

字符串比较

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。

```
#include<iostream>
```

```
#include<vector>
```

```
#include<string>
```

```
using namespace std;
```

```
int main() {  
    string a;
```

```

string b;
cin >> a;
cin >> b;
int size = a.size();
int v;
cin >> v;
vector<int> sub(size);
for (int i{ 0 }; i < size; i++) {
    int temp = a[i] - b[i];
    if (temp < 0) {
        temp = -temp;
    }
    sub[i] = temp;
}
int left{ 0 };
int right{ 0 };
int cur{ 0 };
int ans{ 0 };
while (right < size) {
    if (cur <= v) {
        cur += sub[right];
        right++;
        if (cur <= v) {
            ans = ans > right - left ? ans : right - left;
        }
    }
    else {
        cur -= sub[left];
        left++;
    }
}
cout << ans;
}

```