B 和正整数 V ， A 的长度与 B 的长度相等，请计算 A 中满足如下条件的最大连续子串的长度：

- 1、该连续子串在 A 和 B 中的位置和长度均相同。
- 2、该连续子串 $|A[i] - B[i]|$ 之和小于等于 V 。其中 $|A[i] - B[i]|$ 表示两个字母 $ASCII$ 码之差的绝对值。

输入描述：

输入为三行：

第一行为字符串 A ，仅包含小写字符， $1 \leq A.length \leq 1000$ 。

第二行为字符串 B ，仅包含小写字符， $1 \leq B.length \leq 1000$ 。

第三行为正整数 V ， $0 \leq V \leq 10000$ 。

输出描述：

字符串最大连续子串的长度，要求该子串 $|A[i] - B[i]|$ 之和小于等于 V 。

示例 1

输入：

```
xxcdefg  
cdefghi  
5
```

输出：

```
2
```

说明：

字符串 A 为 $xxcdefg$ ，字符串 B 为 $cdefghi$ ， $V=5$ 。

它的最大连续子串可以是 $cd \rightarrow ef, de \rightarrow fg, ef \rightarrow gh, fg \rightarrow hi$ ，所以最大连续子串是 2 。

```

import java.io.BufferedReader;
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(new BufferedReader(System.in));
        String strA = scanner.nextLine();
        String strB = scanner.nextLine();
        int v = scanner.nextInt();

        int n = strA.length();
        int[] diff = new int[n];
        for (int i = 0; i < n; i++)
            diff[i] = Math.abs(strA.charAt(i) - strB.charAt(i));

        int sum = 0;
        int ans = 0;
        int left = 0;
        int right = 0;
        while (right < n) {
            sum += diff[right];
            right++;

            while (sum > v) {
                sum -= diff[left];
                left++;
            }

            ans = Math.max(ans, right - left);
        }

        System.out.println(ans);
    }
}

```