

第一部分

主讲人：计试2101杨思成

1.输出索引

键盘输入一个数组，找到两个总和为特定值的索引。
例如给定数组 [1, 2, 3, -2, 5, 7]，给定总和 7，则返回索引 [1, 4]。
若有多组符合情况则输出索引对中小索引最小的任意一组。

样例

输入：第一个数是数量，第二行是数组元素输入，第三行是给定总和

6
1 2 3 -2 5 7

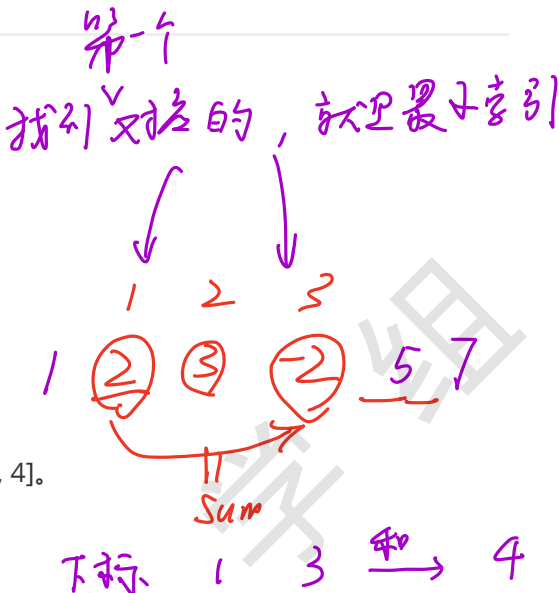
7

输出：(逗号前后没有空格)

[1,4]

注意：数组要求使用动态内存分配

考察：标记的使用，循环体



其他的都大于 4
找到的可能 ans 下标和

两个 for → 相当于 C6



for(int i = ?
j = (i+1) 从后一个开始
1 → 2, 3, -2, 5, 7

```
#include<bits/stdc++.h>
using namespace std;
int main(){
    int n;
    cin>>n;
    int* a = new int[n];
    for(int i = 0;i<n;i++){
        cin>>a[i];
    }
```

int s; 表示
cin>>s; sum.

bool/int flag = 1
for(int i=0; i<n; i++){
 if(flag == 0) break;
 if(flag == 1) continue;
}

怎么判断
这个循环做完了出来？
还没 break 出来？
用 flag!

用 a b c d e f 定义变量的方式不好！

```

int flag = 0;

for(int i = 0; i < n; i++){
    for(int j = i+1; j < n; j++){
        if((a[i]+a[j] == s)){
            cout << '[' << i << ',' << j << ']'<<endl;
            flag = 1;
            break;
        }
    }
}

if(flag) break;

if(!flag) cout<<"Wrong input!"<<endl;
delete []a;
return 0;
}

```

第1次找到的就是最少的 (answer)

1 发生
0 没有

可取的错误输入 (没有答案)

★ → delete []a;

2. 开灯问题

有 n 盏灯，编号为 $1 \sim n$ ，第 1 个人把所有灯打开，第 2 个人按下所有编号为 2 的倍数的开关（这些灯将被关掉），第 3 个人按下所有编号为 3 的倍数的开关（其中关掉的灯将被打开，开着的灯将被关闭），依此类推。一共有 k 个人，问最后有哪些灯开着？输入： n 和 k ，输出开着的灯编号。 $k \leq n \leq 1000$

输入

输入一组数据: n 和 k

输出

输出开着的灯编号

长度定义为 1005

1000 1000 太大了

样例输入

7 3

样例输出

1 5 6 7

考察：01标记与异或运算

$n = 7$ $k = 3$

```
#include<iostream>
using namespace std;
int main()
{
    int n, k, i;
    cin >> n >> k;

    int array[n];
    for(i = 0; i < n; i++)
        array[i] = 1;

    for(i = 2; i <= k; i++)
    {
        for(int p = 1; p < n; p++)
        {
            if((p+1) % i == 0)
                array[p] = !array[p];
        }
    }

    for(i = 0; i < n; i++)
    {
        if(array[i])
            cout << i+1 << " ";
    }
    return 0;
}
```

	1	2	3	4	5	6	7
1 →	1	1	1	1	1	1	1
2 →	1	0	1	0	1	0	1
3 →	1	0	0	1	1	1	0

$!! = 0$ $!0 = 1$ \Rightarrow
 $\underline{\quad}$ $\underline{\quad}$ \times

$$1 \wedge 1 = 0$$

\wedge

异或运算：(1)

$\begin{matrix} x & y \end{matrix}$	1	0
1	0	1
0	1	0

$$\begin{aligned} a=b & \quad a \wedge b = 0 \\ a \neq b & \quad a \wedge b = 1 \end{aligned}$$

```
#include<bits/stdc++.h>
```

```
int a[1007];
```

```
int main(){
```

```
int n, k;
```

```
std::cin >> n >> k;
```

```
for(int i = 1; i <= k; ++i)
```

```
for(int j = i; j <= n; j+=i)
```

```
a[j]^=1;
```

```
for(int i = 1; i <= n; ++i)
```

```
if(a[i])std::cout<< i << ' ';
```

```
return 0;
```

```
}
```

$0 \wedge 1 = 1$
 $1 \wedge 1 = 0$

$a[j] \wedge 1 \Rightarrow a[j] = a[j] \wedge 1$

(+) 类比

没有 (+) (因为从1开始)

3.字符串统计

键盘输入一段文字，输出其中出现次数最多的词及其出现次数。

如果有多个词出现次数相同，则输出最少的那个词及其次数。

注意：此题大小写敏感;有可能输入多行，输入结束时没有回车

样例

输入：

He is a teacher and his teacher is his father

输出：

(中间一个空格)

his 2

测试时用 "ctrl + z" (windows)

EOF

std: getline

string: getline

有区别(重载)

考察: while(cin), 简单桶排序

Linux (ubuntu...)

是另一个键

多行输入:

he is a teacher

you are so good

...

EOF (ctrl + z) < windows

```
#include<bits/stdc++.h>
```

```
using namespace std;
```

```
#define max_len 100
```

```
string save_str[max_len];
```

```
int cnt_str[max_len];
```

```
int pos;
```

```
int main(){
```

```
string s;
```

```
while(cin>>s){
```

maxlen = 100

str = string.

count. → cnt

String array[] 的存储形式

"abc" "d" "def" ↓

save_str [] [] [] [] []

string数组存储:

s = "He"

pos = 0

(He)

字典序

abc
cba

abc < cba

可变长度的数组

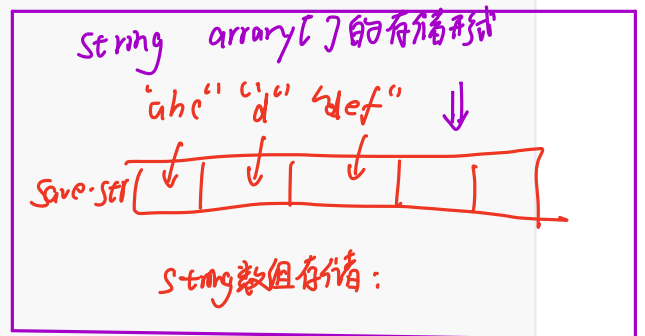
string s

初始化.

Cnt - str [0][0][0][0] ...
save - str [0][0][0][0] ...

0 < " " ⇒ string abo.

全局数组



```

int flag = 0;
for(int i = 0; i < pos; i++){
    if(s == save_str[i]){
        cnt_str[i]++;
        flag = 1;
        break;
    }
}

if(flag){
    save_str[pos] = s;
    cnt_str[pos]++;
    pos++;
}
    
```

桶

扫描顺序

save_str	He	is	he	a	b		
cnt_str	1	2	1	1	1		

pos → ...

用pair可以简化

He is he a is b is he ...

桶

扫描顺序

save_str	He	is	he	a	b
cnt_str	1	3	2	1	1		
下标	0	1	2	3	4		

$3 > 1$
 $3 > 2$
 $3 > 1$
 $3 > 1$

$p = 0$
 $i = 1$
 $i = 2$
 $i = 3$
 $i = 4$

相应双指针 (p, i)

```

int p = 0; // p 类似指针
for(int i = 1; i < pos; i++){
    bool f1 = (cnt_str[i] > cnt_str[p]);
    bool f2 = (cnt_str[i] == cnt_str[p]) && (save_str[i] < save_str[p]);

    if(f1 || f2) p = i;
}
    
```

p → 答案 (当前 max)
 i → 比较的对象

string 实现了比大小

$\{(string) \ a < b\} \Leftrightarrow \{(int) \ 1 < 2\}$
 相当于

```

cout << save_str[p] << ' ' << cnt_str[p] << '\n';
return 0;
    
```

1 3 5 7 2 8 9 0 10 10

for ...

```

int tmp;
cin >> tmp;
    
```

下标

值

→	0	1	2	3	4	5	6	7	8	9	10
→	1	1		1				1	1	1	2

$a[tmp]++$ $a[9]$

0 1 3 7 8 9 10

for () 相同的数

if (a[i]) cout << a[i].

O(n).

