程序设计语言基础回顾

对应《问题求解》

C语言引导(freestanding)


examples


借题发挥


Learning by
Doing

为什么学习程序设计语言

问题规模n很小　　寻找一般方法　　问题规模n很大

人类世界的需求 ⟶ **信息世界的投影**

富有启发性质　　快速, 精准

解线性方程组

列竖式计算

手算有理函数的积分

让我们看到了实在的解决一类问题的方法

Gauss消元法

运用计算机计算

R算法

如何复习: 搭积木说起

我们先不精确, 先上手写一点程序再说...

这幅图会出现若干次...

I. Starter Introduction

引例1. Hello world

tmp — nvim a.cpp — vim — nvim a.cpp ▸ python3 — 80×19

```
1 #include <iostream>
2 using namespace std;
3 int main(){
4     cout<<"Hello world!\n";
5     return 0;
6 }
```

\n 表示在末尾打印换行的符号

每个语句末尾都是有;


```
:!g++ a.cpp && ./a.out
Hello world!

Press ENTER or type command to continue
```

I. Starter Introduction

引例2. 不太好用的计算器

code — nvim calc.cpp — vim — nvim calc.cpp ▸ python3 — 80×26

```
1 #include <iostream>
2 using namespace std;
3 int main(){
4     // Calculates the value of 1+1
5     cout<<1+1<<endl;
6     // Calculates value of 2147483647+1...
7     // Or really?
8     cout<<2147483647+1<<endl;
9     // Calculates 3/2... Really?
10    cout<<3/2<<endl;
11    return 0;
12 }
13
```

```
:!g++ calc.cpp && ./a.out
[No write since last change]
calc.cpp:8:21: warning: overflow in expression; result is -2147483648 with type
'int' [-Winteger-overflow]
    cout<<2147483647+1<<endl;
                    ^
1 warning generated.
2
-2147483648
1

Press ENTER or type command to continue
```

这是什么? 过一会说, 假装不存在...

I. Starter Introduction

引例2. 不太好用的计算器

code — nvim calc.cpp — vim — nvim calc.cpp ▸ python3 — 80×26

```
1 #include <iostream>
2 using namespace std;
3 int main(){
4     // Calculates the value of 1+1
5     cout<<1+1<<endl;
6     // Calculates value of 2147483647+1...
7     // Or really?
8     cout<<2147483647+1<<endl;
9     // Calculates 3/2... Really?
10    cout<<3/2<<endl;
11    return 0;
12 }
13
```

```
:!g++ calc.cpp && ./a.out
[No write since last change]
calc.cpp:8:21: warning: overflow in expression; result is -2147483648 with type
'int' [-Winteger-overflow]
    cout<<2147483647+1<<endl;
                    ^
1 warning generated.
2
-2147483648
1

Press ENTER or type command to continue
```

I. Starter Introduction

引例2. 不太好用的计算器

code — nvim calc.cpp — vim — nvim calc.cpp ▸ python3 — 80×26

```
1  #include <iostream>
2  using namespace std;
3  int main(){
4      // Calculates the value of 1+1
5      cout<<1+1<<endl;
6      // Calculates value of 21474836471...
7      // Or really?
8      cout<<2147483647+1<<endl;
9      // Calculates 3/2... Really?
10     cout<<3/2<<endl;
11     return 0;
12 }
13
```

```
:!g++ calc.cpp && ./a.out
[No write since last change]
calc.cpp:8:21: warning: overflow in expression; result is -2147483648 with type
'int' [-Winteger-overflow]
    cout<<2147483647+1<<endl;
                    ^
1 warning generated.
2
-2147483648
1

Press ENTER or type command to continue
```

正数加起来是负数?这很明显不合道理, 为什么(stay tuned)

I. Starter Introduction

引例2. 不太好用的计算器

code — nvim calc.cpp — vim — nvim calc.cpp ▸ python3 — 80×26

```
1  #include <iostream>
2  using namespace std;
3  int main(){
4      // Calculates the value of 1+1
5      cout<<1+1<<endl;
6      // Calculates value of 2147483647+1...
7      // Or really?
8      cout<<2147483647+1<<endl;
9      // Calculates 3/2... Really?
10     cout<<3/2<<endl;
11     return 0;
12 }
13
```

```
:!g++ calc.cpp && ./a.out
[No write since last change]
calc.cpp:8:21: warning: overflow in expression; result is -2147483648 with type
'int' [-Winteger-overflow]
    cout<<2147483647+1<<endl;
                    ^
1 warning generated.
2
-2147483648
1

Press ENTER or type command to continue
```

3/2=1.5不等于1啊...

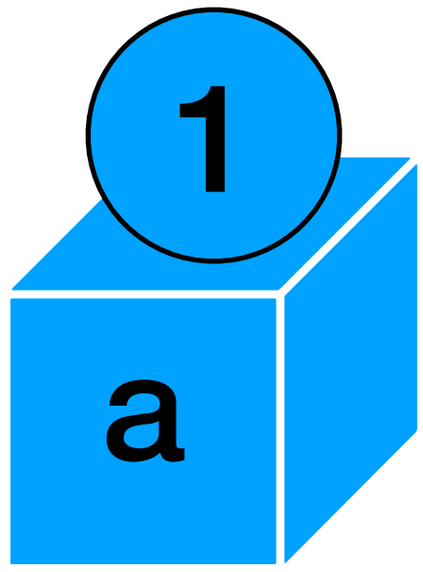
I. Starter Introduction

variable arithmetic expression

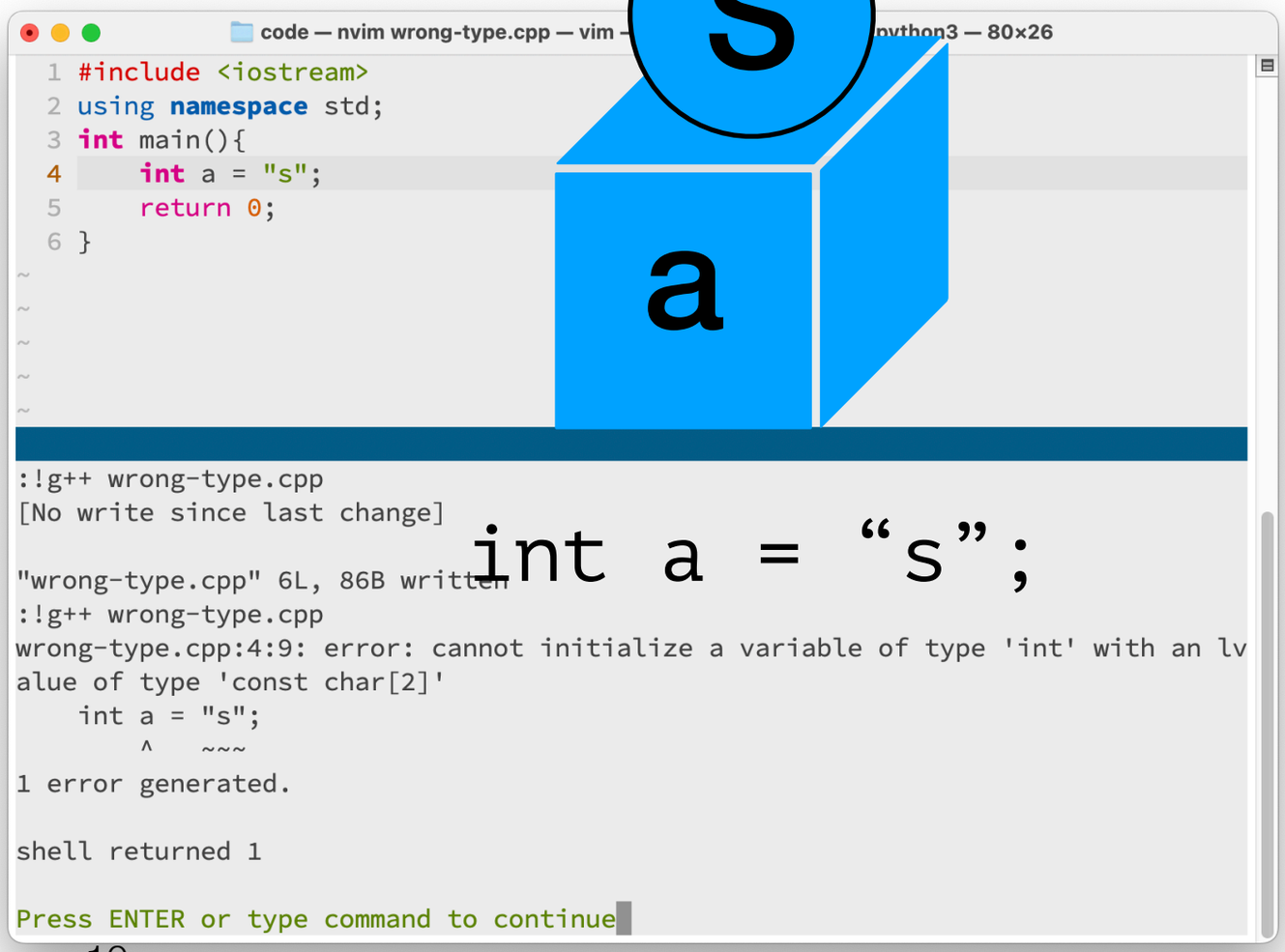
一. 变量与算术表达式

1. 变量: 带有类型的小盒子

```
int a = 1;
```



只能用来存整数(integer)



int a = “s”;

I. Starter Introduction

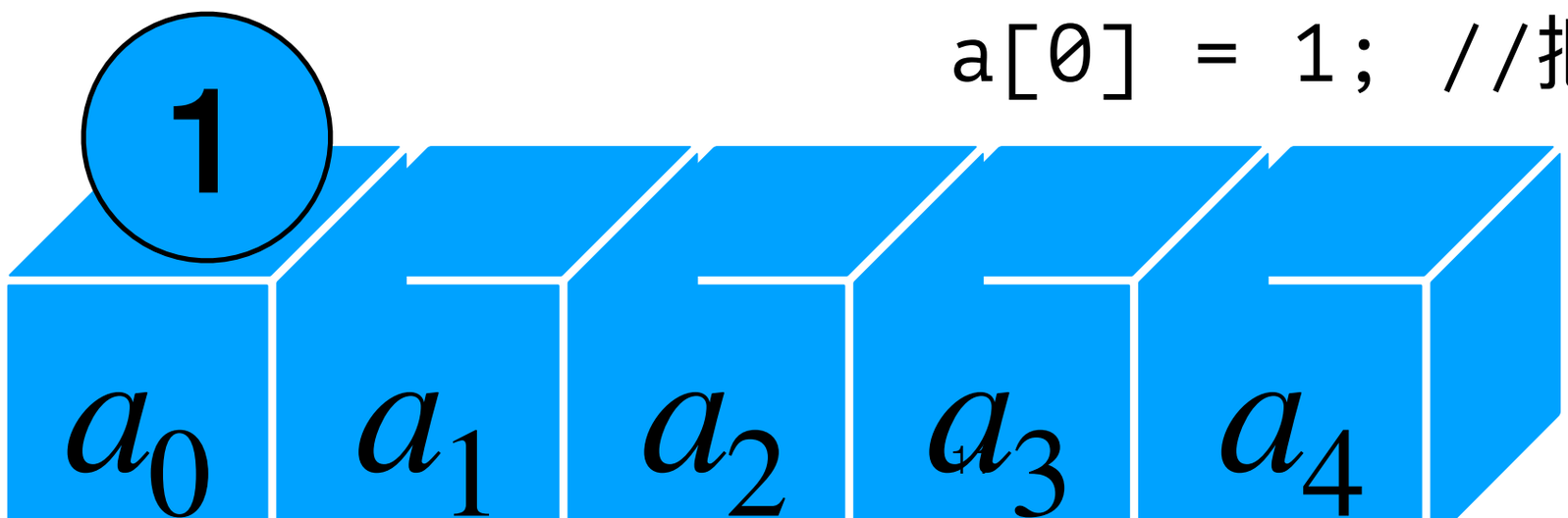
variable arithmetic expression

一. 变量与算术表达式

1. 变量: 带有类型的小盒子
2. 一些基本的类型 *type*
   - 整数: int
   - 小数: float/double
   - 字符: char
   - 真假: bool

创建n个 — 数组, 如: *array*

```
int a[5]; // 创建5个
a[0] = 1; //把第0个改做1
```



I. Starter Introduction

variable arithmetic expression

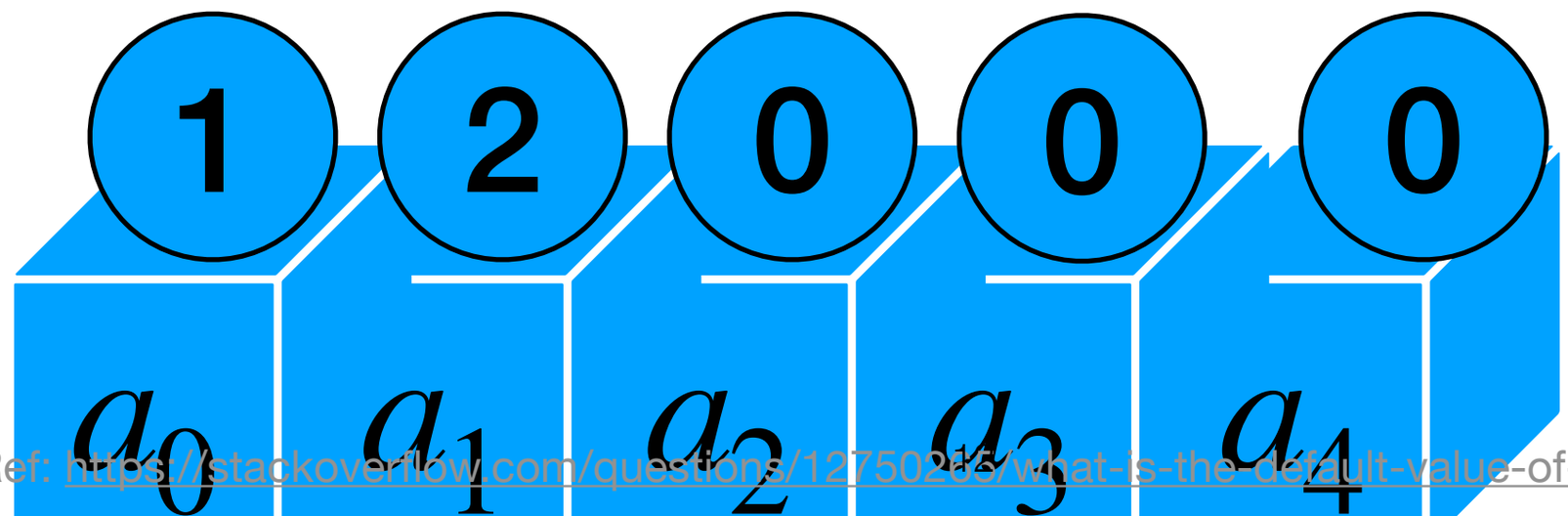
一. 变量与算术表达式

1. 变量: 带有类型的小盒子
2. 一些基本的*type*类型
   - 整数: int
   - 小数: float/double
   - 字符: char
   - 真假: bool

当然也有数组的数组
`a[N][N] = (a[N])[N]`

array

创建n个 — 数组, 如: `int a[5]={1,2}; // 创建5个`



Ref: https://stackoverflow.com/questions/12750269/what-is-the-default-value-of-members-of-a-char-array-in-c

I. Starter Introduction

variable arithmetic expression

一. 变量与算术表达式

3. 变量的转换

```
int a = 1.5;
```


```
double s = 3.0/2.0;
double s1 = (double)3/(double)2;
```

- 小数→整数: 只保留整数部分;
- 字符→整数: 按照ASCII码转换;
- bool→整数: 不是0的都是真.

I. Starter Introduction

variable arithmetic expression

一. 变量与算术表达式

4. 算术表达式

- 和中学学的一样, +, -, *, /, (,).
- 不一样的:
  - 逻辑与 &&, 或 ||, 非 !, 相等==.

例子:

```
int I=(1+2)/2*2; // I=2.
```

I. Starter Introduction

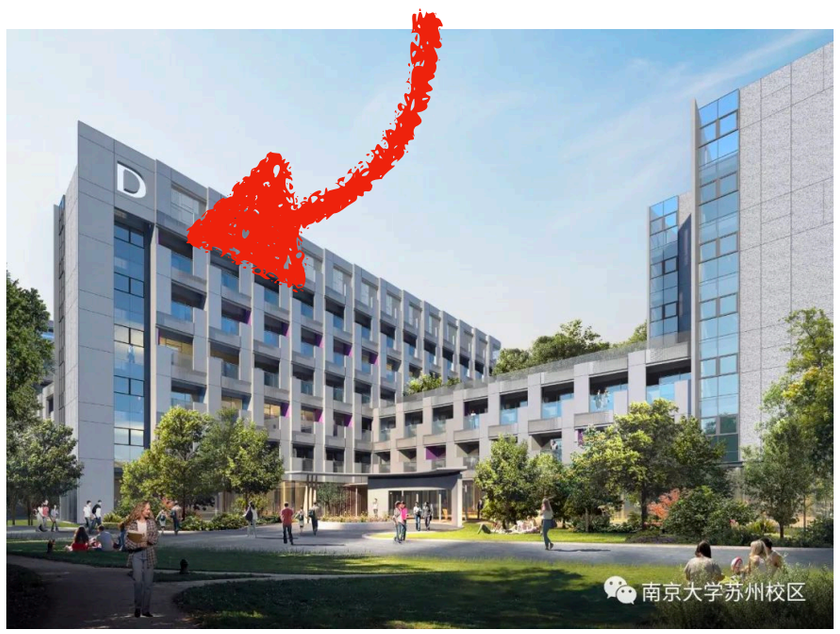
二. 控制程序的执行

排练节目

足够多的人

按照一定的方式做事情

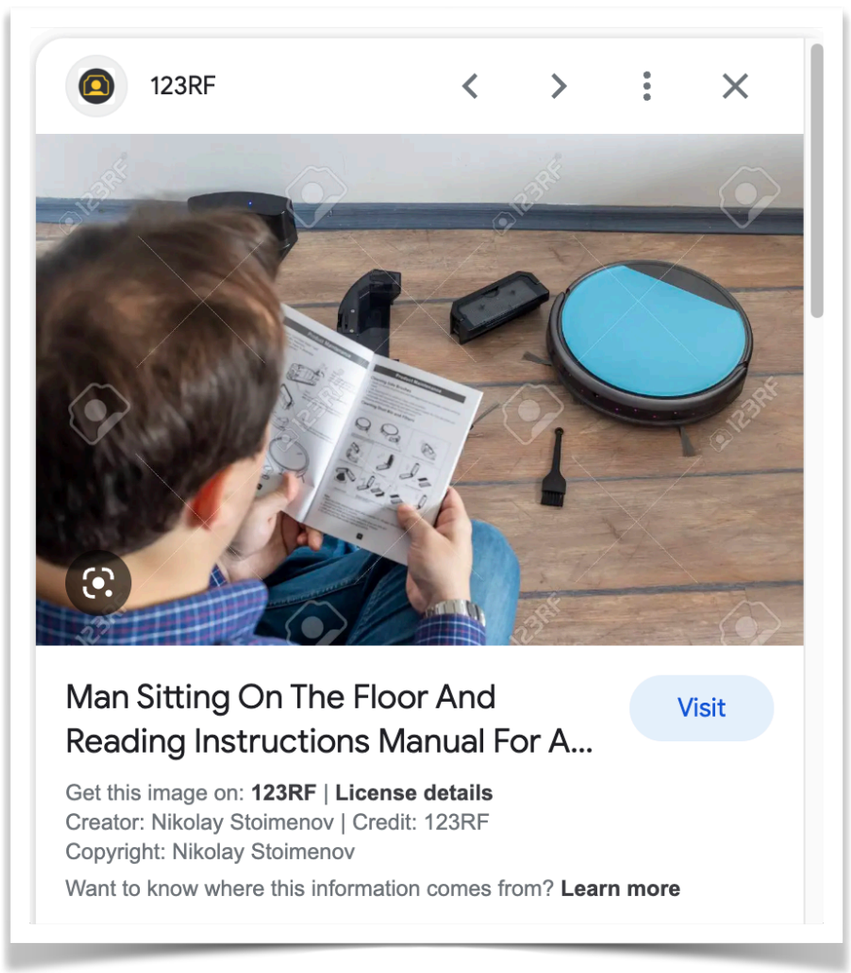
变量



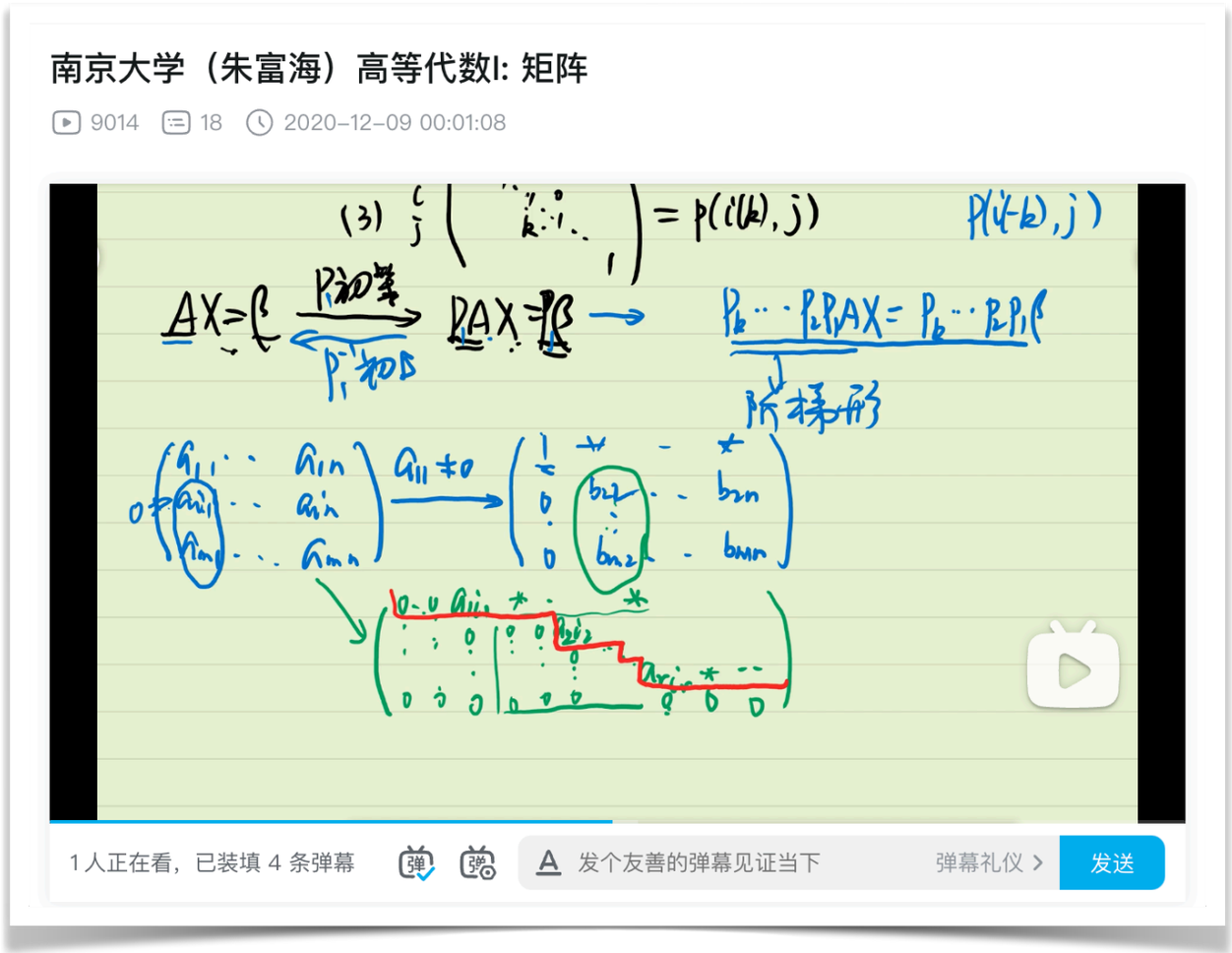
控制结构

I. Starter Introduction

二. 控制程序的执行



有东西出故障
按照说明书做故障排查



做初等行变化把矩阵化为阶梯型

I. Starter Introduction

二. 控制程序的执行

当前程序执行到这一行

如果条件满足的时候

下一次跳到这一行

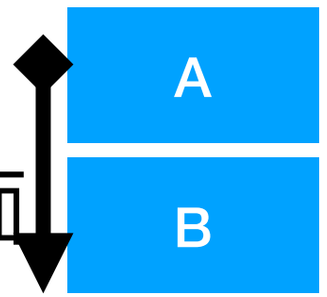
I. Starter Introduction

二. 控制程序的执行

sequential

1. 顺序结构: 按照顺序做事情

   先做A, 再做B. — 单纯把它们写在不同的行里面



如: 变量的交换

```
t=a;
a=b;
b=t;
```

⑸若有定义 int a[8]={1,2,3,4,5}; cout<<a[1]<<a[6]<<endl; 则输出为 ( )

A.16 B.10 C.20 D. 不确定的值

我们可以用pythontutor.com看一看!

I. Starter Introduction

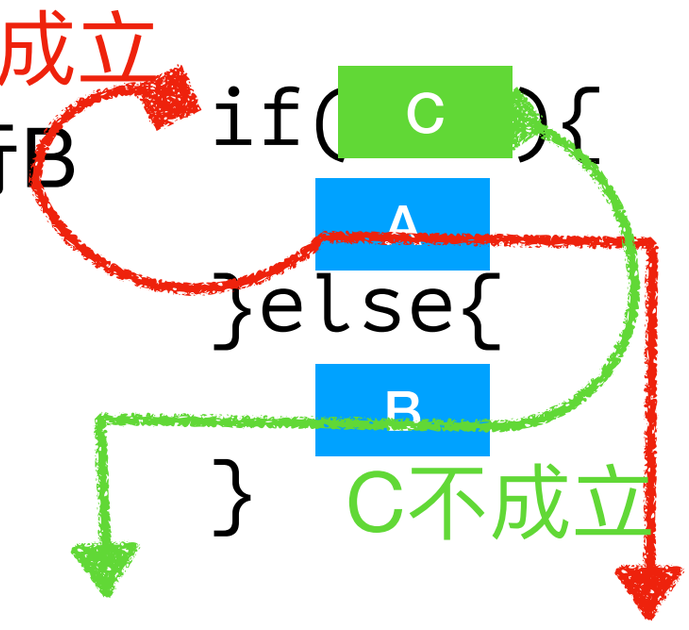
二. 控制程序的执行

conditional

2. 做判断

判断条件C, 如果成立, 就执行A; 否则, 执行B

```
bool light = 1;
if(light){
    cout<<“Go!”;
}else{
    cout<<“Stop!”;
}
```



I. Starter Introduction

二. 控制程序的执行

2. 做判断

判断条件C, 如果成立, 就执行A; 否则, 执行B

当然可以判断多个条件

```
if(条件1){
    对策1
}else if(条件2){
    对策2
}else if(条件2){
    对策3
}else{
    最后的对策
}
```

- 如果某一个条件满足, 那么只会执行它对应的对策, 后面的就不会检验
- 最后的else可有可无

I. Starter Introduction

二. 控制程序的执行

2. 做判断

例子: 找出两个数的最小值

```
if(条件1){
    对策1
}else if(条件2){
    对策2
}else if(条件2){
    对策3
}else{
    最后的对策
}
```

```
1 #include <iostream>
2 using namespace std;
3 int main(){
4     int a, b;
5     cin>>a>>b;
6     if(a>=b){
7         cout<<b<<" is the min of {a,b}\n";
8     }else if(b>=a){
9         cout<<a<<" is the min of {a,b}\n";
10    }
11    return 0;
12 }
```

```
zgw (e) base … pypsolve slides code ./a.out
1 1
1 is the min of {a,b}
zgw (e) base … pypsolve slides code ./a.out
3 4
3 is the min of {a,b}
```

I. Starter Introduction

二. 控制程序的执行

2. 做判断

例子: 找出三个数的最小值

```
3  int main(){
4      int a, b, c;
5      cin>>a>>b>>c;
6      int mi = INT_MAX;
7      if(a>b){
8          if(b>c){
9              mi = c;
10         }else{
11             mi = b;
12         }
13     }else{
14         if(a<c){
15             mi = a;
16         }else{
17             mi = c;
18         }
19     }
20     cout<<"The min val is "<<mi<<endl;
21     return 0;
22
23 }
```

```
if(条件1){
    对策1
}else if(条件2){
    对策2
}else if(条件2){
    对策3
}else{
    最后的对策
}
```

```
zgw (e) base … pypsolve slides code
1 2 3
The min val is 1
zgw (e) base … pypsolve slides code
zgw (e) base … pypsolve slides code
114 514 191
The min val is 114
zgw (e) base … pypsolve slides code
1 1 1
The min val is 1
```

I. Starter Introduction

二. 控制程序的执行

2. 做判断

问: 我们如果要寻找一列数(数组)的最小值怎么办?

```
if(条件1){
    对策1
}else if(条件2){
    对策2
}else if(条件2){
    对策3
}else{
    最后的对策
}
```

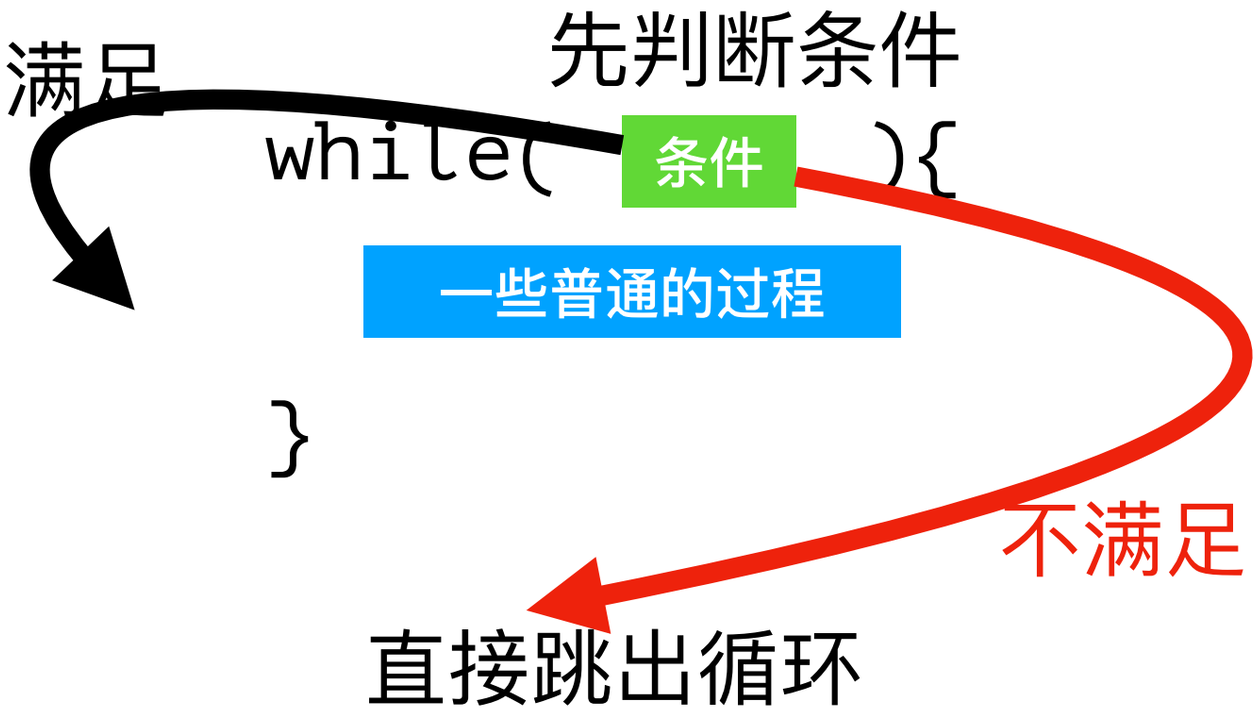
发现: 如果按照这种方式, 要写很多的判断

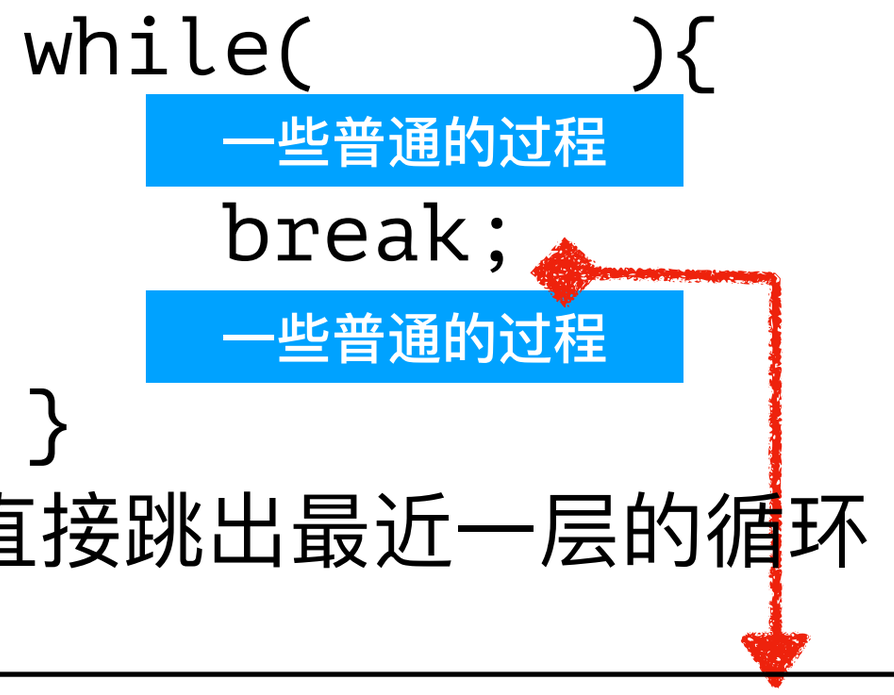
I. Starter Introduction

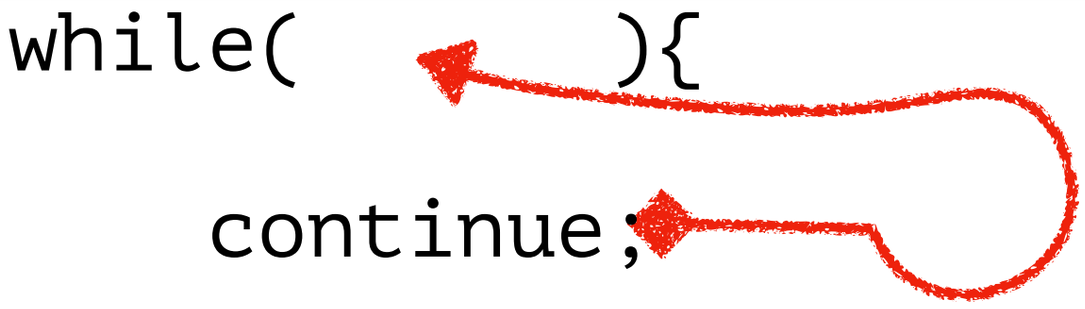
二. 控制程序的执行

loop

3. 做循环







I. Starter Introduction

二. 控制程序的执行

3. 做循环

例子: 找出n个数的最小值

```cpp
#define MAXN 101
int main(){
    int n;
    int a[MAXN];
    cin>>n;
    int counter = 0;
    while(counter < n){
        cin>>a[counter];
        counter = counter +1;
    }
    counter = 0;
    int mi = INT_MAX;
    while(counter < n){
        if(mi>a[counter]){
            mi = a[counter];
        }
        counter = counter + 1;
    }
    cout<<"The min val is "<< mi<<"\n";
    return 0;
}
```

```
zgw (e) base … pypsolve sli
3
1 2 3
The min val is 1
zgw (e) base … pypsolve slides code g++ find-minn.cpp && ./a.out
10
209 572 821 888 102 420 484 096 819 210
The min val is 96
zgw (e) base … pypsolve slides code                    master
```

I. Starter Introduction

二. 控制程序的执行

3. 做循环

但是有时候不是很方便, 能不能简化?

```
for(定义的变量； 判定条件； 结束的时候执行){
    一般的过程
}
```

```
 8      int counter = 0;
 9      while(counter < n){
10          cin>>a[counter];
11          counter = counter +1;
12      }
```

可以写作

一般会写成i

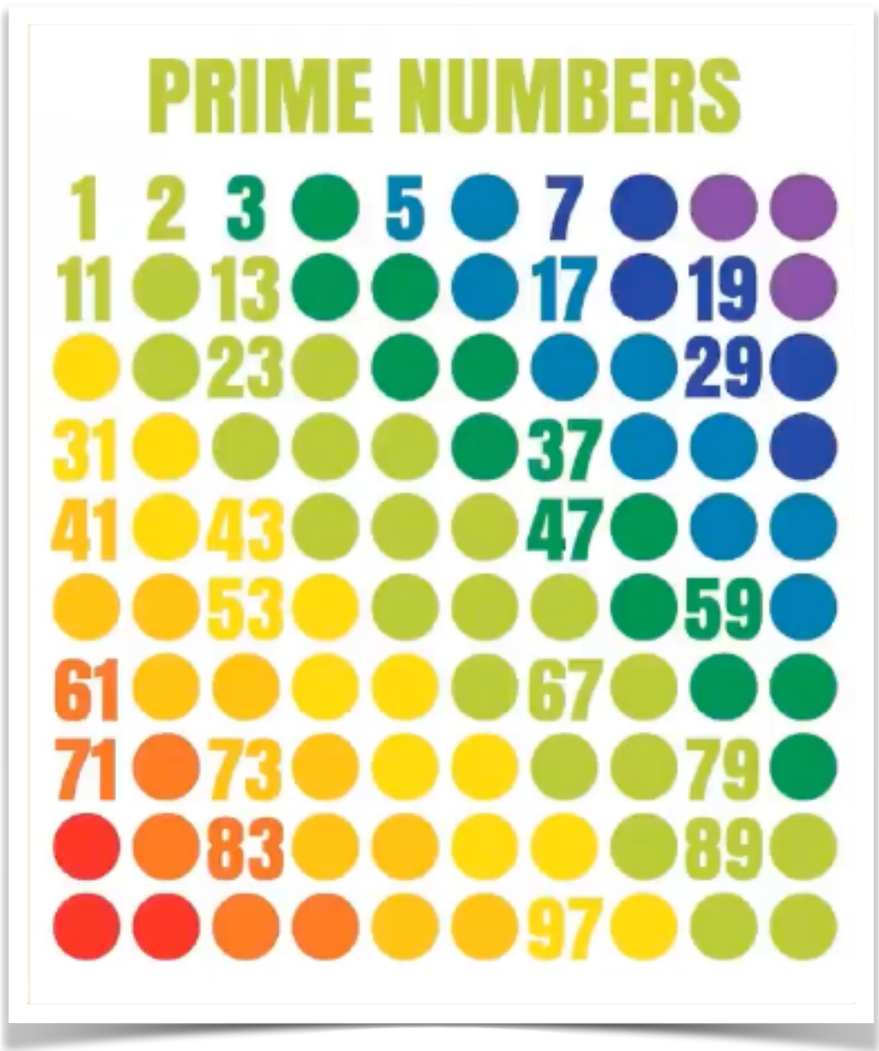
```
for(int counter=0; counter<n; counter++){
    cin>>a[counter];
}
```

I. Starter Introduction

二. 控制程序的执行

3. 做循环

例子: 找质数 *prime*



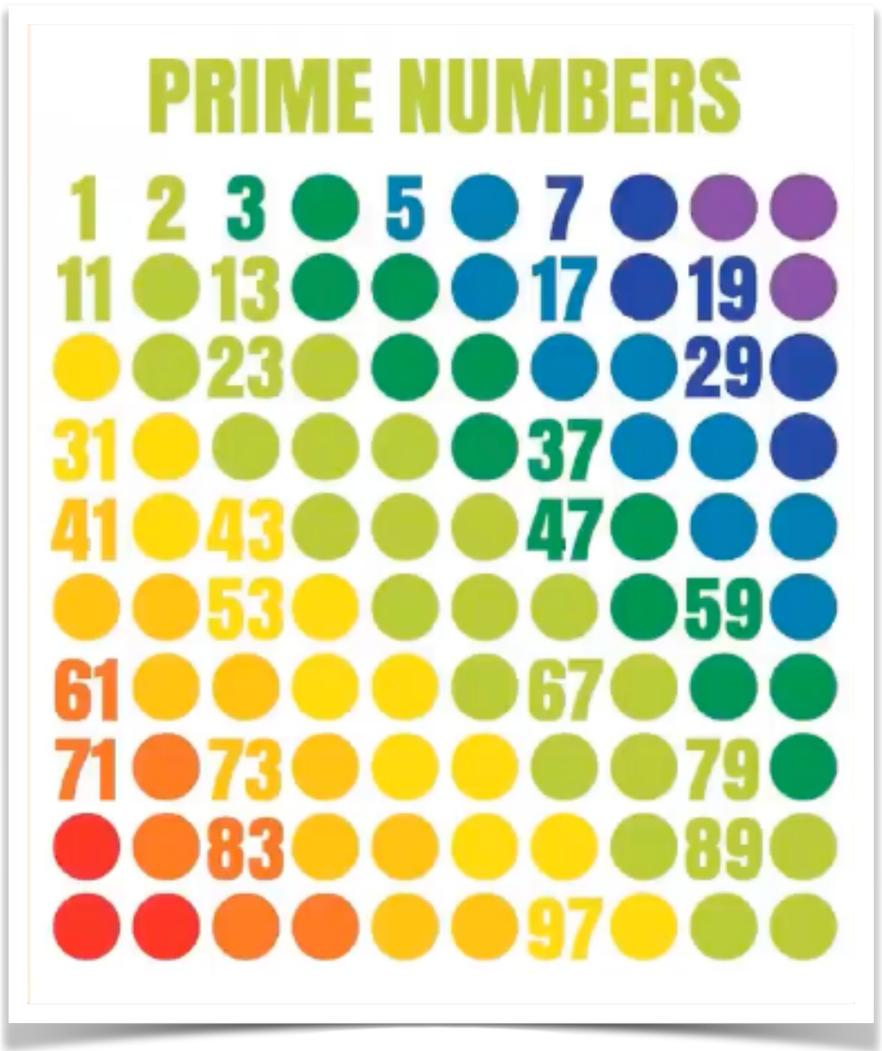

```
1   #include <iostream>
2   using namespace std;
3   #define N 101
4
5   int prime[101], notp[101];
6   int main(){
7       for(int i=2;i<N;i++){
8           if(notp[i]==0){
9               cout<<"Number "<< i <<" is a prime!\n";
10              for(int j=2*i; j<=N;j+=i){
11                  notp[j] = 1;
12              }
13          }
14      }
15  }
```

I. Starter Introduction

二. 控制程序的执行

3. 做循环

例子: 找质数; 扩展, 如果把它储存在数组里面怎么办?




```cpp
#include <iostream>
using namespace std;
#define N 101

int prime[N], notp[N];

int s[N], tp=0;
int main(){
    for(int i=2;i<N;i++){
        if(notp[i]==0){
            s[++tp] = i;
            for(int j=2*i; j<=N;j+=i){
                notp[j] = 1;
            }
        }
    }
    for(int i=1;i<=tp;i++) cout<<s[i]<<" ";
    return 0;
}
```

```
2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97
```

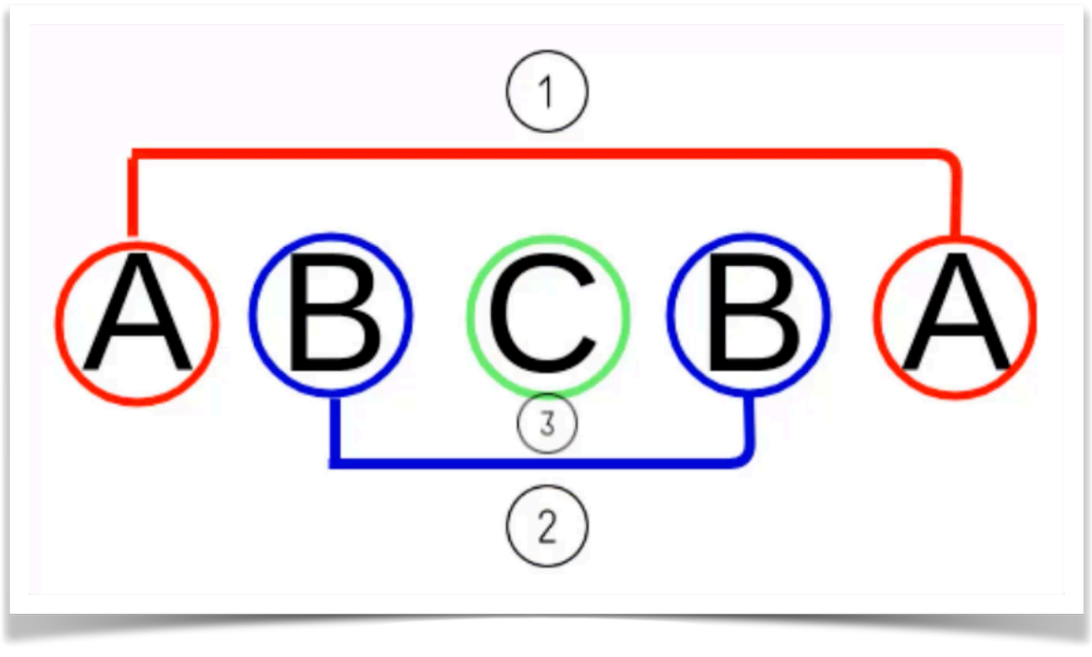
I. Starter Introduction

二. 控制程序的执行

3. 做循环

palindrome

例子: 判断回文




```
al.cpp > main()
1  #include <iostream>
2  #include <cstring>
3  using namespace std;
4  #define N 101
5
6  char s[N];
7  int main(){
8      cin>>s;
9      int n = strlen(s);
10     bool ok = 1;
11     for(int i=0, j=n-1; i!=j; i++, j--){
12         if(s[i]!=s[j]){
13             ok = 0;
14             break;
15         }
16     }
17     cout<< (ok? "Yes" : "No")<<endl;
18 }
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

zgw (e) base ~ program cobj  g++ al.cpp
zgw (e) base ~ program cobj  ./a.out
12321
Yes
zgw (e) base ~ program cobj  ./a.out
4321
No
zgw (e) base ~ program cobj
```

I. Starter Introduction

二. 控制程序的执行

3. 做循环

怎么判断for循环有没有循环完而没有用break提前跳出?

A: 变量放在全局, 看一下是不是满足让循环最后失效的值.

```
int i=1;
for(i=0;i<=10; i++){
   …
}
if(i==11) {
   //OK
}else{
   //Not OK
}
```

I. Starter Introduction

二. 控制程序的执行

* 变量的生存周期

```
int a = 1;
int a = 1;
Error: Redeclaration of variable a.
```

```
int a;
{ int a; a=1;}
```

OK

I. Starter Introduction

至此为止, 理论上, 你已经可以写出任何程序了.

解决程序阅读问题的手段:

- 当前有哪些变量?
- 这一行执行完之后哪些变量发生了变化?
- 下一步执行哪一行?

首先吐槽一下发的试卷:

- 字体不是等宽的, 令人血压升高;
- 奇怪的代码格式, 看起来很不舒服.

总结: 一看就不是计算机系写出来的代码...

I. Starter Introduction

至此为止, 理论上, 你已经可以写出任何程序了.

解决程序阅读问题的手段:

- 当前有哪些变量?
- 这一行执行完之后哪些变量发生了变化?
- 下一步执行哪一行?

(6)若有定义 int a[3][4]={{1,2},{3}}; 执行语句 cout<< a[1][1]<<endl; 后输出为 ( )

A. 0 B. 1 C. 2 D. 3

(8)若 int f(int n) { int i; for(i=n;i>0&&i*i!=n;i--) ; return i; } 则执行 cout<<f(16);后输出为 ( )

A. -1 B. 0 C. 4 D. 256

```
zgw (e) base ~ program cobj g++ test.cpp && ./a.out
test.cpp:5:35: warning: for loop has empty body [-Wempty-body]
    for(i=n; i>0 && i*i!=n ; i--) ; return i;
                                  ^

test.cpp:5:35: note: put the semicolon on a separate line to silence this warning
1 warning generated.
4%
```

I. Starter Introduction

至此为止, 理论上, 你已经可以写出任何程序了.

解决程序阅读问题的手段:

- 当前有哪些变量?
- 这一行执行完之后哪些变量发生了变化?
- 下一步执行哪一行?

1.[程序] # include <iostream>

```
using namespace std;
int main( )
{   int x=10,y=2,z=3;   cout<<x<<y<<z<<endl;
    {  int y=7,z;   z=x+y;   cout<<x<<y<<z<<endl;  }
    cout<<x<<y<<z<<endl;   return 0;
}
```

输出的三行各为_____, _____, _____。

```
zgw (e) base ~ program cobj g++ test.cpp && ./a.out
1023
10717
1023
```

I. Starter Introduction

至此为止, 理论上, 你已经可以写出任何程序了.

解决程序阅读问题的手段:

- 当前有哪些变量?
- 这一行执行完之后哪些变量发生了变化?
- 下一步执行哪一行?

4.[程序] # include <iostream>
using namespace std;
int main( )
{ int a[7]={1,3,5,7,11,13,17} , i , k ;
cout<<a[5]<<endl;
for (k=i=0 ; i<7 ; i++) { a[i]=k; k=(k+4)%7; }
cout<<a[5]<<endl;
for (k=i=0 ; i<7 ; i++) { a[k]=i; k=(k+4)%7; }
cout<<a[5]<<endl;
return 0;
}
输出的三行各为_____, _____

```
zgw (e) base ~ program cobj g++ test.cpp && ./a.out
13
6
3
```

```cpp
#include <iostream>
using namespace std;
int main(){
    int a[7] = {1,3,5,7,11,13,17}, i, k;
    cout<<a[5]<<endl;
    for(k=i=0; i<7; i++){
        a[i] = k; k = (k+4)%7;
    }
    cout<<a[5]<<endl;
    for(k=i=0; i<7; i++){
        a[k] = i; k = (k+4)%7;
    }
    cout<<a[5]<<endl;

}
```

I. Starter Introduction

我们需要好好管理我们的思维!

The major cause of the software crisis is that the machines have become several orders of magnitude more powerful! To put it quite bluntly: as long as there were no machines, programming was no problem at all; when we had a few weak computers, programming became a mild problem, and now we have gigantic computers, programming has become an equally gigantic problem.

— E.W. Dijkestra

I. Starter Introduction

三. 加强思维的管理

1. 函数 (*Function*): 整合相似过程

例子: 搜索一段文本中Fuhai的个数和Yanyan的个数.

```
int srch(string text, string tofd)
```

然后下面的所有的text和tofd就当做已经知道的变量来用

(1) 最外层的逗号分开, 丢进去的个数和函数参数 (*parameter*) 个数必须一样, 执行完了时候替换成一个对应返回值类型的字面量 (*literal*)

```
srch("Zhu Fuhai said that this is only a small test", "Fuhai"); →执行完了变成1
srch("Zhu Fuhai said that this is only a small test"); →编译错误!
```

I. Starter Introduction

三. 加强思维的管理

1. 函数: 整合相似过程

(2) 如果函数有相同的名字, 并且参数的个数或者类型不完全一样, 那么电脑会根据传入的参数自动选择执行哪个函数.

```
int a(int a, int b){…}
int a(int a, char b, int c){…}
```

√

```
int a(int a, int d, int f){…}
int a(int a, int b, int c){…}
```

x

I. Starter Introduction

三. 加强思维的管理

1. 函数: 整合相似过程

(9) 若定义了重载函数 int f(int x) { return x*2; }
double f(double x) { return x+3.5; }
则下列使用错误的是 ( )
A. cout<<f(1); B. cout<<f(0.2); C. cout<<f(0.2,1); D. cout<<f((0.2,1));

2.[程序] # include <iostream>
using namespace std;
int f (int x) { int y=1; y*=x; return y; }
int g (int x) { static int y=1; y*=x; return y; }
int main( ) { cout<<f(6)<<g(6)<<endl; cout<<f(8)<<g(8)<<endl; return 0; }
输出的两行各为_____, _____。

```cpp
#include <iostream>
using namespace std;
int f(int x){
    int y=1; y*=x; return y;
}
int g(int x){
    static int y=1; y*=x; return y;
}
int main(){
    cout<<f(6)<<g(6)<<endl;
    cout<<f(8)<<g(8)<<endl;
}
```

static不试图销毁变量

I. Starter Introduction

三. 加强思维的管理

1. 函数: 整合相似过程

这里有一个例外, 这个已经有默认的值了. 不传也是可以的

3.[程序] # include <iostream>
using namespace std;
void f (int n , int id=2)
{ if(n/id) f(n/id,id); cout<<n%id; }
int main ( )
{ f(123); cout<<endl; f(123,8); cout<<endl; return 0; }

输出的两行各为_____,_____。

```cpp
#include <iostream>
using namespace std;
int f(int x, int e=1, int z=4, int w){
    cout<<x<<" "<<e<<" "<<z<<" "<<w<<" "<<endl;
    return 0;
}
int main(){
    cout<<f(6, 1, 2, 3);
}
```

```
zgw (e) base ~ program cobj g++ test.cpp && ./a.out
test.cpp:3:36: error: missing default argument on parameter 'w'
int f(int x, int e=1, int z=4, int w){
                                   ^
1 error generated.
```

I. Starter Introduction

三. 加强思维的管理

1. 函数: 整合相似过程

问题类型: 补全程序

阅读别人代码这方面, 如果另一个人写得很丑的话, 我也做不了/很难做.

- 看一下大体的框架
- 看一下和目标还差多远
- 试图填入一些内容

I. Starter Introduction

三. 加强思维的管理

5.[程序] 下列程序输出 0°~90° 的正弦值，计算正弦值的近似公式如下：

$$\sin(x)\approx x-\frac{1}{3!}x^3+\frac{1}{5!}x^5-\frac{1}{7!}x^7+\cdots+\frac{(-1)^n}{(2n+1)!}x^{2n+1};0\le x\le\frac{\pi}{2}$$，其中$\left|\frac{x^{2n+1}}{(2n+1)!}\right|\ge10^{-7}$而$\left|\frac{x^{2n+3}}{(2n+3)!}\right|<10^{-7}$。

sin.cpp > main()

```
1   #include <iostream>
2   using namespace std;
3   const double pi = 3.141592653589793;
4   double SIN(double x) {
5       double s = 1, y, t;
6       int n;
7       x = x - int(x / (2 * pi)) * 2 * pi + 3 * pi;
8       x = x - int(x / (2 * pi)) * 2 * pi - pi;
9       if (x < 0) { x = -x; s = -1; }
10      if (x > pi / 2) x = pi - x;
11      y = 0;
12      for (  1  ;t > 1e-7;n++) {
13          y += s * t;
14          2
15          t *= x * x / (2 * n + 2) / (2 * n + 3);
16
17      }
18      3
19  }
20  int main() {
21      for (int i = 0;i <= 90;i++)
22          cout << "sin(" << i << ")= " << SIN(  4  ) << endl;
23      return 0;
24  }
25
```

注意n是弧度制

L7: $x := x - \lfloor\frac{x}{2\pi}\rfloor 2\pi + 3\pi$

L8: $x := x - \lfloor\frac{x}{2\pi}\rfloor 2\pi - \pi$

做$[-\pi,\pi]$的限制(虽然这个方法不太好)

I. Starter Introduction

三. 加强思维的管理

5.[程序] 下列程序输出 0°~90° 的正弦值，计算正弦值的近似公式如下：

$$\sin(x) \approx x - \frac{1}{3!}x^3 + \frac{1}{5!}x^5 - \frac{1}{7!}x^7 + \cdots + \frac{(-1)^n}{(2n+1)!}x^{2n+1}; 0 \le x \le \frac{\pi}{2}$$，其中 $\left|\frac{x^{2n+1}}{(2n+1)!}\right| \ge 10^{-7}$ 而 $\left|\frac{x^{2n+3}}{(2n+3)!}\right| < 10^{-7}$。

sin.cpp > main()

```
1   #include <iostream>
2   using namespace std;
3   const double pi = 3.141592653589793;
4   double SIN(double x) {
5       double s = 1, y, t;
6       int n;
7       x = x - int(x / (2 * pi)) * 2 * pi + 3 * pi;
8       x = x - int(x / (2 * pi)) * 2 * pi - pi;
9       if (x < 0) { x = -x; s = -1; }
10      if (x > pi / 2) x = pi - x;
11      y = 0;
12      for (  1  ;t > 1e-7;n++) {
13          y += s * t;
14          2
15          t *= x * x / (2 * n + 2) / (2 * n + 3);
16
17      }
18          3
19  }
20  int main() {
21      for (int i = 0;i <= 90;i++)
22          cout << "sin(" << i << ")= " << SIN(  4  ) << endl;
23      return 0;
24  }
25
```

观察L12-L15要计算表达式

分别有符号，相乘，累加

*t*保留了前项与后项的关系

I. Starter Introduction

三. 加强思维的管理

5.[程序] 下列程序输出 0°~90° 的正弦值，计算正弦值的近似公式如下：

$$\sin(x) \approx x - \frac{1}{3!}x^3 + \frac{1}{5!}x^5 - \frac{1}{7!}x^7 + \cdots + \frac{(-1)^n}{(2n+1)!}x^{2n+1}; 0 \le x \le \frac{\pi}{2}$$，其中 $\left|\frac{x^{2n+1}}{(2n+1)!}\right| \ge 10^{-7}$ 而 $\left|\frac{x^{2n+3}}{(2n+3)!}\right| < 10^{-7}$。

sin.cpp > main()

```cpp
1   #include <iostream>
2   using namespace std;
3   const double pi = 3.141592653589793;
4   double SIN(double x) {
5       double s = 1, y, t;
6       int n;
7       x = x - int(x / (2 * pi)) * 2 * pi + 3 * pi;
8       x = x - int(x / (2 * pi)) * 2 * pi - pi;
9       if (x < 0) { x = -x; s = -1; }
10      if (x > pi / 2) x = pi - x;
11      y = 0;
12      for (    1    ;t > 1e-7;n++) {
13          y += s * t;
14          2
15          t *= x * x / (2 * n + 2) / (2 * n + 3);
16
17      }
18      3
19  }
20  int main() {
21      for (int i = 0;i <= 90;i++)
22          cout << "sin(" << i << ")= " << SIN(    4    ) << endl;
23      return 0;
24  }
25
```

观察L12-L15要计算表达式

分别有符号, 相乘, 累加

y负责把它最后累加起来, 因此它就是答案

I. Starter Introduction

三. 加强思维的管理

5.[程序] 下列程序输出 0°~90° 的正弦值，计算正弦值的近似公式如下：

$$\sin(x) \approx x - \frac{1}{3!}x^3 + \frac{1}{5!}x^5 - \frac{1}{7!}x^7 + \cdots + \frac{(-1)^n}{(2n+1)!}x^{2n+1}; 0 \le x \le \frac{\pi}{2}$$，其中 $\left|\frac{x^{2n+1}}{(2n+1)!}\right| \ge 10^{-7}$ 而 $\left|\frac{x^{2n+3}}{(2n+3)!}\right| < 10^{-7}$。

sin.cpp > main()

```
1   #include <iostream>
2   using namespace std;
3   const double pi = 3.141592653589793;
4   double SIN(double x) {
5       double s = 1, y, t;
6       int n;
7       x = x - int(x / (2 * pi)) * 2 * pi + 3 * pi;
8       x = x - int(x / (2 * pi)) * 2 * pi - pi;
9       if (x < 0) { x = -x; s = -1; }
10      if (x > pi / 2) x = pi - x;
11      y = 0;
12      for (      1      ;t > 1e-7;n++) {
13          y += s * t;
14          s = -s;
15          t *= x * x / (2 * n + 2) / (2 * n + 3);
16
17      }
18      return y;
19  }
20  int main() {
21      for (int i = 0;i <= 90;i++)
22          cout << "sin(" << i << ")= " << SIN(i * pi / 180) << endl;
23      return 0;
24  }
25
```

每一项的符号都会变化

调用一下sin(弧度)

观察L12-L15要计算表达式

分别有符号，相乘，累加

I. Starter Introduction

三. 加强思维的管理

5.[程序] 下列程序输出 0°~90° 的正弦值，计算正弦值的近似公式如下：

$$\sin(x) \approx x - \frac{1}{3!}x^3 + \frac{1}{5!}x^5 - \frac{1}{7!}x^7 + \cdots + \frac{(-1)^n}{(2n+1)!}x^{2n+1}; 0 \le x \le \frac{\pi}{2}$$，其中 $\left|\frac{x^{2n+1}}{(2n+1)!}\right| \ge 10^{-7}$ 而 $\left|\frac{x^{2n+3}}{(2n+3)!}\right| < 10^{-7}$。

sin.cpp > main()

```cpp
#include <iostream>
using namespace std;
const double pi = 3.141592653589793;
double SIN(double x) {
    double s = 1, y, t;
    int n;
    x = x - int(x / (2 * pi)) * 2 * pi + 3 * pi;
    x = x - int(x / (2 * pi)) * 2 * pi - pi;
    if (x < 0) { x = -x; s = -1; }
    if (x > pi / 2) x = pi - x;
    y = 0;
    for (1;t > 1e-7;n++) {
        y += s * t;
        t *= -x / (2 * n + 2) / (2 * n + 3);
    }
    return y;
}
int main() {
    for (int i = 0;i <= 90;i++)
        cout << "sin(" << i << ")= " << SIN(i * pi / 180) << endl;
    return 0;
}
```

观察L12-L15要计算表达式

分别有符号，相乘，累加

实在当前点的展开，于是现在让t等于x，顺便让n为0

I. Starter Introduction

三. 加强思维的管理

5.[程序] 下列程序输出 0°~90° 的正弦值，计算正弦值的近似公式如下：

$$\sin(x)\approx x-\frac{1}{3!}x^3+\frac{1}{5!}x^5-\frac{1}{7!}x^7+\cdots+\frac{(-1)^n}{(2n+1)!}x^{2n+1};0\le x\le\frac{\pi}{2}$$，其中$\left|\frac{x^{2n+1}}{(2n+1)!}\right|\ge10^{-7}$而$\left|\frac{x^{2n+3}}{(2n+3)!}\right|<10^{-7}$。

sin.cpp > main()

```cpp
#include <iostream>
using namespace std;
const double pi = 3.141592653589793;
double SIN(double x) {
    double s = 1, y, t;
    int n;
    x = x - int(x / (2 * pi)) * 2 * pi + 3 * pi;
    x = x - int(x / (2 * pi)) * 2 * pi - pi;
    if (x < 0) { x = -x; s = -1; }
    if (x > pi / 2) x = pi - x;
    y = 0;
    for (n = 0, t = x;t > 1e-7;n++) {
        y += s * t;
        s = -s;
        t *= x * x / (2 * n + 2) / (2 * n + 3);

    }
    return y;
}
int main() {
    for (int i = 0;i <= 90;i++)
        cout << "sin(" << i << ")= " << SIN(i * pi / 180) << endl;
    return 0;
}
```

观察状态机的执行

I. Starter Introduction

三. 加强思维的管理

6.[程序] 对一个输入的正整数 n，判断是否对所有 2~n-1 的数 a，满足 a^{n-1} %n=1

```
#include <iostream>
using namespace std;
int powerModn(int a,int n)       // 计算 a^(n-1) %n 的值，1≤n≤10^4
{     int i,y;      for(i=y=1;i<=n-1;i++)    y= _____ ;
}
int    main( )
{      _____ ;      cin>>n;
       for(a=2;a<n;a++) if(powerModn(a,n)!=1) break ;
       if(_____)    cout<< "式子对"<<a<< "不成立"<<endl;
       else          cout<< "式子对所有 2~n-1 的数成立"<<endl;
       return 0;
}
```

I. Starter Introduction

三. 加强思维的管理

6.[程序] 对一个输入的正整数 n，判断是否对所有 2~n-1 的数 a，满足 a^{n-1} %n=1

```
1    #include <iostream>
2    using namespace std;
3    int foo(int a, int n){
4        int i, y;
5        for(i=y=1; i<=n-1; i++)  1
6            2
7    }
8    int main(){
9        3
10       cin>>n;
11       for(a=2;a<n;a++) if(foo(a,n)!=1) break ;
12       if  4  ) cout<< "式子对"<<a<< "不成立"<<endl;
13       else cout<< "式子对所有 2~n-1 的数成立"<<endl;
14       return 0;
15   }
```

这里是一个幂次方，肯定有一个计数器和答案

I. Starter Introduction

三. 加强思维的管理

6.[程序] 对一个输入的正整数 n，判断是否对所有 2~n-1 的数 a，满足 a^{n-1} %n=1

```
1    #include <iostream>
2    using namespace std;
3    int foo(int a, int n){
4        int i, y;
5        for(i=y=1; i<=n-1; i++) y=a*y%n;
6        return y%n;
7    }
8    int main(){
9        3
10       cin>>n;
11       for(a=2;a<n;a++) if(foo(a,n)!=1) break ;
12       if 4 ) cout<< "式子对"<<a<< "不成立"<<endl;
13       else cout<< "式子对所有 2~n-1 的数成立"<<endl;
14       return 0;
15   }
```

(*)%n;

要定义下面的一些变量!

I. Starter Introduction

三. 加强思维的管理

6.[程序] 对一个输入的正整数 n，判断是否对所有 2~n-1 的数 a，满足 a^{n-1} %n=1

```
1    #include <iostream>
2    using namespace std;
3    int foo(int a, int n){
4        int i, y;
5        for(i=y=1; i<=n-1; i++) y=a*y%n;
6        return y%n;
7    }
8    int main(){
9        int n,a;
10       cin>>n;
11       for(a=2;a<n;a++) if(foo(a,n)!=1) break ;
12       if( 4 ) cout<< "式子对"<<a<< "不成立"<<endl;
13       else cout<< "式子对所有2~n-1的数成立"<<endl;
14       return 0;
15   }
```

判断上面的是不是循环完了

I. Starter Introduction

三. 加强思维的管理

6.[程序] 对一个输入的正整数 n，判断是否对所有 2~n-1 的数 a，满足 a^{n-1} %n=1

```
1    #include <iostream>
2    using namespace std;
3    int foo(int a, int n){
4        int i, y;
5        for(i=y=1; i<=n-1; i++) y=a*y%n;
6        return y%n;
7    }
8    int main(){
9        int n,a;
10       cin>>n;
11       for(a=2;a<n;a++) if(foo(a,n)!=1) break ;
12       if(a!=n) cout<< "式子对"<<a<< "不成立"<<endl;
13       else cout<< "式子对所有 2~n-1 的数成立"<<endl;
14       return 0;
15   }
```

I. Starter Introduction

三. 加强思维的管理

7.[程序] 在 10000 以内验证哥德巴赫猜想 (任何大于 3 的偶数必可表示成两个素数的和) 。

```
#include <iostream>
#include <cmath>
using namespace std;
bool    isPrime( _____ )   //判断大于 1 的整数 n 是否素数，是则返回 true，否则返回 false
{     int k,i;    for(k=sqrt(double(n)),i=2 ; i<=k&&n%i ; i++ );    return    i>k;    }
int    main( )
{     int k, i;
      for(k=4;k<=10000; _____ )
      {     for(i=2;i<=k/2;i++)    if( _____ )    break;
            if( _____ ) break;
      }
      if(k>10000)    cout<< "哥德巴赫猜想在 10000 以内成立"<<endl;
      else         cout<< "k="<<k-2<< "时，哥德巴赫猜想不成立"<<endl;        return 0;
}
```

I. Starter Introduction

三. 加强思维的管理

7.[程序] 在 10000 以内验证哥德巴赫猜想 (任何大于 3 的偶数必可表示成两个素数的和) 。

```
1    #include <iostream>
2    #include <cmath>
3    using namespace std;
4    bool isPrime( 1 ) {这里要传入一个参数, 看下文就行了
5        int k, i;
6        for (k = sqrt(double(n)), i = 2; i <= k && n % i; i++);
7        return i > k;
8    }
9    int main() {
10       int k, i;
11       for (k = 4;k <= 10000; 2 2) {
12           for (i = 2;i <= k / 2;i++) if ( 3 ) break;
13           if ( 4 break;
14       }
15       if (k > 10000) cout << "哥德巴赫猜想在 10000 以内成立" << endl;
16       else cout << "k=" << k - 2 << "时, 哥德巴赫猜想不成立" << endl;
17       return 0;
18   }
```

I. Starter Introduction

三. 加强思维的管理

7.[程序] 在 10000 以内验证哥德巴赫猜想 (任何大于 3 的偶数必可表示成两个素数的和) 。

```
1   #include <iostream>
2   #include <cmath>
3   using namespace std;
4   bool isPrime(int n) {
5       int k, i;
6       for (k = sqrt(double(n)), i = 2; i <= k && n % i; i++);
7       return i > k;
8   }
9   int main() {
10      int k, i;
11      for (k = 4;k <= 10000; 2) {
12          for (i = 2;i <= k / 2;i++) if ( 3 ) break;
13          if ( 4 break;
14      }
15      if (k > 10000) cout << "哥德巴赫猜想在 10000 以内成立" << endl;
16      else cout << "k=" << k - 2 << "时, 哥德巴赫猜想不成立" << endl;
17      return 0;
18  }
```

题目中的大于3的偶数

I. Starter Introduction

三. 加强思维的管理

7.[程序] 在 10000 以内验证哥德巴赫猜想 (任何大于 3 的偶数必可表示成两个素数的和) 。

```
1   #include <iostream>
2   #include <cmath>
3   using namespace std;
4   bool isPrime(int n) {
5       int k, i;
6       for (k = sqrt(double(n)), i = 2; i <= k && n % i; i++);
7       return i > k;
8   }
9   int main() {
10      int k, i;
11      for (k = 4;k <= 10000; k += 2) {
12          for (i = 2;i <= k / 2;i++) if (  3  ) break;
13          if (  4  break;
14      }
15      if (k > 10000) cout << "哥德巴赫猜想在 10000 以内成立" << endl;
16      else cout << "k=" << k - 2 << "时, 哥德巴赫猜想不成立" << endl;
17      return 0;
18  }
```

9=1+8=2+7=3+6=...4+5

分拆成了i和$n-i$两部分

I. Starter Introduction

三. 加强思维的管理

7.[程序] 在 10000 以内验证哥德巴赫猜想 (任何大于 3 的偶数必可表示成两个素数的和) 。

```
1   #include <iostream>
2   #include <cmath>
3   using namespace std;
4   bool isPrime(int n) {
5       int k, i;
6       for (k = sqrt(double(n)), i = 2; i <= k && n % i; i++);
7       return i > k;
8   }
9   int main() {
10      int k, i;
11      for (k = 4; k <= 10000; k += 2) {
12          for (i = 2;i <= k / 2;i++) if (isPrime(i) && isPrime(k - i)) break;
13          if (        4       break;
14      }
15      if (k > 10000) cout << "哥德巴赫猜想在 10000 以内成立" << endl;
16      else cout << "k=" << k - 2 << "时, 哥德巴赫猜想不成立" << endl;
17      return 0;
18  }
```

如果循环完了都没有找到合适的, break掉

I. Starter Introduction

三. 加强思维的管理

7.[程序] 在 10000 以内验证哥德巴赫猜想 (任何大于 3 的偶数必可表示成两个素数的和) 。

```
#include <iostream>
#include <cmath>
using namespace std;
bool isPrime(int n) {
    int k, i;
    for (k = sqrt(double(n)), i = 2; i <= k && n % i; i++);
    return i > k;
}
int main() {
    int k, i;
    for (k = 4;k <= 10000; k += 2) {
        for (i = 2;i <= k / 2;i++) if (isPrime(i) && isPrime(k - i)) break;
        if (i > k / 2) break;
    }
    if (k > 10000) cout << "哥德巴赫猜想在 10000 以内成立" << endl;
    else cout << "k=" << k - 2 << "时, 哥德巴赫猜想不成立" << endl;
    return 0;
}
```

I. Starter Introduction

三. 加强思维的管理

2． 糅合进行新的元素： struct

比如复数有实部和虚部, 有没有办法

```
struct 新结构的类型{
    新定义杂糅起来的一些变量， 方法…
};

//访问里面的： 新结构的类型.(正常语句);
```

I. Starter Introduction

三. 加强思

2． 糅合

比如

stru

};

//访

```cpp
#include <iostream>
using namespace std;

struct Complex{
    double real=0, imag=0;
    void print(){
        printf("%f+%fi\n",real, imag);
    }
};

Complex c;
int main(){
    c.imag = 2;
    c.print();
}
```

一个新的类型，里面也有对应的数据
和它里面的函数

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL zsh

```
0+0i
zgw (e) base ~ program cobj g++ struct.cpp && ./a.out
struct.cpp:5:16: warning: default member initializer for non-static data member is a C++11 extension [-Wc++11-extensions]
    double real=0, imag=0;
               ^
struct.cpp:5:24: warning: default member initializer for non-static data member is a C++11 extension [-Wc++11-extensions]
    double real=0, imag=0;
                   ^
2 warnings generated.
0.000000+2.000000i
zgw (e) base ~ program cobj
```

I. Starter Introduction

三. 加强思维的管理

10.[程序]　下列程序定义结构类型 stack 来描述存放 int 数据的栈。栈是一种只能在一批数据的末端存取的数据结构，在栈中末端称为栈顶。stack 成员中，数组 st 保存数据，tos 记录栈顶，即数组中最后一个数据的下标。函数 empty(a)判断 a 是否空栈，即 a 中没有数据。push(a,x)将数据 x 加入栈 a 中。pop(a)释放一个栈顶数据，即 tos 减 1。top(a)为取栈 a 的栈顶元素。主函数输入 10 个数据存入栈，并输出所保存的数据。

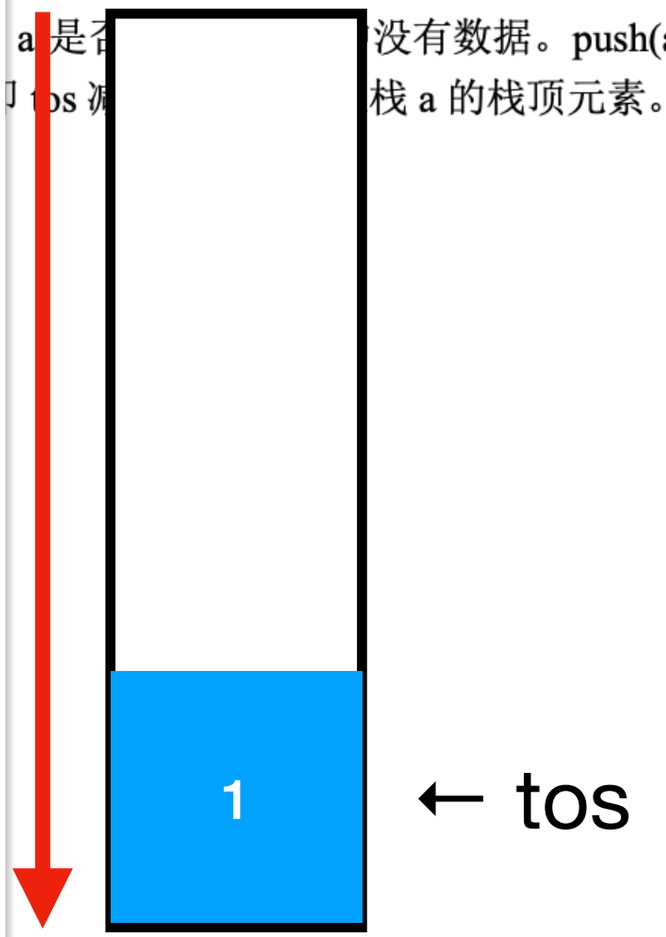
```
# include <iostream>
using namespace std;
struct stack    {    int st[200];    int tos;    };
bool empty(stack s)    //判断是否空栈
{    return s.tos<0;    }
stack push(stack s,int x)        //数据 x 加入栈 s 中
{      if(s.tos<199)    _____;    return s;    }
stack pop(stack s)      //释放一个栈顶数据
{      if(!empty(s))  _____;    return s;    }
int top(stack s)    //取栈顶数据
{      if(empty(s))    return 0x80000000;    else return s.st[s.tos];    }
int    main( )
{      stack    a={{0},-1};    int x,i;
       for(i=1;i<=10;i++)    {    cin>>x;    a=_____;    }
       while(!empty(a) )    {    cout<< top(a)<<endl;    a=_____;    }    return 0;
}
```

```
4   struct stack{
5       int st[200];
6       int tos;
7   };
8   bool empty(stack s){
9       return s.tos < 0;
10  }
11  stack push(stack s, int x){
12      if (s.tos < 199) 1
13      return s;
14  }
15  stack pop(stack s){
16      if (!empty(s)) s  2
17      return s;
18  }
19  int top(stack s){
20      if (empty(s)) return 0x80000000;
21      else return s.st[s.tos];
22
23  }
24  int main(){
25      stack a = {{0}, -1};
26      int x, i;
27      for (i = 1; i <= 10; i++){
28          cin >> x;
29          a =    3    ;
30      }
31      while (!empty(a)){
32          cout << top(a) << endl;
33          a =    4
34      }
35      return 0;
36  }
```

先增加再赋值

tion

数据的栈。栈是一种只能在一批数据的末端存
员中，数组 st 保存数据，tos 记录栈顶，即数
a 是否 没有数据。push(a,x)将数
tos 减 栈 a 的栈顶元素。主函数

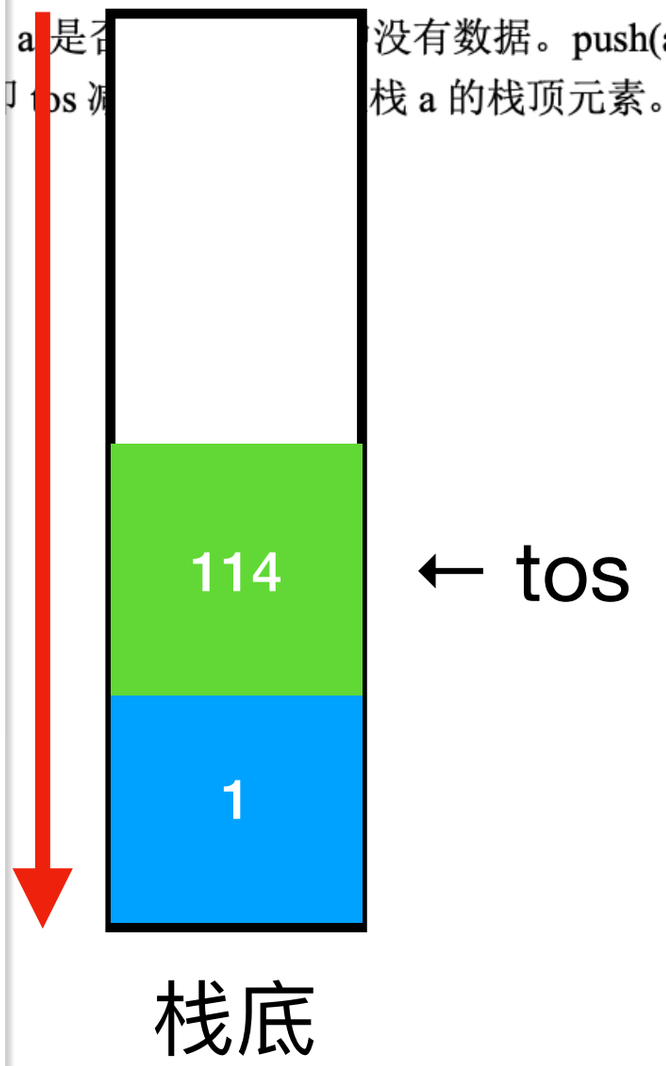


```
4   struct stack{
5       int st[200];
6       int tos;
7   };
8   bool empty(stack s){
9       return s.tos < 0;
10  }
11  stack push(stack s, int x){
12      if (s.tos < 199) s.st[++s.tos] = x;
13      return s;
14  }
15  stack pop(stack s){
16      if (!empty(s)) s      2
17      return s;
18  }
19  int top(stack s){
20      if (empty(s)) return 0x80000000;
21      else return s.st[s.tos];
22
23  }
24  int main(){
25      stack a = {{0}, -1};
26      int x, i;
27      for (i = 1; i <= 10; i++){
28          cin >> x;
29          a =      3      ;
30      }
31      while (!empty(a)){
32          cout << top(a) << endl;
33          a =      4
34      }
35      return 0;
36  }
```

让tos往下移动

tion

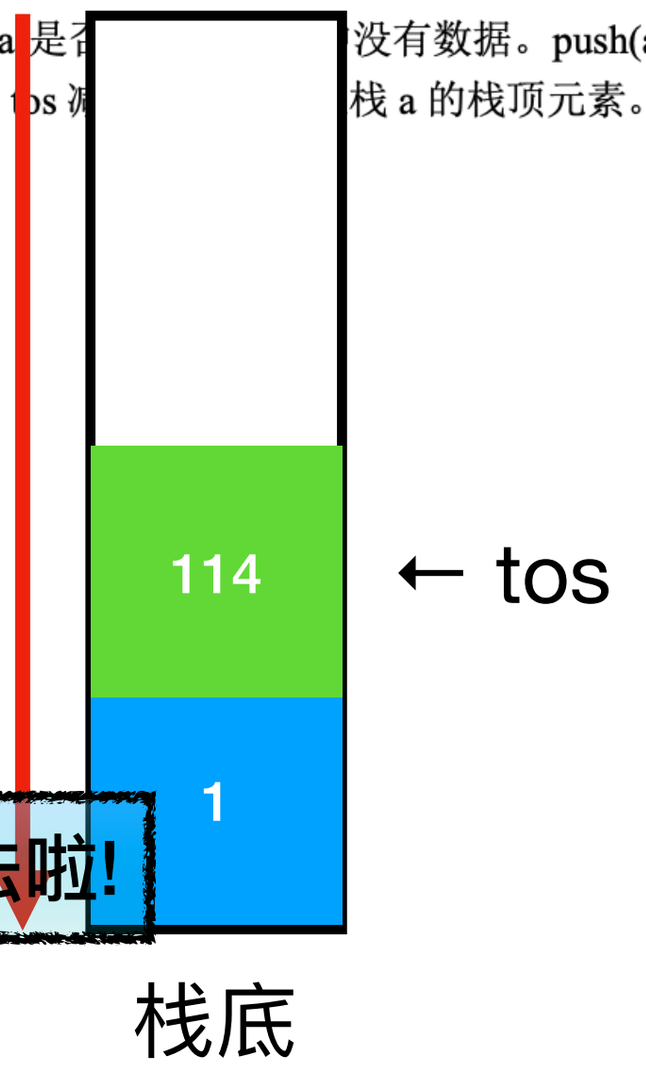
数据的栈。栈是一种只能在一批数据的末端存
员中，数组 st 保存数据，tos 记录栈顶，即数
a 是否 没有数据。push(a,x)将数
tos 减 栈 a 的栈顶元素。主函数



```
4   struct stack{
5       int st[200];
6       int tos;
7   };
8   bool empty(stack s){
9       return s.tos < 0;
10  }
11  stack push(stack s, int x){
12      if (s.tos < 199) s.st[++s.tos] = x;
13      return s;
14  }
15  stack pop(stack s){
16      if (!empty(s)) s.tos--;
17      return s;
18  }
19  int top(stack s){
20      if (empty(s)) return 0x80000000;
21      else return s.st[s.tos];
22
23  }
24  int main(){
25      stack a = {{0}, -1};
26      int x, i;
27      for (i = 1; i <= 10; i++){
28          cin >> x;
29          a =      3      ;
30      }
31      while (!empty(a)){
32          cout << top(a) << endl;
33          a =      4      ;
34      }
35      return 0;
36  }
```

tion

数据的栈。栈是一种只能在一批数据的末端存
员中，数组 st 保存数据，tos 记录栈顶，即数
a 是否 没有数据。push(a,x)将数
tos 减 栈 a 的栈顶元素。主函数

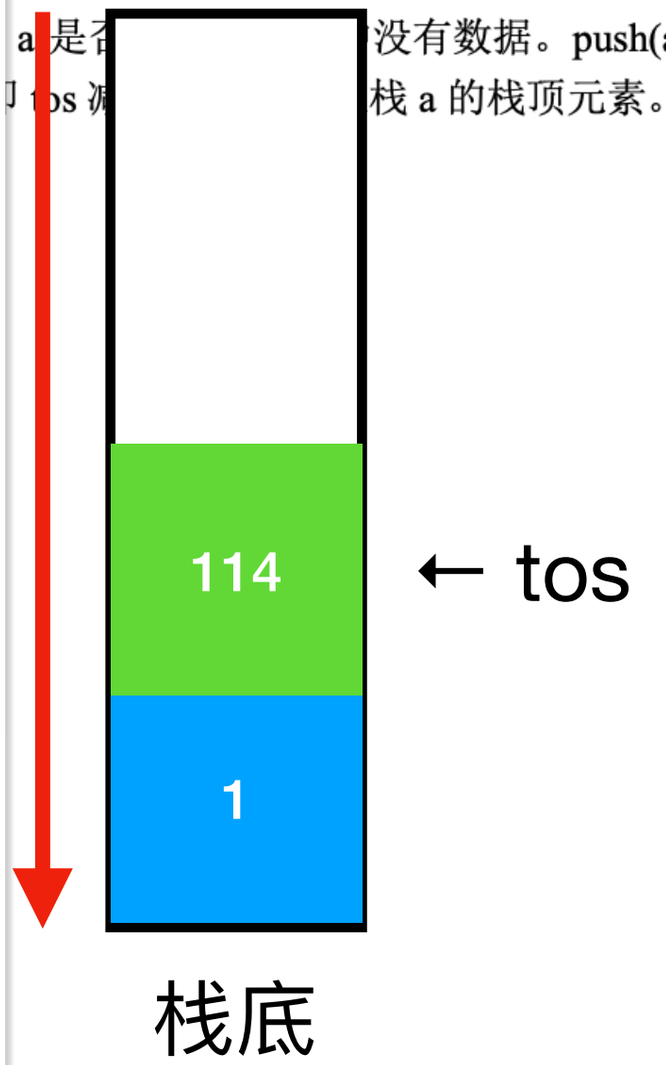


剩下的都是调用方法啦!

```
4   struct stack{
5       int st[200];
6       int tos;
7   };
8   bool empty(stack s){
9       return s.tos < 0;
10  }
11  stack push(stack s, int x){
12      if (s.tos < 199) s.st[++s.tos] = x;
13      return s;
14  }
15  stack pop(stack s){
16      if (!empty(s)) s.tos--;
17      return s;
18  }
19  int top(stack s){
20      if (empty(s)) return 0x80000000;
21      else return s.st[s.tos];
22
23  }
24  int main(){
25      stack a = {{0}, -1};
26      int x, i;
27      for (i = 1; i <= 10; i++){
28          cin >> x;
29          a = push(a, x);
30      }
31      while (!empty(a)){
32          cout << top(a) << endl;
33          a = pop(a);
34      }
35      return 0;
36  }
```

tion

数据的栈。栈是一种只能在一批数据的末端存
员中，数组 st 保存数据，tos 记录栈顶，即数
a 是否 没有数据。push(a,x)将数
tos 减 栈 a 的栈顶元素。主函数



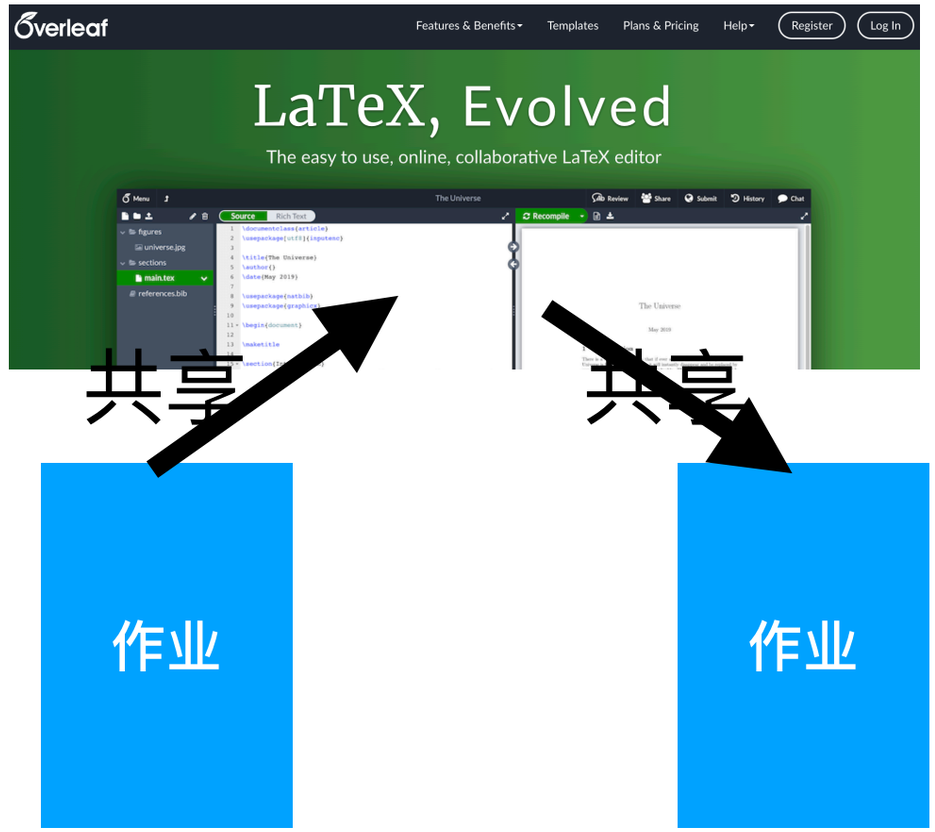
I. Starter Introduction

四. 一个鲁莽的行为

1． 变量的两种引用形式 *Reference* 指针 *Pointer*

(1) 引用形式: 复印机? 共享文档?

By value *By address*

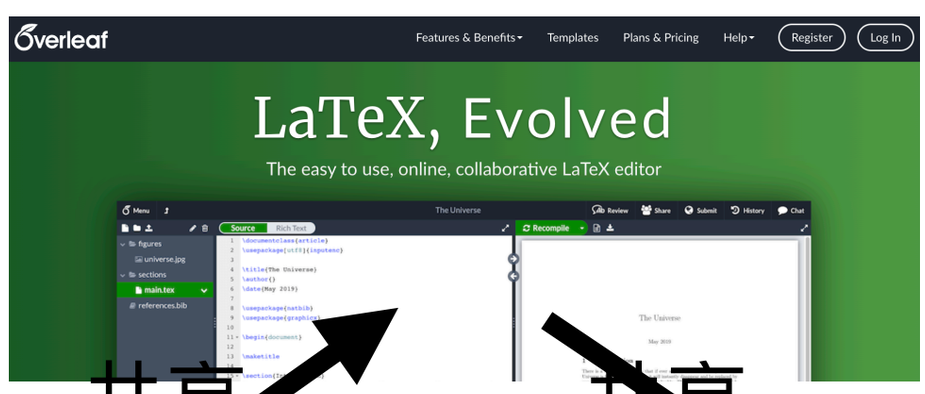


I. Starter Introduction

四. 一个鲁莽的行为

1． 变量的两种引用形式　　指针

(1) 引用形式: 复印机? 共享文档?



I. Starter Introduction

四. 一个鲁莽的行为

1． 变量的两种引用形式　　指针

(2) 回到变量—

一个变量有type, value和address

问: 怎么取变量的地址(门牌号)? — 使用&

I. Starter Introdu

四. 一个鲁莽的行为

1． 变量的两种引用形式 指针

(2) 回到变量—

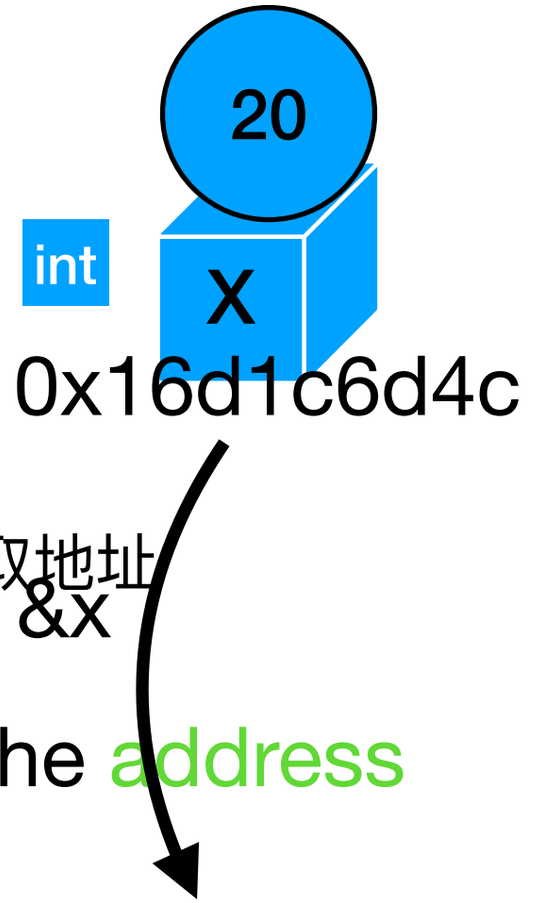
一个变量有type, valu

问: 怎么取变量的地址(门牌号)? — 使用&

```cpp
#include <iostream>
using namespace std;
int main(){
    int x=20;
    cout<<"The value of x is "<<x<<endl;
    cout<<"The address of x is "<<&x<<endl;
    //                              ^^ This is a pointer.

    printf("We use printf for clearer view\n");
    printf("The value of x is %d\n", x);
    printf("The address of x is %p", &x);
}
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
zgw (e) base ~ program cobj g++ addr.cpp && ./a.out
The value of x is 20
The address of x is 0x16d1c6d4c
We use printf for clearer view
The value of x is 20
The address of x is 0x16d1c6d4c
```

I. Starter Introduction



四. 一个鲁莽的行为

1． 变量的两种引用形式　指针

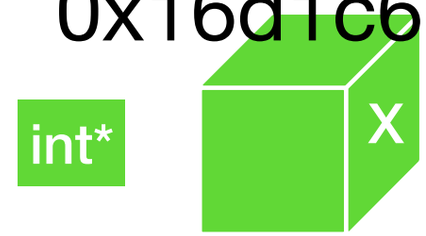
(3) 指针:

定义: A pointer is a variable that contains the address of a variable.

```
3   int main(){
4       int x=20;
5       cout<<"The value of x is "<<x<<endl;
6       cout<<"The address of x is "<<&x<<endl;
7       /                                  ^^ This is a pointer.
8
9       printf("We use printf for clearer view\n");
10      printf("The value of x is %d\n", x);
11      int* p2 = &x;
12      printf("The address of p2 is %p", &p2);
13  }
```

int *p2

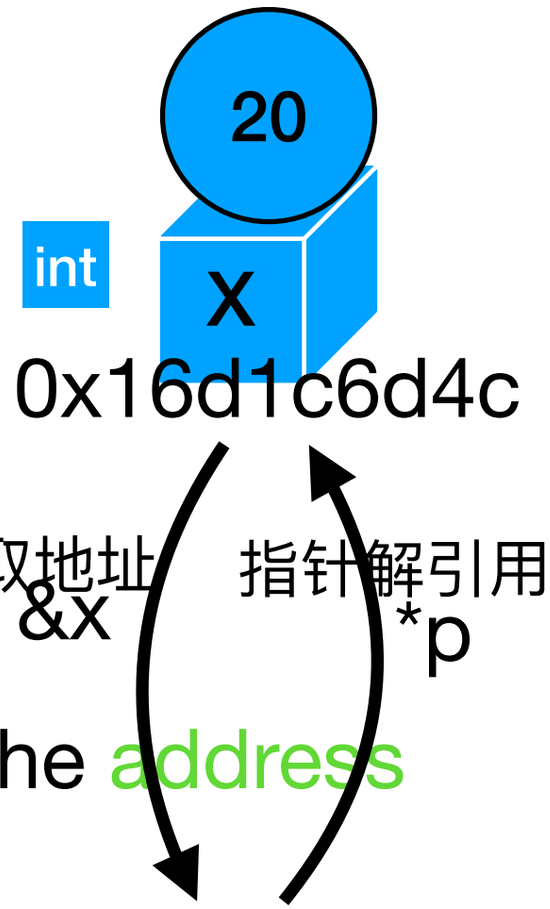
```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL

zgw (e) base ~ program cobj  g++ addr.cpp && ./a.out
The value of x is 20
The address of x is 0x16efdad4c
We use printf for clearer view
The value of x is 20
The address of p2 is 0x16efdad40%
```



他也有自己的门牌号

0x16efdad40

I. Starter Introduction



四. 一个鲁莽的行为

1． 变量的两种引用形式　指针

(3) 指针:

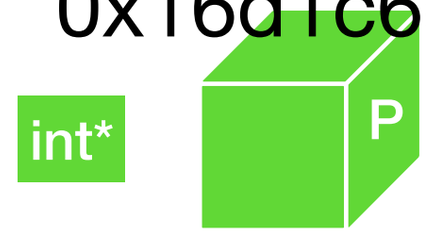
定义: A pointer is a variable that contains the address of a variable.

addr.cpp > main()

```
1   #include <iostream>
2   using namespace std;
3   int main(){
4       int x=20;
5       cout<<"The value of x is "<<x<<endl;
6       int *p2 = &x;
7       int **p3 = &p2;
8       **p3 = 1;
9       printf("The value of x is %d\n", x);
10  }
```

PROBLEMS　OUTPUT　DEBUG CONSOLE　TERMINAL

```
zgw (e) base ~ program cobj  g++ addr.cpp && ./a.out
The value of x is 20
The value of x is 1
```



他也有自己的门牌号

0x16efdad40

I. Starter Introduction

四. 一个鲁莽的行为

1． 变量的两种引用形式　指针

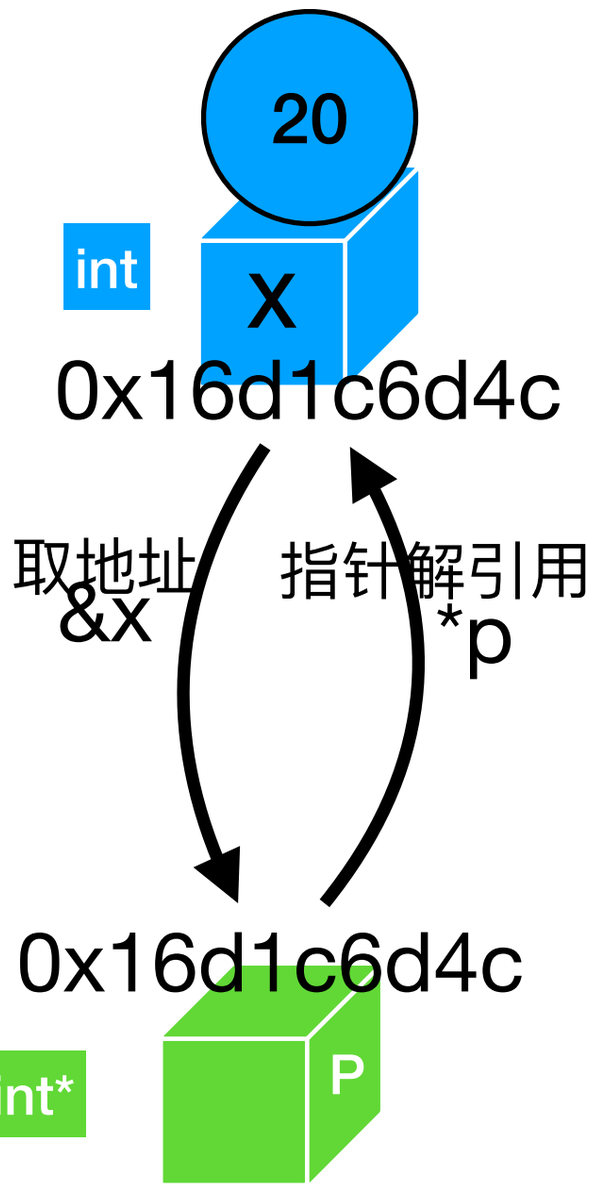
例子: 交换变量

swap.cpp > main()

```
9
10  void swp(int *l, int *r){
11      int tmp = *l;
12      *l=*r;
13      *r=tmp;
14  }
15
16  int main(){
17      int a=1, b=2;
18      swp(&a, &b);
19      cout<<a<<" "<<b<<endl;
20  }
```

PROBLEMS　OUTPUT　DEBUG CONSOLE　TERMINAL

```
zgw (e) base ~ program cobj  g++ swap.cpp && ./a.out
2 1
zgw (e) base ~ program cobj
```



I. Starter Introduction

四. 一个鲁莽的行为

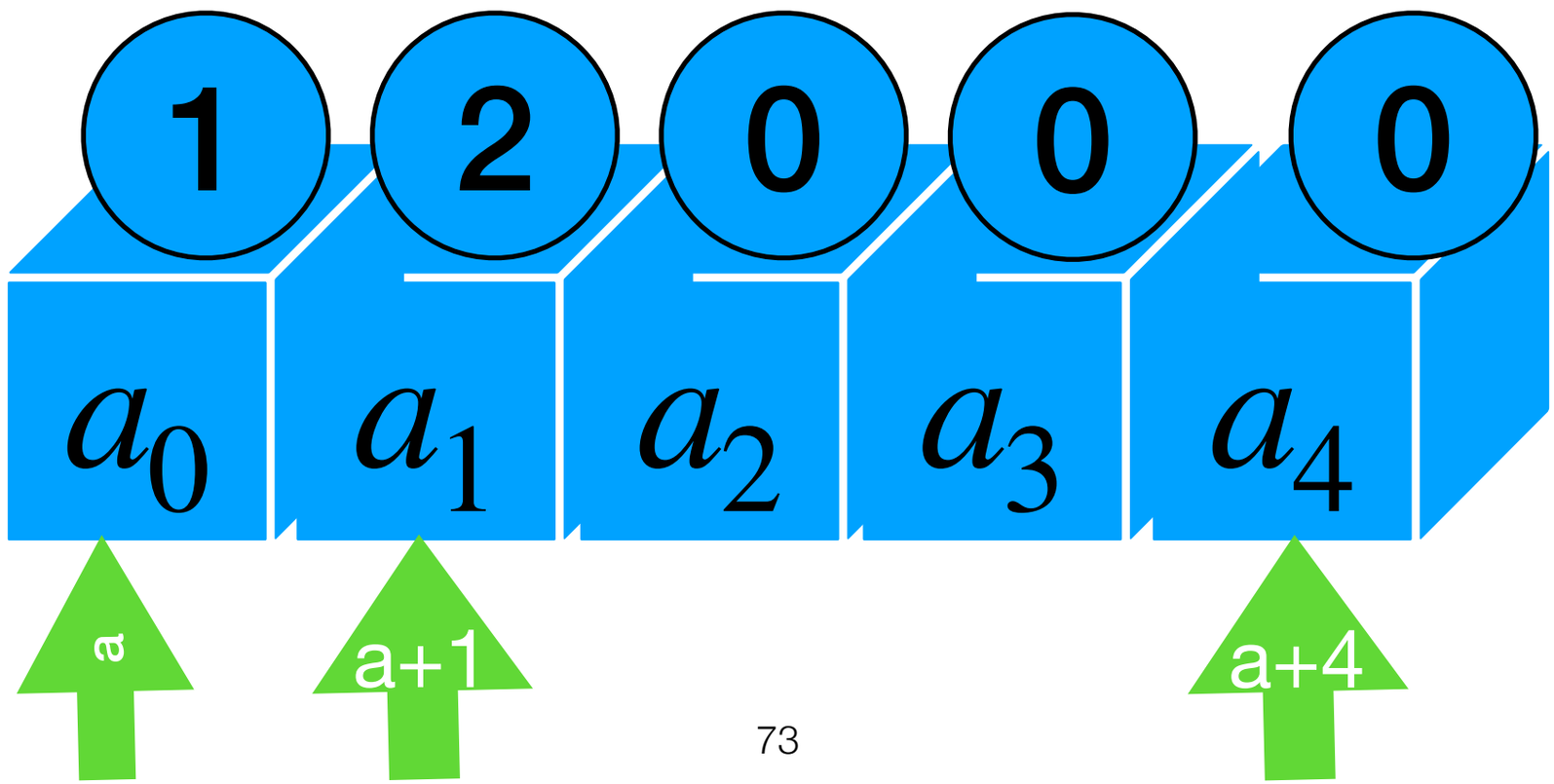
1． 变量的两种引用形式　指针

(3) 指针与数组:

- In expressions, the name of the array is a synonym for the address of its first element;
- but an array name is not a variable

表达式　同义词

数组a:



I. Starter Introduction

四. 一个鲁莽的行为

1． 变量的两种引用形式　指针

(4) 分配动态内存

```
int x; cin>>x;
int *a = new int[x];
```

```
delete [] a;
```

I. Starter Introduction

四. 一个鲁莽的行为

1． 变量的两种引用形式　指针

⑽ 若有定义 int x=10,*p1=&x; const int *p2=&x;　int *const p3=&x; 则下列选项错误的是　(　　)

A. p1=p2　　B. p1=p3　　C. p2=p1　　D. p2=p3

3. `int const * y;` //both mean the same
   `const int * y;`
   It means that y is a pointer that points to a constant integer value. If we initialize y while declaration as below:
   `int const * y = &a;`
   Then, it is possible to do: `y=&b;(true)` because y is a non-constant pointer that can point to anywhere.
   However, we cannot do: `*y=1;(wrong)` because the variable that y points to is a constant variable and cannot be modified.

I. Starter Introduction

四. 一个鲁莽的行为

1． 变量的两种引用形式　指针

9.[程序] 下列程序用指针数组数组对输入的 int 型数据按绝对值从大到小进行索引排序。

```
#include <iostream>
#include <cmath>
using namespace std;
_____ ;
int   main( )
{     int   *a,**id;     int   i,j,n;
      cin>>n;   a=new int[n];     id=_____ ;
      for(i=0;i<n;i++)   { cin>>a[i];   id[i]=&a[i];   }
      sort(id,n);     //进行索引排序
      for(i=0;i<n;i++)   cout<<_____<<"   ";   cout<<endl;   //按排序次序输出数据
      delete[ ]a;   delete[ ]id;     return 0;
}
void sort(int *p[ ],int n)   //对 p 数组指向的数据*p[0]、…、*p[n-1]按绝对值从大到小进行索引排序
{     for(int k,j,i=0;i<n;i++)
      {     for(k=i,j=i+1;j<n;j++) if( _____ ) k=j;
            if(k!=i) {   _____ ; p[k]=p[i]; p[i]=t;   }
      }
}
```

I. Starter Introduction

四. 一个鲁莽的行为

1． 变量的两种引用形式　指针

9.[程序] 下列程序用指针数组数组对输入的 int 型数据按绝对值从大到小进行索引排序。

抄过来!

```
1

int main( ) {
    int *a, **id;
    int i, j, n;
    cin >> n;
    a = new int[n];
    id = 2
    for (i = 0; i < n; i++) {
        cin >> a[i];
        id[i] = &a[i];
    }
    sort(id, n); //进行索引排序
    for (i = 0; i < n; i++) {
        cout 3 << " ";
    }
    cout << endl;
    delete[] a;
    delete[] id;
    return 0;
}
```

```
void sort(int *p[], int n) {
    for (int k, j, i = 0; i < n; i++) {
        for (k = i, j = i + 1; j < n; j++) {
            if ( 4 ) {
                k = j;
            }
        }
        if (k != i) {
            5
            p[k] = p[i];
            p[i] = t;
        }
    }
}
```

I. Starter Introduction

四. 一个鲁莽的行为

1． 变量的两种引用形式　指针

9.[程序] 下列程序用指针数组数组对输入的 int 型数据按绝对值从大到小进行索引排序。

```
5   void sort(int *p[], int n);
6
7   int main( ) {
8       int *a, **id;
9       int i, j, n;
10      cin >> n;
11      a = new int[n];
12      id = 2
13      for (i = 0; i < n; i++) {
14          cin >> a[i];
15          id[i] = &a[i];
16      }
17      sort(id, n); //进行索引排序
18      for (i = 0; i < n; i++) {
19          cout << 3 << " ";
20      }
21      cout << endl;
22      delete[] a;
23      delete[] id;
24      return 0;
25  }
```

id是一个二维数组...所以要new一些int*

```
27  void sort(int *p[], int n) {
28      for (int k, j, i = 0; i < n; i++) {
29          for (k = i, j = i + 1; j < n; j++) {
30              if ( 4 ) {
31                  k = j;
32              }
33          }
34          if (k != i) {
35              5
36              p[k] = p[i];
37              p[i] = t;
38          }
39      }
40  }
```

I. Starter Introduction

四. 一个鲁莽的行为

1． 变量的两种引用形式　指针

9.[程序] 下列程序用指针数组数组对输入的 int 型数据按绝对值从大到小进行索引排序。

```
5   void sort(int *p[], int n);
6
7   int main( ) {
8       int *a, **id;
9       int i, j, n;
10      cin >> n;
11      a = new int[n];
12      id = new int*[n];
13      for (i = 0; i < n; i++) {
14          cin >> a[i];
15          id[i] = &a[i];
16      }
17      sort(id, n); //进行索引排序
18      for (i = 0; i < n; i++) {
19          cout << 3 << " ";
20      }
21      cout << endl;
22      delete[] a;
23      delete[] id;
24      return 0;
25  }
```

按照绝对值排序

```
27  void so
28      for
29          fo
30              if (  4  ) {
31                  k = j;
32              }
33          }
34          if (k != i) {
35              5
36              p[k] = p[i];
37              p[i] = t;
38          }
39      }
40  }
```

I. Starter Introduction

四. 一个鲁莽的行为

1． 变量的两种引用形式　指针

9.[程序] 下列程序用指针数组数组对输入的 int 型数据按绝对值从大到小进行索引排序。

```
5   void sort(int *p[], int n);
6
7   int main( ) {
8       int *a, **id;
9       int i, j, n;
10      cin >> n;
11      a = new int[n];
12      id = new int*[n];
13      for (i = 0; i < n; i++) {
14          cin >> a[i];
15          id[i] = &a[i];
16      }
17      sort(id, n); //进行索引排序
18      for (i = 0; i < n; i++) {
19          cout << *id[i] << " ";
20      }
21      cout << endl;
22      delete[] a;
23      delete[] id;
24      return 0;
25  }
```

3

```
27  void sort(int *p[], int n) {
28      for (int k, j, i = 0; i < n; i++) {
29          for (k = i, j = i + 1; j < n; j++) {
30              if (abs(*p[j]) > abs(*p[k])) {
31                  k = j;
32              }
33          }
34          if (k != i) {
35              int t = *p[k];
36              p[k] = p[i];
37              p[i] = t;
38          }
39      }
40  }
```

这里是交换元素来了

5

I. Starter Introduction

四. 一个鲁莽的行为

1． 变量的两种引用形式　指针

9.[程序] 下列程序用指针数组数组对输入的 int 型数据按绝对值从大到小进行索引排序。

```
5   void sort(int *p[], int n);
6
7   int main( ) {
8       int *a, **id;
9       int i, j, n;
10      cin >> n;
11      a = new int[n];
12      id = new int*[n];
13      for (i = 0; i < n; i++) {
14          cin >> a[i];
15          id[i] = &a[i];
16      }
17      sort(id, n);//进行索引排序
18      for (i = 0; i < n; i++) {
19          cout << *id[i] << " ";
20      }
21      cout << endl;
22      delete[] a;
23      delete[] id;
24      return 0;
25  }
```

输出元素

3

```
27  void sort(int *p[], int n) {
28      for (int k, j, i = 0; i < n; i++) {
29          for (k = i, j = i + 1; j < n; j++) {
30              if (abs(*p[j]) > abs(*p[k])) {
31                  k = j;
32              }
33          }
34          if (k != i) {
35              int* t = p[k];
36              p[k] = p[i];
37              p[i] = t;
38          }
39      }
40  }
```

I. Starter Introduction

四. 一个鲁莽的行为

1． 变量的两种引用形式　指针

9.[程序] 下列程序用指针数组数组对输入的 int 型数据按绝对值从大到小进行索引排序。

```
5   void sort(int *p[], int n);
6
7   int main( ) {
8       int *a, **id;
9       int i, j, n;
10      cin >> n;
11      a = new int[n];
12      id = new int*[n];
13      for (i = 0; i < n; i++) {
14          cin >> a[i];
15          id[i] = &a[i];
16      }
17      sort(id, n); //进行索引排序
18      for (i = 0; i < n; i++) {
19          cout << *id[i] << " ";
20      }
21      cout << endl;
22      delete[] a;
23      delete[] id;
24      return 0;
25  }
```

```
27  void sort(int *p[], int n) {
28      for (int k, j, i = 0; i < n; i++) {
29          for (k = i, j = i + 1; j < n; j++) {
30              if (abs(*p[j]) > abs(*p[k])) {
31                  k = j;
32              }
33          }
34          if (k != i) {
35              int* t = p[k];
36              p[k] = p[i];
37              p[i] = t;
38          }
39      }
40  }
```

查看状态机的执行

I. Starter Introduction

四. 一个鲁莽的行为

1． 变量的两种引用形式　指针

(4) 特别说说C类型的的字符串

> char s[6] = “□□□□□□”

也是一个特殊的数组, 之后一个位置必须是\0, 表示结束

一些比较方便常用工具:

- strlen(str) – 字符串的长度
- strcpy(s2, s1) – 把s2复制到s1中
- strcmp(s1, s2) – 比较字符串
- strcat(s1, s2) – 拼接起来

I. Starter Introduction

四. 一个鲁莽的行为

1． 变量的两种引用形式　指针

(4) 特别说说C类型的的字符串

> char s[6] = “□□□□□□”

也是一个特殊的数组, 之后一个位置必须是\0, 表示结束

一些比较方便常用工具:

- strlen(str) – 字符串的长度
- strcpy(s2, s1) – 把s2复制到s1中
- strcmp(s1, s2) – 比较字符串
- strcat(s1, s2) – 拼接起来

I. Starter Introduction

四. 一个鲁莽的行为

1． 变量的两种引用形式　指针

(4) 特别说说C类型的的字符串

8.[程序] 下列程序对一个无符号十进制数串进行加 1 和减 1 运算，输出运算后的数串。

```
#include <iostream>
#include <cstring>
using namespace std;
void increase(char *s)   //数串加 1
{    char *p,*q;    for(p=s;*p;p++) ;    q=p--;
     while(p>=s&&*p== '9')  {    *p= '0';    _____ ;  }  //处理进位
     if(p<s)    // 第一位数字有进位
     {    *(q+1)= '\0';    for(q--;q>=s;q--) *(q+1)=*q;    *s= '1';    }
     else   _____ ;
}
void decrease(char *s)   //数串减 1
{    char *p,*q;    for(p=s;*p;p++) ;    q=p--;
     while(p>=s&&*p== '0')  {   *p= '9';   p--;   }   //处理借位
     if(_____)   {   strcpy(s, "-1");   }    // "0"减去 1
     else   {    _____;    if(*s== '0'&&q>s+1) strcpy(s,s+1);    }
}
int   main( )
{    char s[80] = "32341251489999239124993999",s2[80]= "1021500000";
     cout<<s<<"\t"<<s2<<endl;   increase(s);   decrease(s2); cout<<s<<"\t" <<s2<<endl;   return 0;
}
```

L Standardization

```
4   void increase(char *s){ char *p,*q; for(p=s;*p;p++) ; q=p--;
5       while(p>=s&&*p== '9'){
6           *p= '0';
7           1
8       }
9       if(p<s){
10          *(q+1)= '\0';
11          for(q--;q>=s;q--) *(q+1)=*q;
12          *s= '1';
13      }else{
14          2
15      }
16
17  }
18  void decrease(char *s){
19      char *p,*q; for(p=s;*p;p++) ; q=p--;
20      while(p>=s&&*p== '0'){
21          *p= '9'; p--;
22      }
23      if(p  3  ){ strcpy(s, "-1"); }
24      else {  4  strcpy(s,s+1);
25  }
26  int main( ){
27      char s[80] = "32341251489999239124993999",s2[80]= "1021500000";
28      cout<<s<<"\t"<<s2<<endl;
29      increase(s); decrease(s2); cout<<s<<"\t" <<s2<<endl;
30      return 0;
31      }
```

算，输出运算后的数串。

L Starter Introduction


```
4   void increase(char *s) { char *p,*q; for(p=s;*p;p++) ; q=p--;
5       while(p>=s&&*p== '9') {
6           *p= '0';
7
8       }
9       if(p<s){
10          *(q+1)= '\0';
11          for(q--;q>=s;q--) *(q+1)=*q;
12          *s= '1';
13      }else{
14
15      }
16
17  }
18  void decrease(char *s){
19      char *p,*q; for(p=s;*p;p++) ; q=p--;
20      while(p>=s&&*p== '0'){
21          *p= '9'; p--;
22      }
23      if(p                    ){ strcpy(s, "-1"); }
24      else {                      strcpy(s,s+1);
25  }
26  int main( ){
27      char s[80] = "32341251489999239124993999",s2[80]= "1021500000";
28      cout<<s<<"\t"<<s2<<endl;
29      increase(s); decrease(s2); cout<<s<<"\t" <<s2<<endl;
30      return 0;
31      }
```

1

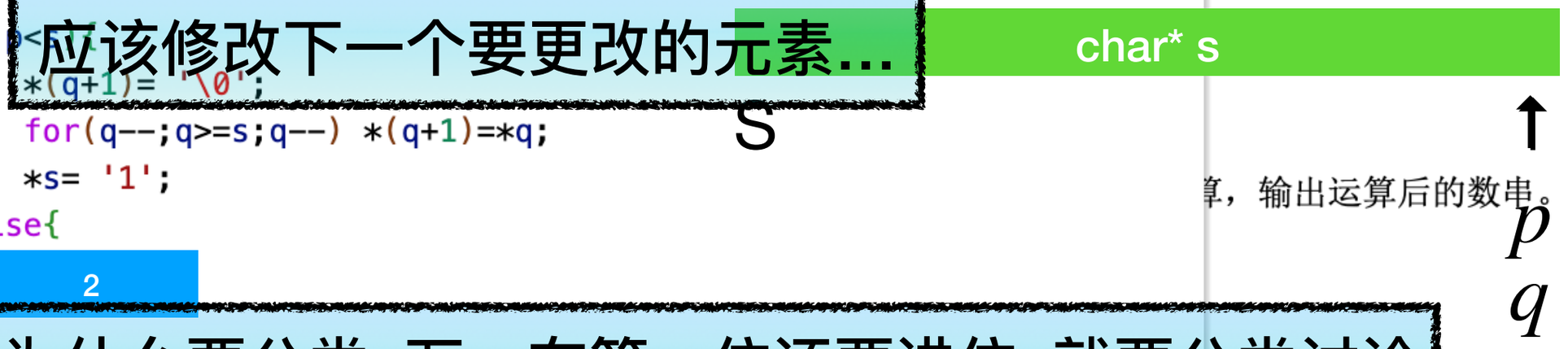
应该修改下一个要更改的元素...

2

为什么要分类: 万一在第一位还要进位, 就要分类讨论

3

4



算，输出运算后的数串。

I. Starter Introduction

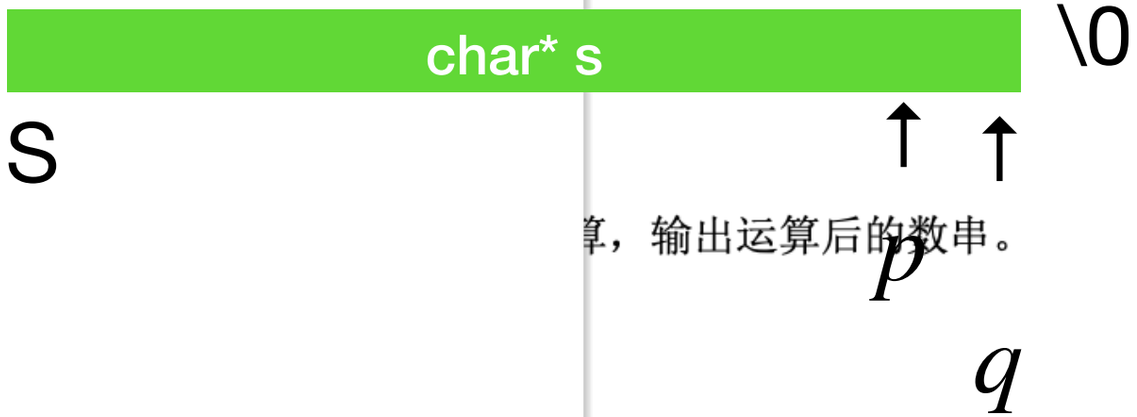

```
4   void increase(char *s) { char *p,*q; for(p=s;*p;p++) ; q=p--;
5       while(p>=s&&*p== '9') {
6           *p= '0';
7           p--;
8       }
9       if(p<s){
10          *(q+1)= '\0';
11          for(q--;q>=s;q--) *(q+1)=*q;
12          *s= '1';
13      }else{
14          2
15      }
16
17  }
18  void decrease(char *s){
19      char *p,*q; for(p=s;*p;p++) ; q=p--;
20      while(p>=s&&*p== '0'){
21          *p= '9'; p--;
22      }
23      if(p      3      ){ strcpy(s, "-1"); }
24      else { ;      4      strcpy(s,s+1);
25  }
26  int main( ){
27      char s[80] = "32341251489999239124993999",s2[80]= "1021500000";
28      cout<<s<<"\t"<<s2<<endl;
29      increase(s); decrease(s2); cout<<s<<"\t" <<s2<<endl;
30      return 0;
31      }
```

否则就是不在第一位的情形

算，输出运算后的数串。

I. Starter Introduction


```
4   void increase(char *s) { char *p,*q; for(p=s;*p;p++) ; q=p--;
5       while(p>=s&&*p== '9') {
6           *p= '0';
7           p--;
8       }
9       if(p<s){
10          *(q+1)= '\0';
11          for(q--;q>=s;q--) *(q+1)=*q;
12          *s= '1';
13      }else{
14          *p=*p+1;
15      }
16
17  }
18  void decrease(char *s){
19      char *p,*q; for(p=s;*p;p++) ; q=p--;
20      while(p>=s&&*p== '0'){
21          *p= '9';p--;
22      }
23      if(p                    ){ strcpy(s, "-1"); }
24      else                      strcpy(s,s+1);
25  }
26  int main( ){
27      char s[80] = "32341251489999239124993999",s2[80]= "1021500000";
28      cout<<s<<"\t"<<s2<<endl;
29      increase(s); decrease(s2); cout<<s<<"\t" <<s2<<endl;
30      return 0;
31      }
```



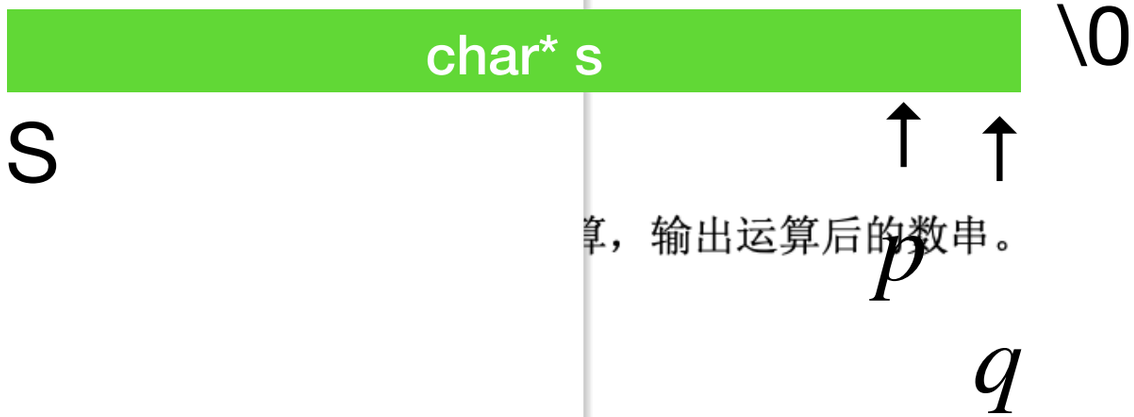
算，输出运算后的数串。

借位也是一样的，首先确定字符串的长度区间

借位的条件

I. Starter Introduction


```
4   void increase(char *s) { char *p,*q; for(p=s;*p;p++) ; q=p--;
5       while(p>=s&&*p== '9') {
6           *p= '0';
7           p--;
8       }
9       if(p<s){
10          *(q+1)= '\0';
11          for(q--;q>=s;q--) *(q+1)=*q;
12          *s= '1';
13      }else{
14          *p=*p+1;
15      }
16
17  }
18  void decrease(char *s){
19      char *p,*q; for(p=s;*p;p++) ; q=p--;
20      while(p>=s&&*p== '0'){
21          *p= '9'; p--;
22      }
23      if(p == s && *p == '0'){ strcpy(s, "-1"); }
24      else { strcpy(s,s+1);
25  }
26  int main( ){
27      char s[80] = "32341251489999239124993999",s2[80]= "1021500000";
28      cout<<s<<"\t"<<s2<<endl;
29      increase(s); decrease(s2); cout<<s<<"\t" <<s2<<endl;
30      return 0;
31      }
```



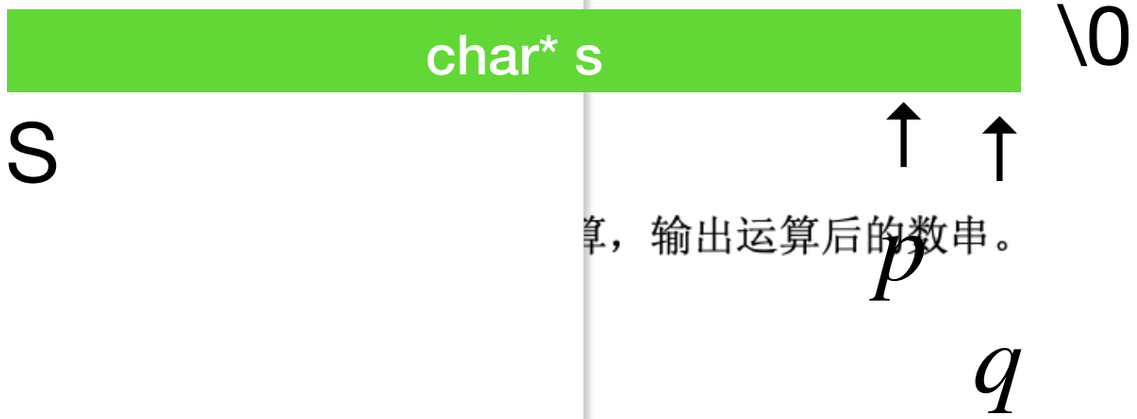
算，输出运算后的数串。

4

和上面是对偶的

I. Starter Introduction


```
4   void increase(char *s) { char *p,*q; for(p=s;*p;p++) ; q=p--;
5       while(p>=s&&*p== '9') {
6           *p= '0';
7           p--;
8       }
9       if(p<s){
10          *(q+1)= '\0';
11          for(q--;q>=s;q--) *(q+1)=*q;
12          *s= '1';
13      }else{
14          *p=*p+1;
15      }
16
17  }
18  void decrease(char *s){
19      char *p,*q; for(p=s;*p;p++) ; q=p--;
20      while(p>=s&&*p== '0'){
21          *p= '9'; p--;
22      }
23      if(p == s && *p == '0'){ strcpy(s, "-1"); }
24      else { *p = *p - 1; }if(*s== '0'&&q>s+1) strcpy(s,s+1);
25  }
26  int main( ){
27      char s[80] = "32341251489999239124993999",s2[80]= "1021500000";
28      cout<<s<<"\t"<<s2<<endl;
29      increase(s); decrease(s2); cout<<s<<"\t" <<s2<<endl;
30      return 0;
31      }
```



算，输出运算后的数串。

I. Starter Introduction

三. 加强思维的管理

1. 函数: 整合相似过程

(2) 递归(recursion): 递归的定义 — 参见递归

```
1  #include <iostream>
2  #include <cmath>
3  using namespace std;
4
5  int fib(int n){
6    if(n == 1 || n==2) return 1;
7    else return fib(n-1)+fib(n-2);
8  }
9
10 int main( ) {
11     fib(5);
12     return 0;
13 }
```

查看状态机的执行

I. Starter Introduction

三. 加强思维的管理

1. 函数: 整合相似过程

(2) 递归(recursion): 递归的定义 — 参见递归

例子: 用展开的方法求行列式

$$D_i = \sum_{i=1}^{n} (-1)^{1+i} a_{1i} M_{1i}$$

http://paulbourke.net/miscellaneous/determinant/determinant.c

I. Starter Introduction

三. 加

1. 函

(2

例子4. 刚刚奇怪的东西(contd)

像极了这样的场景

- 你在晚自习上看课外书(执行原来的函数)
- 老师来了, 让你写作业(函数调用)
- 你把作业叠放在课外书上, 开始做作业 (执行函数)
- ~~(象征性的)~~ 做完作业之后你把作业扔了继续看课外书(回到原来的函数)

程序是不会犯错的, 所以它不会象征性的做函数调用.

> 程序就是状态机, 我们可以观测状态机的设计, 实现, 执行.

I. Starter Introduction

三改错题(10 分)（直接在程序上改正）

[要求] 改错时，在错误下划线，然后在旁边写上正确的内容（删除则写上删除字样）。只能修改语句中的一部分内容，不能增加语句，也不能删去语句。

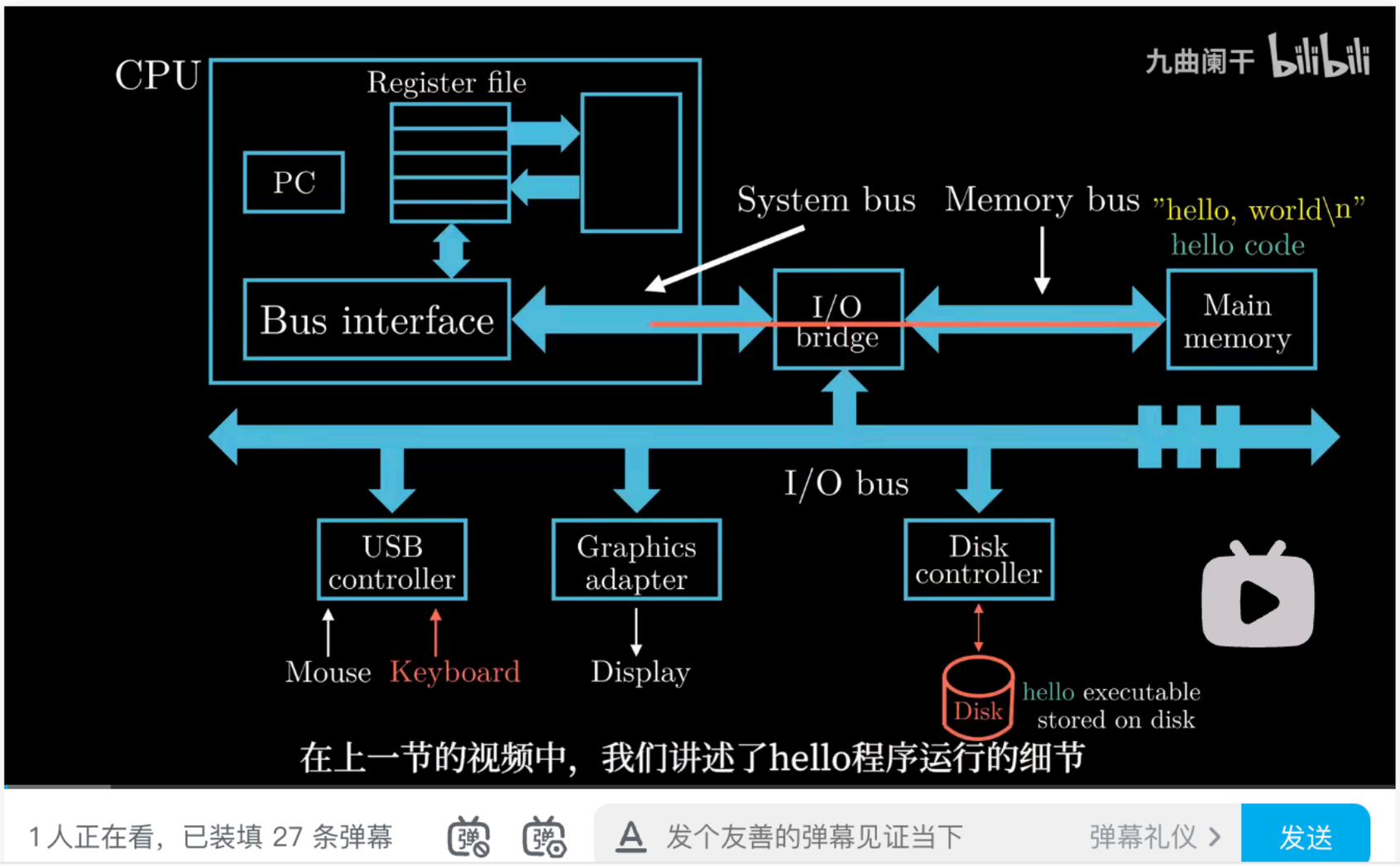
[程序] 输入整数 a,b,n，输出卢卡斯序列中的第 n+1 项，即 $U_n(a,b),V_n(a,b)$。

卢卡斯序列为：$\begin{cases} U_k(a,b)=aU_{k-1}(a,b)-bU_{k-2}(a,b), \\ V_k(a,b)=aV_{k-1}(a,b)-bV_{k-2}(a,b), \end{cases} k=2,3,\cdots$，初始值：$\begin{cases} U_0(a,b)=0, & U_1(a,b)=1, \\ V_0(a,b)=2, & V_1(a,b)=a, \end{cases}$

```
#include <iostream>
using namespace std;
int U(int k,int a,int b)
{     if(k<=1)    return k;    U(k,a,b)= a*U(k-1,a,b)-b*U(k-2,a,b);    }
int V(int k,int a,int b)
{     if(!k)    return 2;
      if(k=1)    return a;
      int V0=2,V1=a,V2,i;
      For(i=2;i<=k;i++)    { V2=a*V1-b*V0;    V0=V2;    V1=V2;    }
      return V2;
}
int    main( )
{     int a,b,n;
      cin>>a,b,n;
      cout<< "U="<<U(n,a,b)<< "\tV="<<V(n,a,b)<<endl;
      return 0;
}
```

II. 计算机系统基础简介

【CSAPP_深入理解计算机系统】1-3.计算机系统漫游



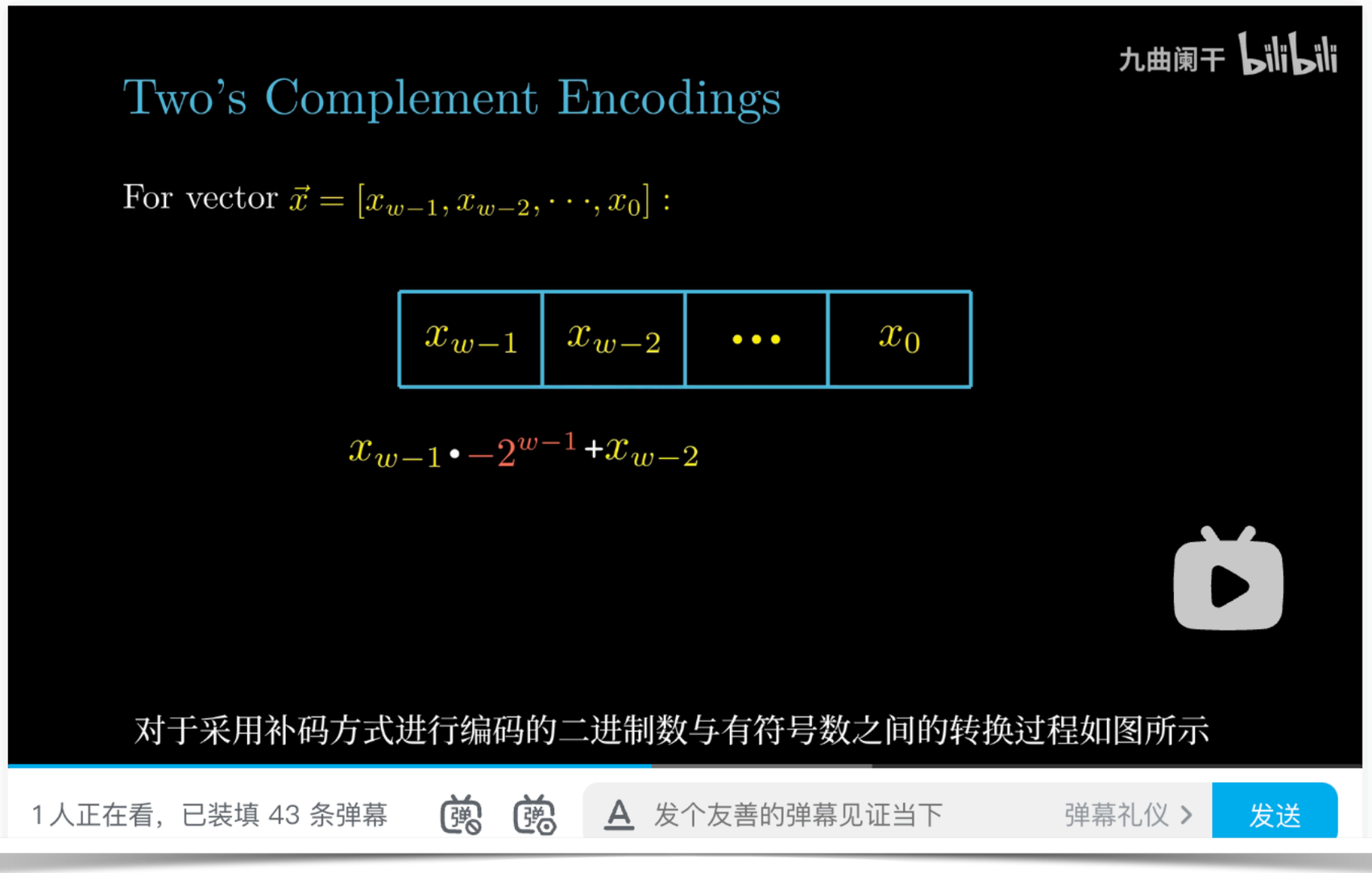
寄存器 计算机系统架构

https://www.bilibili.com/video/BV1mi4y137g8/

II. 计算机系统基础简介

【CSAPP-深入理解计算机系统】2-2.整数的表示(上)

2.3万 43 2020-08-30 13:49:53 未经作者授权，禁止转载



反码 补码

https://www.bilibili.com/video/BV1HK411K7TX

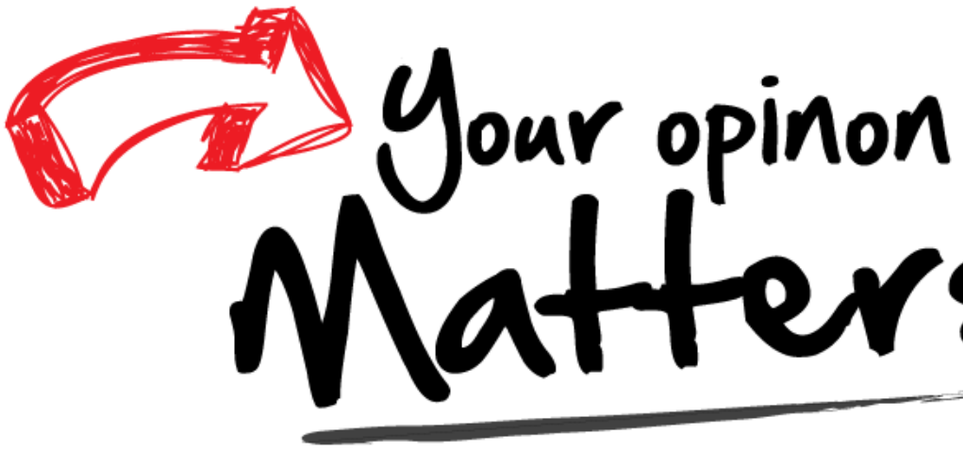
参考资料

1. BRIAN W. KERNIGHAN DENNIS; M. RITCHIE *The C Programming language*
2. 魏恒峰. *C*编程语言*2022* https://www.bilibili.com/video/BV1H84y117LN
3. 南京大学计算机系. 问题求解 http://cslabcms.nju.edu.cn/problem_solving/index.php/2022%E7%BA%A7
4. Randal Bryant. *Computer Systems: A Programmer's Perspective*
5. 九曲阑干. 计算机系统基础*(CSAPP)* https://www.bilibili.com/video/BV1cD4y1D7uR
6. 李樾等. 程序分析 https://zhuanlan.zhihu.com/p/417187798

参考资料

7. 蒋炎岩. 代码风格和调试指南 https://www.bilibili.com/video/BV1aT411H77C/
8. Pythontutor代码可视化工具 https://pythontutor.com/cpp.html

Thank You!



QQ: 2095728218
Email: micoael@qq.com
(学校) gwzhang@cug.edu.cn