

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，仅包含小写字母， $1 \leq A.length \leq 1000$ 。

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

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。

补充说明：

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->ef,de->fg,ef->gh,fg->hi，所以最大连续子串是 2。

```
import java.util.*;
public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        String strA = scanner.nextLine();
        String strB = scanner.nextLine();
        int m = scanner.nextInt();

        int size = strA.length();
        int[] dif = new int[size];
        for (int i = 0; i < size; i++) {
            dif[i] = Math.abs(strA.charAt(i) - strB.charAt(i));
        }
    }
}
```

```

boolean flag = false;
int[] pre = new int[size];
pre[0] = dif[0];
if (pre[0] <= m) {
    flag = true;
}
for (int i = 1; i < size; i++) {
    pre[i] = pre[i - 1] + dif[i];
    if (dif[i] <= m) {
        flag = true;
    }
}
if (!flag) {
    System.out.println("0");
    return;
}

int max = 0;
for (int i = 0; i < size; i++) {
    for (int j = size - 1; j >= i; j--) {
        int sum = pre[j] - pre[i] + dif[i];
        if (sum <= m) {
            max = Math.max(max, j - i + 1);
            break;
        }
    }
}
System.out.println(max);
}
}

```