

### 最长重复子序列

时间限制：C/C++语言 1000MS；其他语言 3000MS

内存限制：C/C++语言 65536KB；其他语言 589824KB

题目描述：

输入两个整数序列A和B，输出同时在A，B中出现的最长子序列的长度。注意，子序列由原序列中的连续元素构成。

#### 输入

第一行一个数n，表示序列A的长度  
第二行n个数，表示序列A  
第三行一个数m，表示序列B的长度  
第四行m个数，表示序列B  
( $1 \leq n, m \leq 1000$ )

#### 输出

输出结果

#### 样例输入

5  
1 2 3 2 1  
5  
3 2 1 4 7

#### 样例输出

3

### 样例输入

```
5
1 2 3 2 1
5
3 2 1 4 7
```

### 样例输出

```
3
```

### 提示

即最长重复子列为[3,2,1]

### 规则

请尽量在全场考试结束10分钟前调试程序，否则由于密集排队提交，可能查询不到编译结果  
点击“调试”亦可保存代码  
编程题可以使用本地编译器，此页面不记录跳出次数

```
#include<iostream>
#include<algorithm>
#include"stdio.h"
#include<cstring>
#include<queue>
#include<vector>
#include<cmath>
#include<stack>
#include<deque>
#include<string>
#include <sstream>
#include <unordered_map>

using namespace std;

const int MAX_N = 1001;
const int MAX_M = 1001;

int dp[MAX_N][MAX_M];
char str1[MAX_N];
char str2[MAX_M];

int main()
{
    int n, m;

    cin >> n;
    for(int i = 0; i < n; i++)
        cin >> str1[i];

    cin >> m;
    for(int i = 0; i < m; i++)
        cin >> str2[i];
```

```
memset(dp, 0, sizeof(dp));

int res = 0;

for(int i = 0; i < n; i++)
    for(int j = 0; j < m; j++)
    {
        if(str1[i] == str2[j])
        {
            dp[i+1][j+1] = dp[i][j] + 1;
            res = max(res, dp[i+1][j+1]);
        }
    }

cout << res << endl;

return 0;
}
```

## 外卖小哥的保温箱

时间限制: C/C++语言 1000MS; 其他语言 3000MS

内存限制: C/C++语言 65536KB; 其他语言 589824KB

题目描述:

众所周知,美团外卖的口号是:"美团外卖,送啥都快"。身着黄色工作服的骑手作为外卖业务中商家和客户的重要纽带,在工作中,以快速送餐突出业务能力;工作之余,他们会通过玩智力游戏消遣闲暇时光,以反应速度彰显智慧,每位骑手拿出装有货物的保温箱,参赛选手需在最短的时间内用最少的保温箱将货物装好。

我们把问题简单描述一下:

1 每个货物占用空间都一模一样

2 外卖小哥保温箱的最大容量是不一样的,每个保温箱由两个值描述:保温箱的最大容量  $b_i$ ,当前已有货物个数  $a_i$  ( $a_i \leq b_i$ )

3 货物转移的时候,不必一次性全部转移,每转移一件货物需要花费 1秒 的时间

### 输入

第一行包含 一个整数  $n$ , ( $1 \leq n \leq 100$ ) 表示保温箱的数量

第二行有  $n$  个正整数  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 100$ )

$a_i$ 表示第  $i$ 个保温箱的已有货物个数

第三行有  $n$  个正整数  $b_1, b_2, \dots, b_n$  ( $1 \leq b_i \leq 100$ ),  $b_i$  表示第  $i$  个保温箱的最大容量

显然,每一个  $a_i \leq b_i$

### 输出

输出为两个整数  $k$  和  $t$ ,  $k$  表示能容纳所有货物的保温箱的最少个数,  $t$  表示将所有货物转移到这  $k$  个保温箱所花费的最少时间,单位为秒。

### 样例输入

```
4
3 3 4 3
4 7 6 5
```

### 样例输入

```
4
3 3 4 3
4 7 6 5
```

### 样例输出

```
2 6
```

### 提示

在样例一中,我们可以把第一个保温箱中的货物全部挪到第二个保温箱中,花费时间为 3 秒,此时第二个保温箱剩余容量为 1,然后把第四个保温箱中的货物转移一份到第二个保温箱中,转移最后两份到第三个保温箱中,总花费时间也是 3 秒,所以最少保温箱个数是 2,最少花费时间为 6 秒

### 规则

```

#include<stdio.h>
#include<algorithm>
#include<string>
#include<iostream>
#include<string.h>
using namespace std;
#define inf 1e7
int a[105],b[105];
int dp[10005][105];

/*
4
3 3 4 3
4 7 6 5
*/

int main(){
    int n;
    while(scanf("%d",&n)!=EOF){
        int sum = 0;
        for(int i=0;i<n;i++){
            scanf("%d",&a[i]);
            sum+=a[i];
        }
        for(int i=0;i<n;i++)scanf("%d",&b[i]);

        for(int i=0;i<=10000;i++){
            for(int j=0;j<=n;j++){
                dp[i][j]=-1;
            }
        }
        dp[0][0] = 0;
        int min_k = 1000;
        for(int i=0;i<n;i++){

            for(int j=sum+105;j>=b[i];j--){
                for(int k=n;k>0;k--){
                    if(dp[j-b[i]][k-1]!=-1){
                        dp[j][k] = max(dp[j][k], dp[j-b[i]][k-1]+a[i]);
                        if (j>=sum)
                            min_k = min(min_k, k);
                    }
                }
            }

            int min_c = 0;
            for(int j=sum;j<=sum+105;j++){
                if(dp[j][min_k]!=-1){
                    min_c = max(min_c, dp[j][min_k]);
                }
            }
            printf("%d %d\n", min_k, sum-min_c);
        }

        return 0;
    }
}

```

