

太戈编程
etiger.vip

信奥算法

数组的另类表示

指针和数组

请预测
输出结果

```
1 #include<iostream>
2 #include<string>
3 using namespace std;
4 string x[3]={"ha","wa","la"};
5 int main(){
6     cout<<x<<endl;
7     cout<<x+1<<endl;
8     cout<<x+2<<endl;
9     cout<<*x<<endl;
10    cout<<*(x+1)<<endl;
11    cout<<*(x+2)<<endl;
12    cout<<(x+1)[0]<<endl;
13    cout<<(x+1)[1]<<endl;
14    return 0;
15 }
```

存储
内存
地址

数组名x
也是指针

x+1
也是指针

x+2
也是指针

请口头翻译每一行

```
1  #include<iostream>
2  using namespace std;
3  int x[4]={5,6,7,8};
4  int main(){
5      int *p;
6      p=x+1;
7      cout<<p[0]<<endl;
8      cout<<p[1]<<endl;
9      cout<<p[2]<<endl;
10     return 0;
11 }
```

指针p也是数组名

将x数组1号元素的地址赋值给p

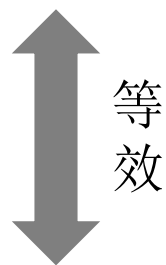
6

7

8

太戈编程
www.etiger.vip

指针变量名



数组名

都记录内存地址

大数模型 高精度计算

数据类型： 范围

类型	范围	占用字节
char	-128 ... 127	1
short	-32768 ... 32767	2
int	-2147483648 ... 2147483647	4
long long	-9223372036854775808 ... 9223372036854775807	8
unsigned short	0 ... 65535	2
unsigned int	0 ... 4294967295	4
unsigned long long	0 ... 18446744073709551615	8

x+y 问题

输入正整数 x 和 y ，输出 $x+y$

注意: $x, y \leq 10^{200}$ ← 200位不够

输入样例

123456789012345678901234567890

999999999999999999999999999999999999

输出样例

1123456789012345678901234567889

如何输入大数？

用字符串

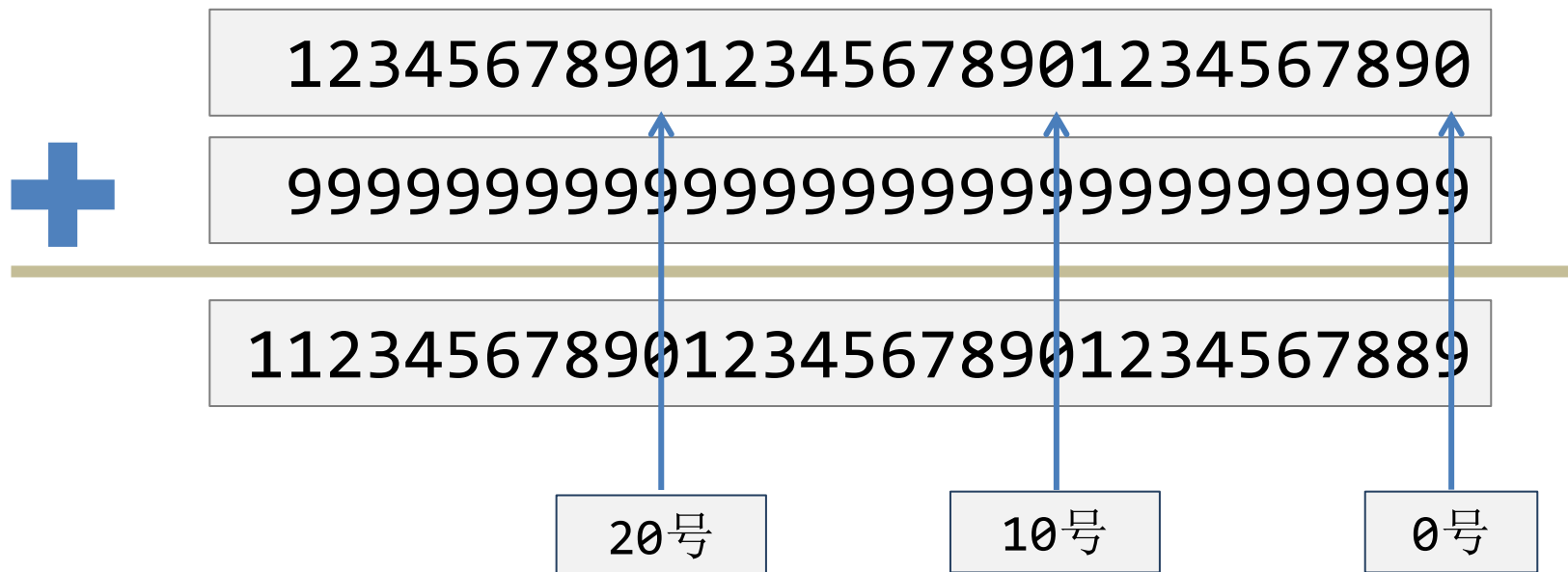
如何存储大数?

用数组

大数存储：用数组

```
#define SIZE 209
int x[SIZE],y[SIZE];
```

大数x,y有209位高精度



记 笔 记

大数的个位数 放数组0号位

x+y问题： 算法步骤

```
23 int x[SIZE],y[SIZE];
24 int main() {
25     string s1,s2;
26     cin>>s1>>s2;
27     converts(x,s1);
28     converts(y,s2);
29     add(x,y);
30     print(x);
31     return 0;
32 }
```

定义大数x,y高精度数组

输入字符串s1,s2

将字符串s1,s2
转换成大数x,y
高精度数组

算加法: $x+=y$

输出x

请同学将中文
算法步骤记笔记

高精度大数的输入

1

先输入字符串变量

2

再把字符串变量
转成高精度数组

数组0号元素是个位
能对齐

字符串转成高精度大数

`int *a` 表明`a`是数组

```
5 void converts(int *a, string s){  
6     int i, len=s.size();  
7     for(i=0; i<len; i++) a[i]=s[len-i-1]-'0';  
8     for(; i<SIZE; i++) a[i]=0;  
9 }
```

个位数 放在数组的0号位置

十位数 放在数组的1号位置

。 。 。 。 。 。

空的位置全部用0填充

字符串转成高精度大数

```
5 void converts(int *a, string s){  
6     int i, len=s.size();  
7     for(i=0; i<len; i++) a[i]=s[len-i-1]-'0';  
8     for(; i<SIZE; i++) a[i]=0;  
9 }
```

易
错
点
汇
总

i的定义在**for**循环外
for循环里不可以再定义**i**

len-i-1不能忘了减**1**

字符转整数不能忘了减'**0**'

剩余位置不要忘了填充**0**

高精度大数输出

```
18 void print(int *a){  
19     int i;  
20     for(i=SIZE-1;i>0;i--) if(a[i]>0) break;  
21     for(;i>=0;i--) cout<<a[i]; cout<<endl;  
22 }
```

先找到数组中非0位置的最高位，或个位

再逐位输出
直到数组的0号下标位置
也就是个位数

高精度大数输出

```
18 void print(int *a){  
19     int i;  
20     for(i=SIZE-1;i>0;i--) if(a[i]>0) break;  
21     for(;i>=0;i--) cout<<a[i]; cout<<endl;  
22 }
```

易
错
点
汇
总

i的定义在for循环外
for循环里不可以再定义i

循环编号从大到小,而不是从小到大

不能遗忘换行的输出

2种高精度加法

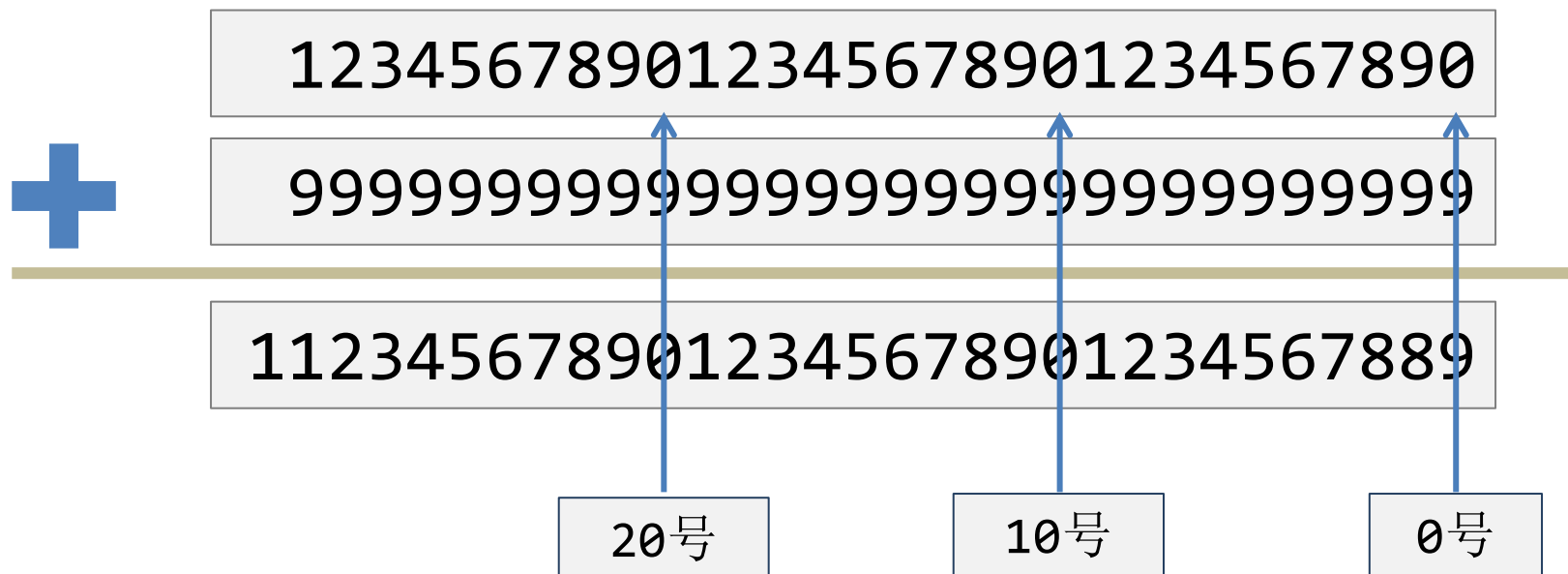
高精度+高精度

```
void add(int *a, int *b, int len=SIZE)
```

高精度+int

```
void add(int *a, int b)
```


加法步骤



高精度+高精度

```
10 void add(int *a,int *b,int len=SIZE){//a+=b
11     //逐位相加
12     for(int i=0;i<len;i++)
13         if((a[i]+=b[i])>=10){//进位
14             a[i+1]++;
15             a[i]-=10;
16         }
17 }
```

从个位开始，逐位相加

如果当前位置的数 ≥ 10 ，就进位

高精度+int

```
10 void add(int *a, int b){ //a+=b
11     //逐位相加, 循环条件: b有剩余 或者 a[i]结果需进位
12     for(int i=0; b || a[i]>=10; b/=10, i++)
13     {
14         if((a[i]+=b%10)>=10) { //进位
15             a[i+1]++;
16             a[i]-=10;
17         }
18     }
```

请理解循环条件的含义

注意循环里b和i每次变化

模块化编程

把完整的程序划分成一块一块的模块

每个模块功能相对独立

每个模块内部代码不根据题目变化

现场挑战

太戈编程1002

$x+y+z$ 问题： 算法步骤

定义大数 x, y, z 高精度数组

设置数组多大?

输入字符串 sx, sy, sz

将字符串 sx 转换成大数 x 数组
将字符串 sy 转换成大数 y 数组
将字符串 sz 转换成大数 z 数组

算加法: $x+=y$

算加法: $x+=z$

输出 x

请同学将中文
算法步骤记笔记
给老师查看

```
2 #define SIZE 509
```

```
19 int x[SIZE],y[SIZE],z[SIZE];
```

```
20 int main(){
```

```
21     freopen("plus.in","r",stdin);
```

```
22     freopen("plus.out","w",stdout);
```

```
23     string sx,sy,sz;
```

```
24     cin>>sx>>sy>>sz;
```

```
25     converts(x,sx);
```

```
26     converts(y,sy);
```

```
27     converts(z,sz);
```

```
28     add(x,y);
```

```
29 

|  |
|--|
|  |
|--|


```

```
30 

|  |
|--|
|  |
|--|


```

```
31     return 0;
```

```
32 }
```

太戈编程

1001

1002

拓展题

353