



例题一、求完数



西南大学附属中学
High School Affiliated to Southwest University

题目描述

一个数如果恰好等于它的因子（能被它除尽的数，除本身）之和。比如6，它的因子有1,2,3,并且 $6=1+2+3$ ，则6是完数。输出n以内所有完数。（ $n < 10000$ ）

输入

一行，一个整数n

输出

输出n以内的所有完数，每个数之间用空格隔开。

样例输入

1000

样例输出

6 28 496

西 | 大 | 附 | 中 | 信 | 息 | 学 | 竞 | 赛 |
High School Affiliated to Southwest University



算法

列举1~n之间的数字i

列举i的因数j **范围: 1~i/2**

将因数j加起来

判断是否等于i本身, 如果是, 输出数字

```
int n,i,j,sum;
cin>>n;
sum=0;
for(i=1;i<=n;i++)
{
    sum=0; //变量初始化
    for(j=1;j<=i/2;j++){
        if(i%j==0)
            sum+=j;
    }
    if(sum==i) cout<<i<<" ";
}
```



例题二、与7无关的数



西南大学附属中学
High School Affiliated to Southwest University

题目描述

一个正整数，如果它能被7整除，或者它的某一位上的数字为7，则称它为“与7相关”的数。现在，请变成求出所有小于等于n的“与7无关的数”的正整数个数。

输入

一行一个正整数n， $n \leq 10^6$

输出

一行一个整数，表示答案

样例输入

21

样例输出

17

西 | 大 | 附 | 中 | 信 | 息 | 学 | 竞 | 赛 |
High School Affiliated to Southwest University

算法

列举1~n之间的数字

判断是否能被7整除

判断是否含有7的数字

判断以上两种情况是否都不存在，计数器+1

```
int i,n,t,flag1,flag2,cnt=0;
cin>>n;
flag1=flag2=1;
for(i=1;i<=n;i++){
    flag1=1,flag2=1;
    if(i%7==0){
        flag1=0;
    }
    else{
        t=i;
        while(t!=0){
            if(t%10==7){
                flag2=0;
            }
            t=t/10;
        }
        if(flag1==1&&flag2==1)
            cnt++;
    }
}
cout<<cnt<<endl;
```



无flag版本



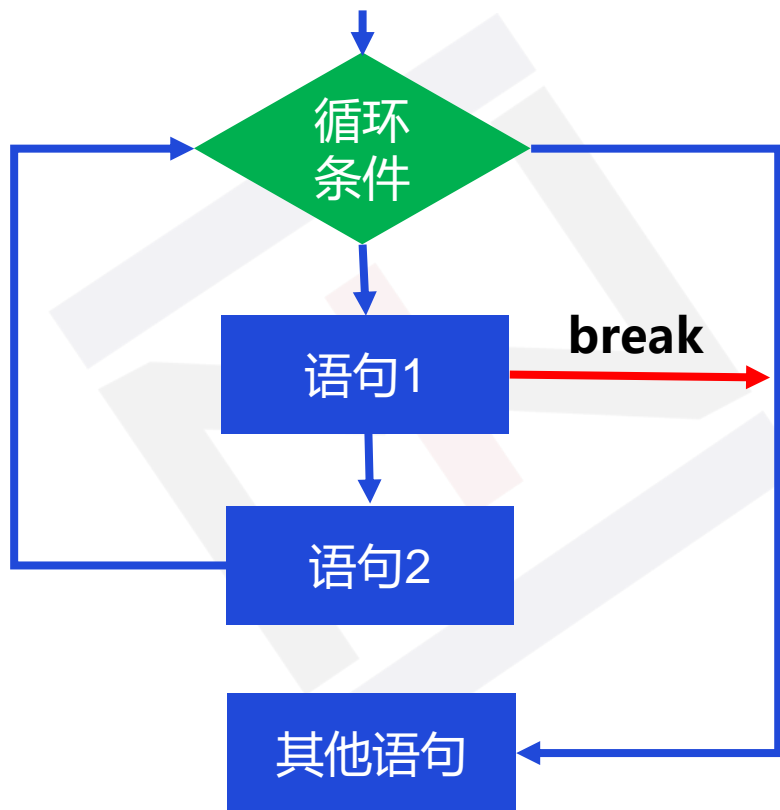
西南大学附属中学
High School Affiliated to Southwest University

```
int n,cnt,i,j,t;
cin>>n;
cnt=0;
for(i=1;i<=n;i++){
    if(i%7==0){
        continue;
    }
    t=i;
    while(t!=0){
        if(t%10==7){
            break;
        }
        t/=10;
    }
    if(t==1){
        continue;
    }
    cnt++;
}
cout<<cnt<<endl;
```

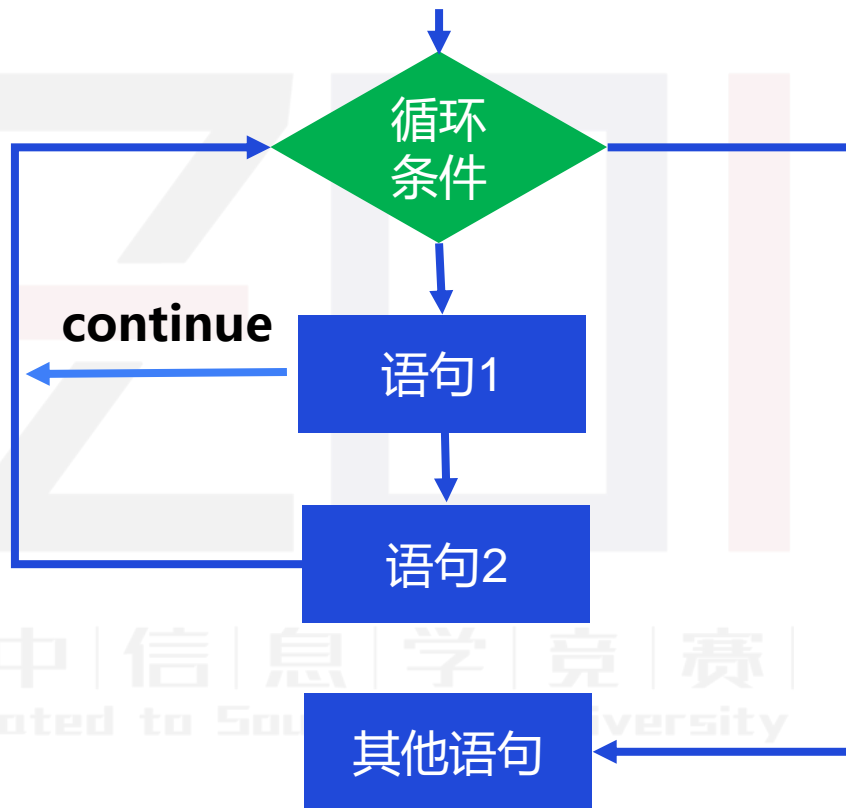
break:终止本层循环

continue:跳过本层的本次循环

西 | 大 | 附 | 中 | 信 | 息 | 学 | 竞 | 赛 |
High School Affiliated to Southwest University



break(终止本层循环)



continue(跳过本次循环)



例题三、金币



西南大学附属中学
High School Affiliated to Southwest University

题目描述

国王将金币作为工资，发放给忠诚的骑士。

第一天骑士收到一枚金币；之后两天（第二天和第三天），每天收到两枚金币；之后三天（第四、五、六天），每天收到三枚金币；之后四天，每天收到四枚金币，以此类推；这种工资发放模式会一直延续下去，当连续 N 天收到 N 枚金币后，骑士会在之后的 $N+1$ 天，每天收到 $N+1$ 枚金币。

请计算前 K 天里，骑士一共获得了多少金币。

输入

输入包含一个正整数 K ，表示发放金币的天数。

输出

输出一个正整数，即骑士收到的金币数。

样例输入

6

样例输出

14

此题仅作简单的提示，下来再思考一下

可以发现，国王给金币的规律是这样的：

1
22
333
4444
55555
.....



这是第六天

天数≠行数 行数需要计算

与三角形系列题目类似，但有所不同



导入：陶陶摘苹果



西南大学附属中学
High School Affiliated to Southwest University

陶陶家的院子里有一棵苹果树，每到秋天树上就会结出10个苹果。苹果成熟的时候，陶陶就会跑去摘苹果。陶陶有个30厘米高的板凳，当她不能直接用手摘到苹果的时候，就会踩到板凳上再试试。

现在已知10个苹果到地面的高度，以及陶陶把手伸直的时候能够达到的最大高度，请帮陶陶算一下她能够摘到的苹果的数目。假设她碰到苹果，苹果就会掉下来。

输入

第一行一个整数 h ，表示陶陶把手伸直的时候能够达到的最大高度。

第二行包含10个苹果到地面的高度

输出

输出陶陶能够摘到的苹果的数目。

样例输入

110

100 200 150 140 129 134 167 198 200 111

样例输出

5

样例输入

110

100 200 150 140 129 134 167 198 200 111

样例输出

5

```
#include<bits/stdc++.h>
using namespace std;
int main()
{
    int x,i,h,s=0;
    cin>>h;
    for(i=1;i<=10;i++){
        cin>>x;
        if((h+30)>=x) s=s+1;
    }
    cout<<s;
    return 0;
}
```



导入：陶陶摘苹果(NOIP 2005普及组)



西南大学附属中学
High School Affiliated to Southwest University

陶陶家的院子里有一棵苹果树，每到秋天树上就会结出10个苹果。苹果成熟的时候，陶陶就会跑去摘苹果。陶陶有个30厘米高的板凳，当她不能直接用手摘到苹果的时候，就会踩到板凳上再试试。

现在已知10个苹果到地面的高度，以及陶陶把手伸直的时候能够达到的最大高度，请帮陶陶算一下她能够摘到的苹果的数目。假设她碰到苹果，苹果就会掉下来。

输入

第一行包含10个苹果到地面的高度

第二行一个整数 h ，表示陶陶把手伸直的时候能够达到的最大高度。

输出

输出陶陶能够摘到的苹果的数目。

样例输入

100 200 150 140 129 134 167 198 200 111
110

样例输出

5

样例输入

100 200 150 140 129 134 167 198 200 111

110

样例输出

5

```
#include<bits/stdc++.h>
using namespace std;
int main()
{
    int x,i,h,s=0;
    for(i=1;i<=10;i++){
        cin>>x;
        if((h+30)>=x) s=s+1;
    }
    cin>>h;
    cout<<s;
    return 0;
}
```

问题：前十个苹果的值需要先存下，等高度输入了才能判断

解决办法：设置10个变量a1,a2,a3...a10

```
#include<bits/stdc++.h>
using namespace std;
int main()
{
    int x,i,h,s=0;
    int a1,a2,a3,a4,a5,a6,a7,a8,a9,a10; //10个苹果高度
    cin>>a1>>a2>>a3>>a4>>a5>>a6>>a7>>a8>>a9>>a10; //输入
    cin>>h;
    if((h+30)>=a1) s=s+1; //判断
    if((h+30)>=a2) s=s+1;
    if((h+30)>=a3) s=s+1;
    if((h+30)>=a4) s=s+1;
    .....
    cout<<s;
    return 0;
}
```

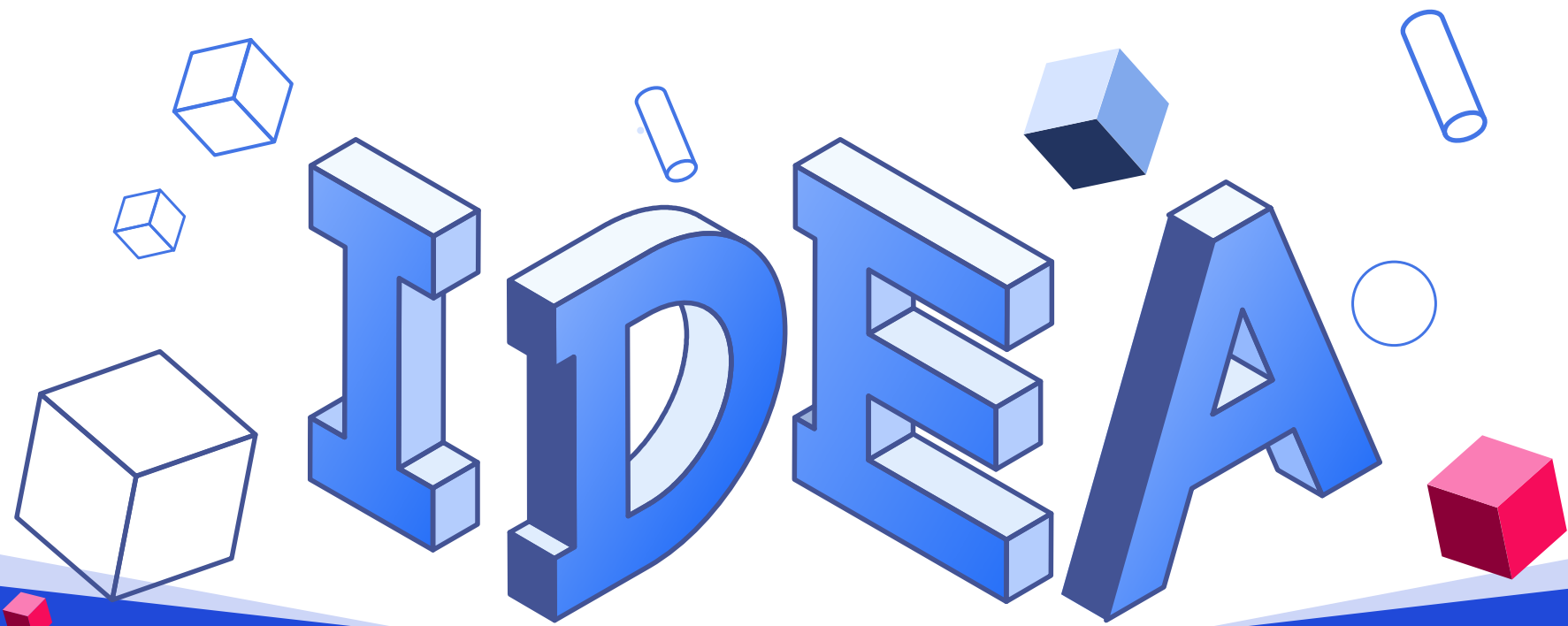
- 由于变量值的覆盖，需要定义10个int变量存储
- 就算有足够的耐心定义完这些变量，你还得不断地写对应的判断语句

Q:如果出题人“良心发现”，觉得给陶陶10个苹果不够，给10000个，甚至给n个($n \leq 1000000$)，怎么办？

```
#include<bits/stdc++.h>
using namespace std;
int main()
{
    int x,i,h,s=0;
    int a1,a2,a3,a4,a5,a6,a7,a8,a9,a10; //10个苹果高度
    cin>>a1>>a2>>a3>>a4>>a5>>a6>>a7>>a8>>a9>>a10; //输入
    cin>>h;
    if((h+30)>=a1) s=s+1; //判断
    if((h+30)>=a2) s=s+1;
    if((h+30)>=a3) s=s+1;
    if((h+30)>=a4) s=s+1;
    .....
    cout<<s;
    return 0;
}
```

所遇到的问题:

- 1.需要定义大量的、相同数据类型的变量存储数据
- 2.代码冗长，代码重复



信息思维系列课程

数组之初识——一维数组



什么是数组?



西南大学附属中学
High School Affiliated to Southwest University

有限个 **数据类型相同** 的变量的集合
(存放数据的容器)

| 西 | 大 | 附 | 中 | 信 | 息 | 学 | 竞 | 赛 |
High School Affiliated to Southwest University



格式：数据类型标识符 数组名 [数组大小]

常量/常量表达式

例：定义一个整型的数组，取名为a，数组大小为10

```
int a[10];    int a[2*5];
```



定义一维数组



西南大学附属中学
High School Affiliated to Southwest University

例：定义一个a数组，存储n个数字($n \leq 10000$)

错误定义

```
int n;  
cin >> n;  
int a[n];
```

```
int n=10000;  
int a[n];
```

可修改为



```
const int n=10000;  
int a[n];
```

说明：const可将变量设置为不可修改，等同于常量

新的C++标准虽然已经支持变量作为数组的大小

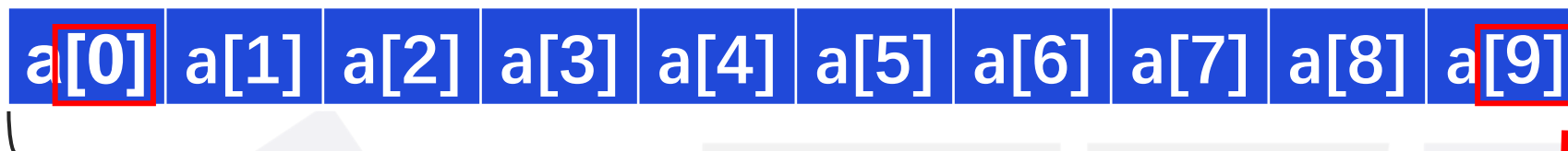
但现行信息学比赛的C++标准里，数组[]内，必须是一个常量或者是常量表达式



一维数组的存储形式



西南大学附属中学
High School Affiliated to Southwest University



`int a[10];`

数组元素的访问：数组下标

精髓

类似于一排柜子：



特点

1. 内存连续分配
2. 存储相同的数据类型
3. 数组下标从0开始，最大的下标不超过数组大小-1
4. 访问数组元素，通过下标进行访问



定义数组: `int a[10];`

使用数组:

`a[0]` `a[1]` `a[2]` `a[3]` `a[4]` `a[5]` `a[6]` `a[7]` `a[8]` `a[9]`

两种情况的区别:

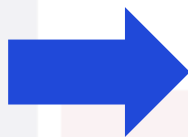
定义数组时, []内部的数字代表申请的数组**大小**

使用数组时, []内部的数字代表数组元素的**下标**



与变量输入输出类似

```
cin>>a1;  
cin>>a2;  
cin>>a3;  
cout<<a1;  
cout<<a2;  
cout<<a3;
```



```
cin>>a[1];  
cin>>a[2];  
cin>>a[3];  
cout<<a[1];  
cout<<a[2];  
cout<<a[3];
```

常见错误

```
int a[10];  
cin>>a;  
cout<<a;
```

~~对数组整体进行输入输出~~

只能通过数组下标对数组元素**一一操作**，无法对数组整体进行操作

如果仅仅像前面那样对一维数组进行访问...

1.输入10个数:

```
cin>>a[1];  
cin>>a[2];  
cin>>a[3];
```

.....

2.判断是否大于h+30:

```
if((h+30)>=a[1]) s=s+1;  
if((h+30)>=a[2]) s=s+1;  
if((h+30)>=a[3]) s=s+1;  
if((h+30)>=a[4]) s=s+1;  
...
```

重复在做某种事情

循环

数组和循环天生是一对“好朋友”

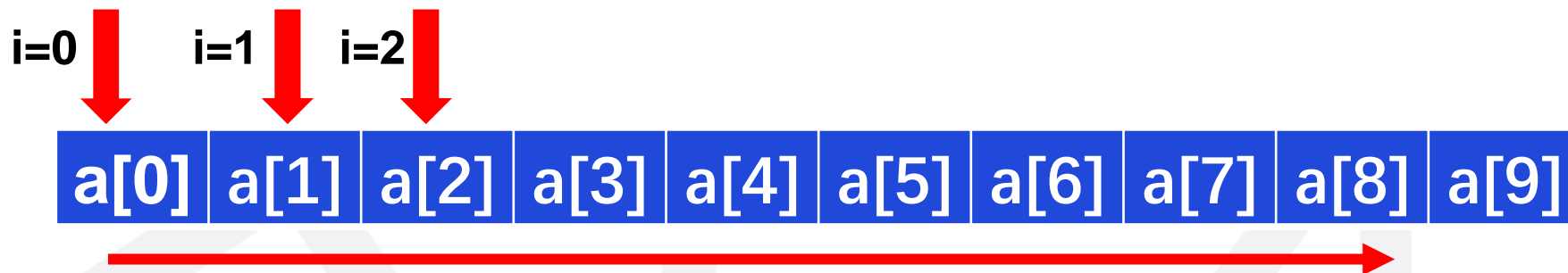
是不是与直接定义变量进行使用没有区别?



循环访问数组



西南大学附属中学
High School Affiliated to Southwest University



下标递增 i++

通过**改变下标**的值来访问数组

```
cin>>a[1];  
cin>>a[2];  
cin>>a[3];
```



```
int i;  
for(i=1;i<=50;i++)  
    cin>>a[i];
```

通过循环访问所有数组元素，
我们称为“**遍历**”操作

西 | 大 | High School Affiliated to Southwest University

学 | 竞 | 赛



数组与循环的关系



西南大学附属中学
High School Affiliated to Southwest University



西 | 大 | 附 | 中 | 信 | 息 | 学 | 竞 | 赛 |
High School Affiliated to Southwest University

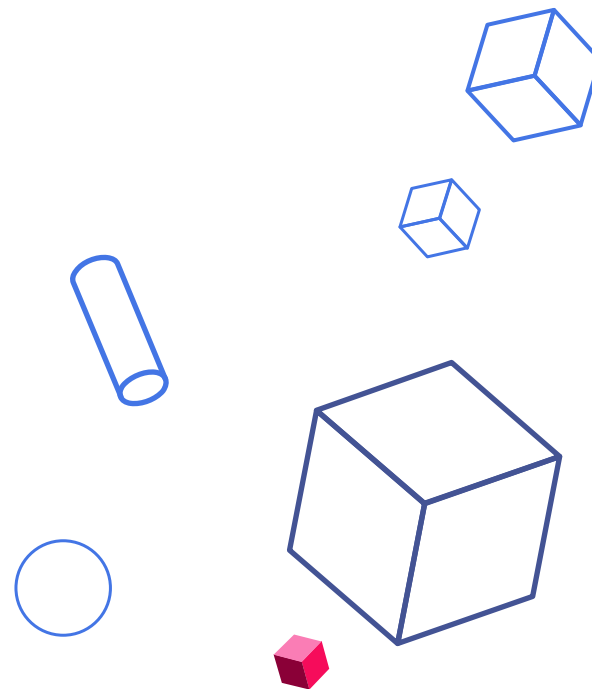
数组的定义

有限个**数据类型相同**的变量的集合

数组的使用

通过数组下标进行访问，通常与循环结合使用，每个数组元素与一般变量使用没有区别。

5min: 做一下逆序输出





顺序输入输出

```
for(i=1;i<=n;i++)  
    cin>>a[i];
```

```
for(i=1;i<=n;i++)  
    cout<<a[i];
```

逆序输入输出

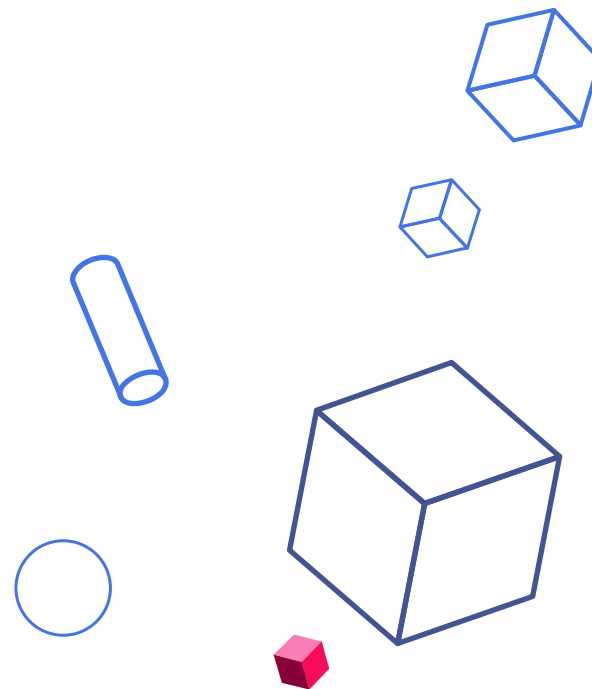
```
for(i=n;i>=1;i--)  
    cin>>a[i];
```

```
for(i=n;i>=1;i--)  
    cout<<a[i];
```

通过数组下标，可以直接访问数组里元素的值

如果舍弃a[0]不用，数组大小为：最大的n+1

针对数组的几个操作



操作一：删除操作



西南大学附属中学
High School Affiliated to Southwest University

第一行输入 n ($n \leq 100$), 第二行输入 n 个数的序列, 最后一行再输入一个数, 是需要删除的位置, 最后把整个数列输出。

样例输入

5

1 3 5 7 9

3

样例输出

1 3 7 9

题意

给定一个数列, 删除指定位置的数

分析



西南大学附属中学
High School Affiliated to Southwest University



所做的事情:

1. 从第k位 (要删除的位置) 开始;
2. 让第i+1位的数挪动到第i位, $a[i]=a[i+1]$

分析



西南大学附属中学
High School Affiliated to Southwest University



代码实现

```
for( i=k ; i<=n-1 ; i++ ){//循环代表挪动次数, 挪动n-k次  
    a[i]=a[i+1];        //向前挪动  
}
```

代码



西南大学附属中学
High School Affiliated to Southwest University

```
#include <bits/stdc++.h>
using namespace std;
int main(){
    int i,n,k,a[101]; //k代表删除的位置
    cin>>n;
    for(i=1;i<=n;i++){ //输入
        cin>>a[i];
    }
    cin>>k;
    for(i=k;i<=n-1;i++){ //循环代表挪动次数，挪动n-k次
        a[i]=a[i+1]; //向前挪动
    }
    for(i=1;i<=n-1;i++){ //输出时，i循环到n-1
        cout<<a[i]<<" ";
    }
    return 0;
}
```


操作二：插入操作



西南大学附属中学
High School Affiliated to Southwest University

第一行输入 n ($n \leq 100$)，第二行输入 n 个数的序列，最后一行再输入两个数，一个是插入的位置，一个是插入的数，最后把整个数列输出。

样例输入

5

1 3 5 7 9

3 4

样例输出

1 3 4 5 7 9

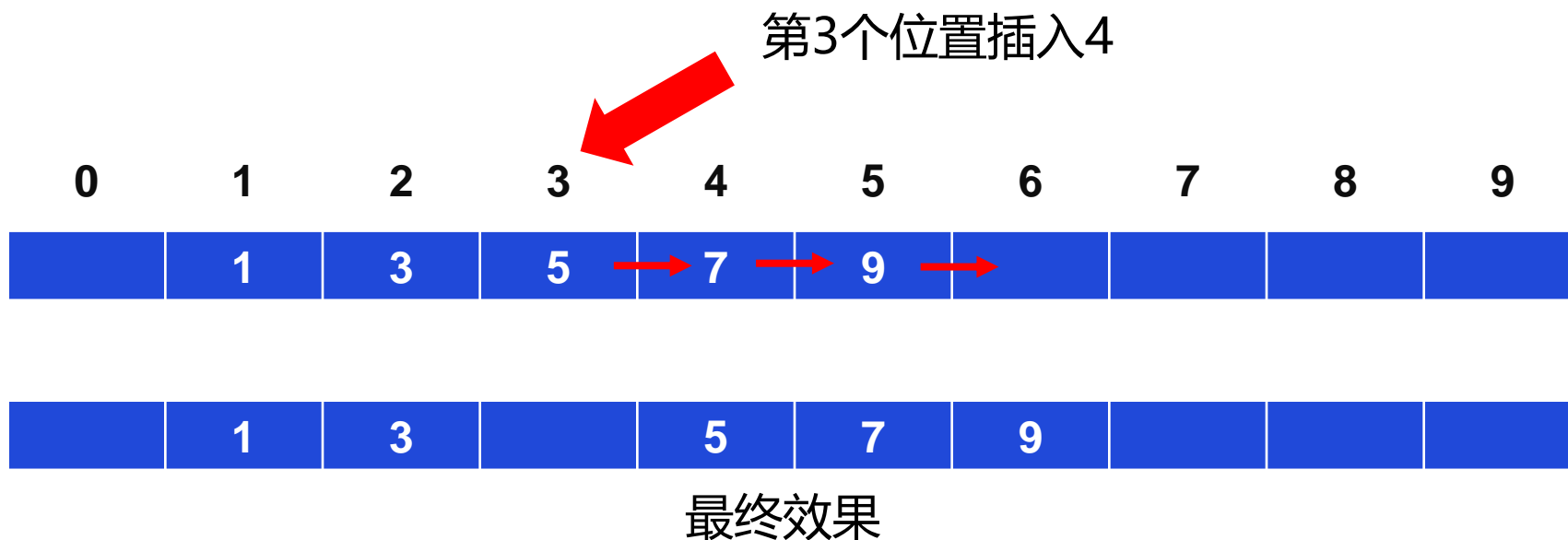
题意

给定一个数列，在指定的位置插入该数

如何挪位置?



西南大学附属中学
High School Affiliated to Southwest University



挪动步骤:

```
a[6]=a[5];  
a[5]=a[4];  
a[4]=a[3];
```

所要做的事情:

- 1.从最后一个位置开始, 不断往后挪动, 即 $a[i+1]=a[i]$
- 2.直到第k位(插入的位置)结束

插入



西南大学附属中学
High School Affiliated to Southwest University

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

再把4填入，完成插入

总结：

1.从最后一个位置开始，不断往后挪动， $a[i+1]=a[i]$ ，直到第k位(插入的位置)结束

2.腾出位置后，插入该数， $a[k]=x$;



代码实现

```
for( i=n; i>=k; i--){ //k代表插入的位置
    a[i+1]=a[i];      //不断地挪位置
}
a[k]=x;              //x代表插入的数
```

插入操作是从最后一个位置开始挪动

代码



西南大学附属中学
High School Affiliated to Southwest University

```
int i,n,k,x,a[102]; //k代表插入的位置, x代表插入的数
cin>>n;
for(i=1;i<=n;i++){
    cin>>a[i];
}
cin>>k>>x;
for(i=n;i>=k;i--){ //循环代表挪动次数, 挪动n-k+1次
    a[i+1]=a[i];    //向后挪动
}
a[k]=x;    //插入该数

for(i=1;i<=n+1;i++){ //输出时, i循环到n+1
    cout<<a[i]<<" ";
}
```

定义为102大小, 预防插入到最后一个位置的情况

操作三：查找操作



西南大学附属中学
High School Affiliated to Southwest University

第一行输入 n ($n \leq 100$)，第二行输入 n 个数的序列，第三行输入 t 表示需要查找 t 个数，最后一行是 t 个需要查找的数。如果值存在，输出编号；如果不存在，输出-1。

样例1输入

4

1 2 3 4

2

2 5

样例1输出

2 -1

题意

给定一个数列，找到某数的下标。

如何查找?



西南大学附属中学
High School Affiliated to Southwest University

1.把数存到一个数组

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

2.读入数字, 遍历数组

$x=2$									
	1	2	3	4					

找到

$x=5$									
	1	2	3	4					

输出下标

遍历数组还是没找到

输出-1

Q:如何判断是否找到?

设立一个标记变量flag

顺序查找

分析



西南大学附属中学
High School Affiliated to Southwest University

```
int i,j,n,t,x,flag,a[101]; //x存储查找的数, flag是标记作用的变量
cin>>n;
for(i=1;i<=n;i++){
    cin>>a[i];
}
cin>>t;
for(i=1;i<=t;i++){ //查找t次
    cin>>x;
    flag=0; //0表示没找到, 每次查找前都要初始化
    for(j=1;j<=n;j++){ //遍历a数组
        if(x==a[j]){ //找到x
            cout<<j<<" ";
            flag=1; //标记置为1, 代表找到
            break; //找到了就没必要再遍历数组了, break跳出
        }
    }
    if(flag==0) cout<<-1<<" "; //根据flag标记判断是否找到, 条件写j>n也可以
}
```

设立标记在程序中十分普遍, 也是常用的技巧

操作三：统计操作



西南大学附属中学
High School Affiliated to Southwest University

第一行输入 n ($n \leq 100$)，第二行输入 n 个数的序列，第三行输入 t 表示需要查找 t 个数，最后一行是 t 个需要查找的数。如果值存在，输出有几个；如果不存在，输出-1。

样例1输入

6

1 2 3 4 2 2

2

2 5

样例1输出

3 -1

题意

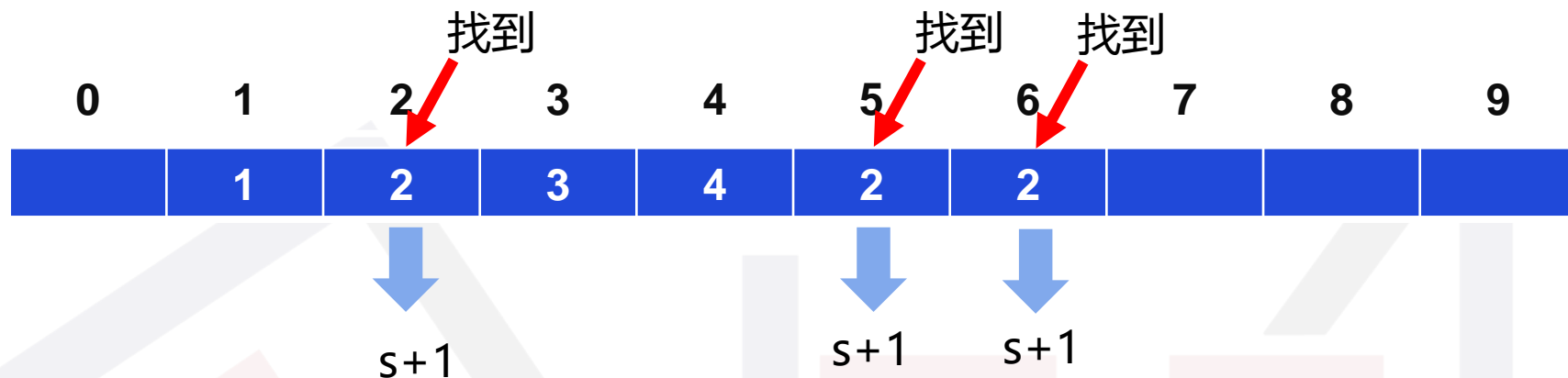
给定一个数列，输出某数的个数。



分析



西南大学附属中学
High School Affiliated to Southwest University



核心代码:

```
s=0;  
for(j=1;j<=n;j++){ //遍历a数组  
    if(x==a[j]){ //找到x  
        s=s+1; //统计个数  
    }  
}
```

统计是基于查找操作的



例题、校门外的树



西南大学附属中学
High School Affiliated to Southwest University

某校大门外长度为 L 的马路上有一排树，每两棵相邻的树之间的间隔都是1米。我们可以把马路看成一个数轴，马路的一端在数轴0的位置，另一端在 L 的位置；数轴上的每个整数点，即0, 1, 2, ..., L ，都种有一棵树。

由于马路上有一些区域要用来建地铁。这些区域用它们在数轴上的起始点和终止点表示。已知任一区域的起始点和终止点的坐标都是整数，区域之间可能有重合的部分。现在要把这些区域中的树（包括区域端点处的两棵树）移走。你的任务是计算将这些树都移走后，马路上还有多少棵树。

输入

第一行有两个整数 L ($1 \leq L \leq 10000$) 和 M ($1 \leq M \leq 100$)， L 代表马路的长度， M 代表区域的数目， L 和 M 之间用一个空格隔开。接下来的 M 行每行包含两个不同的整数，用一个空格隔开，表示一个区域的起始点和终止点的坐标。

输出

一行，这一行只包含一个整数，表示马路上剩余的树的数目。

样例输入

```
500 3
150 300
100 200
470 471
```

思考分析

样例输出

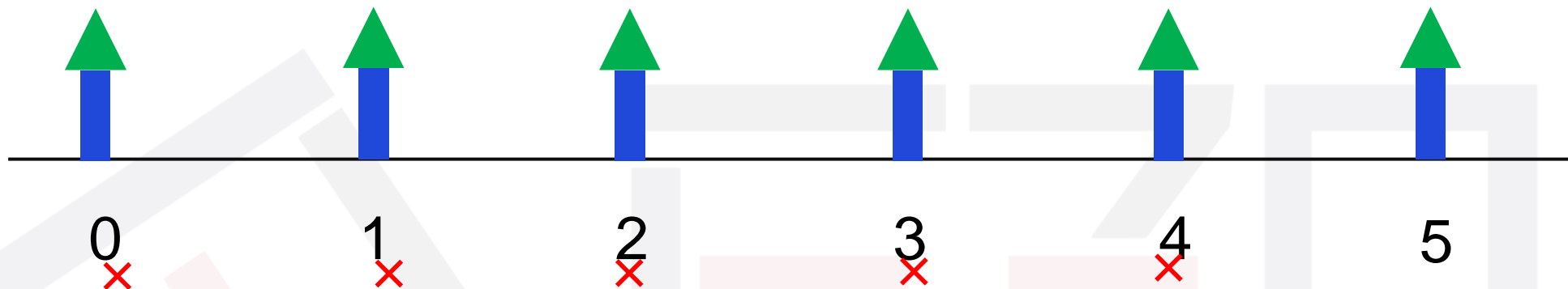
```
298
```



分析



西南大学附属中学
High School Affiliated to Southwest University



L=5

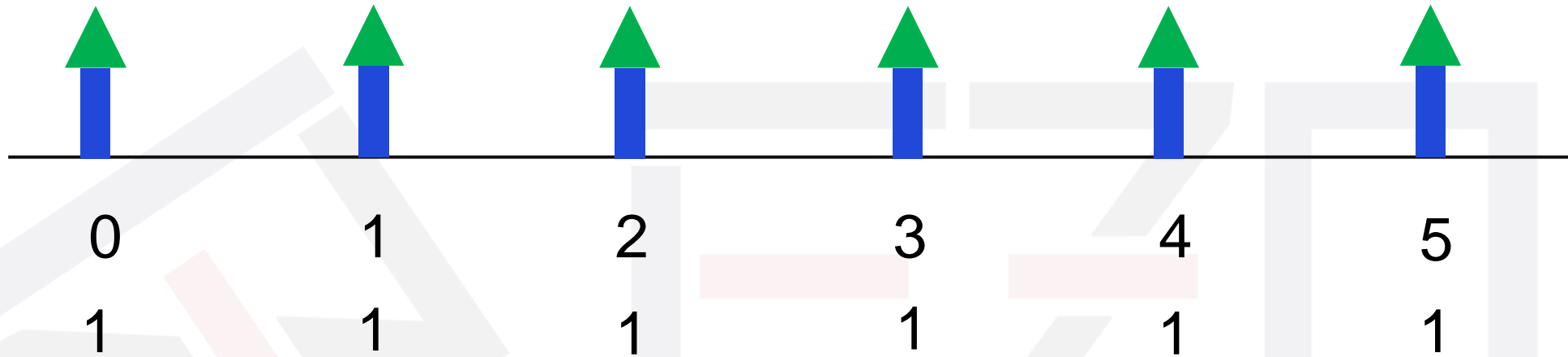
1 2

1 3

0 4

Q: 最终还剩下几棵树?

Q: 人是怎么铲树的?



Q:在计算机世界，我们用什么来表示有和无？

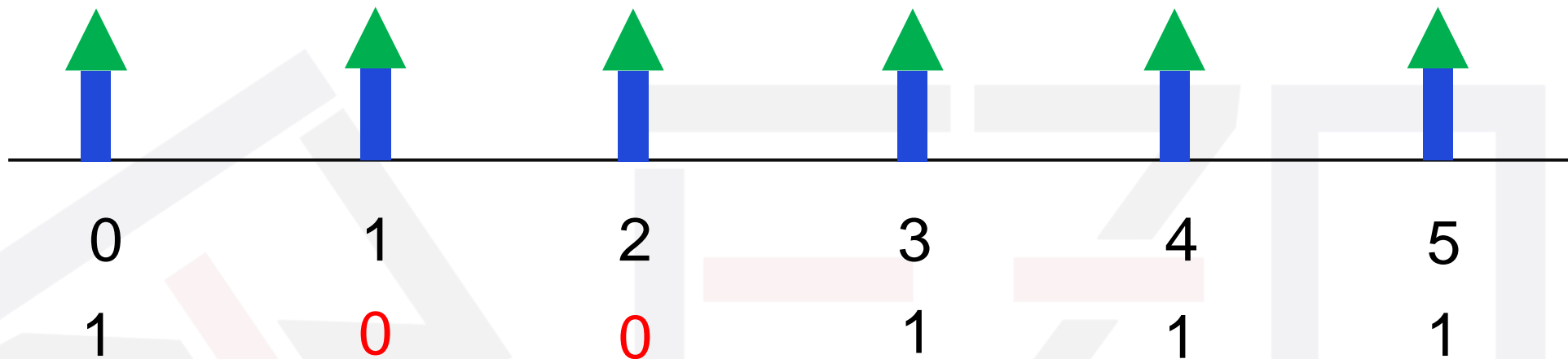
1和0



分析



西南大学附属中学
High School Affiliated to Southwest University



1 2

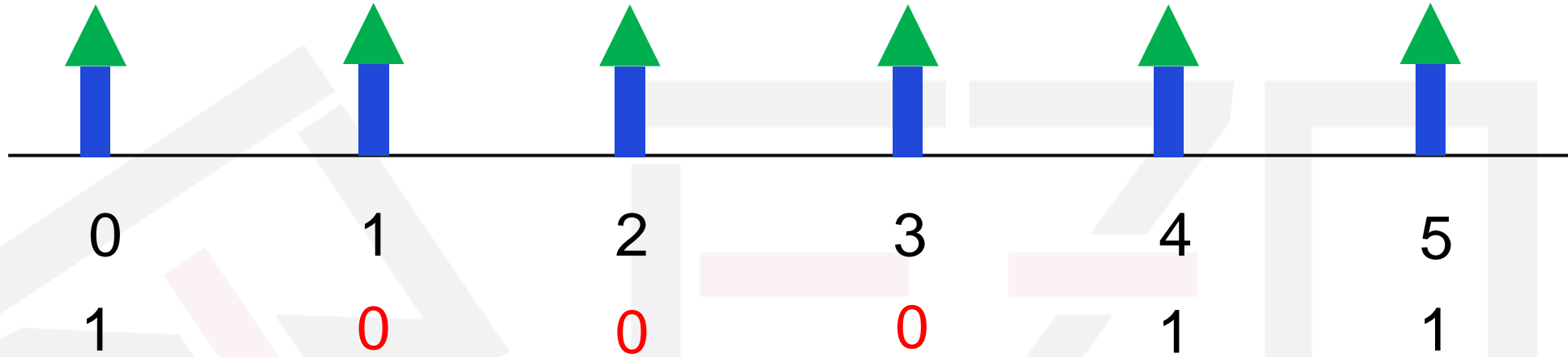
西 | 大 | 附 | 中 | 信 | 息 | 学 | 竞 | 赛 |
High School Affiliated to Southwest University



分析



西南大学附属中学
High School Affiliated to Southwest University



1 2
1 3

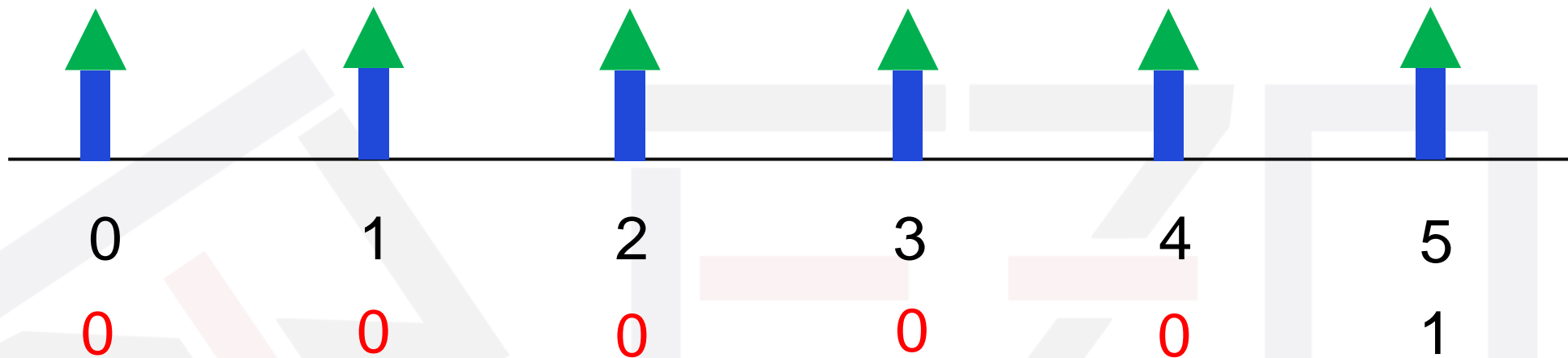
西 | 大 | 附 | 中 | 信 | 息 | 学 | 竞 | 赛 |
High School Affiliated to Southwest University



分析



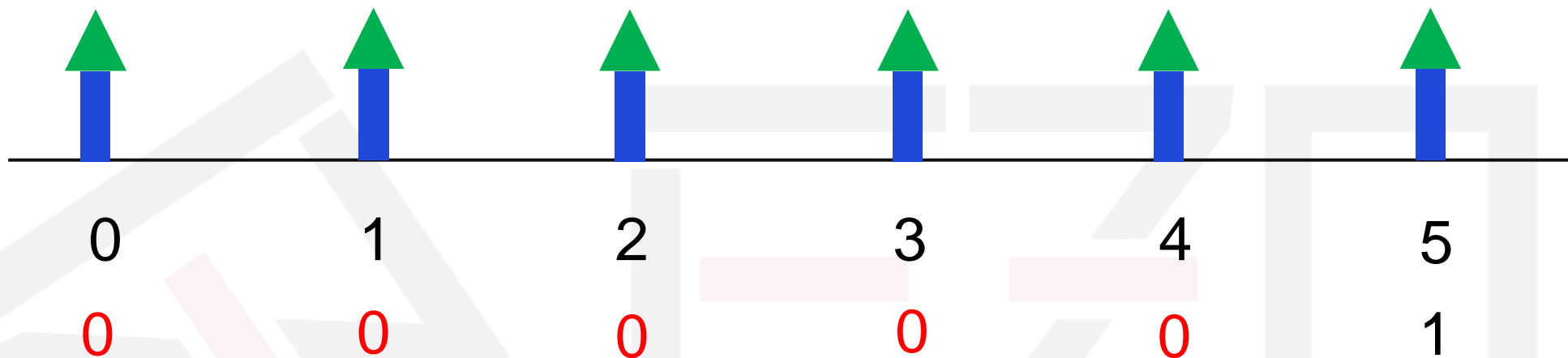
西南大学附属中学
High School Affiliated to Southwest University



1 2
1 3
0 4

问题的本质就转化为了 **“数1的个数”**

信息学的题目常常会有一些背景，要锻炼自己剥开问题外壳，看透问题本质的能力



1 2
1 3
0 4

Q: 如何表示这样的一组0和1

数组

0表示没有树, 1表示有树

标记数组 标记一组状态的情况



- 一维数组的定义，当有大量相同数据类型变量需要存储时，需要用到数组
- 一维数组具有连续分配的空间、下标访问数组元素(循环)
- 一维数组最大下标=数组大小-1
- 一维数组插入、删除、查找与统计操作

Thanks

For Your Watching

