



§ 6. 指针基础 – 画内存图并分析

要求:

- 1、模仿第06模块PDF课件中 (P. 15-18/P. 26-29) 的样式，画出下列每小题每一步执行的内存分配及指向图示，分析为什么得到最后的结果。
 - ★ PDF课件的P. 30（如何同时得到周长和面积）
 - ★ PDF课件的P. 31（为什么无法进行交换）
 - ★ PDF课件的P. 32（为什么会出现错误，导致错误的关键词句是哪一句）
- 2、每个语句要画一张内存状态图，每小题都是4张图
 - ★ 第1张初始内存分配图附件已给出
- 3、不允许手写、手写后贴图
- 4、转换为pdf后在“实验报告”中提交（11.25前）

§ 6. 指针基础 – 画内存图并分析



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

```
#define PI 3.14159
```

```
double SL(double R, double *L)
```

```
{ double S;
  S = PI*R*R;
  *L = 2*PI*R;
  return S;
}
```

```
int main()
```

```
{ double s, l, r=3;
  s=SL(r, &l);
  cout << "s=" << s << endl;
  cout << "l=" << l << endl;
}
```

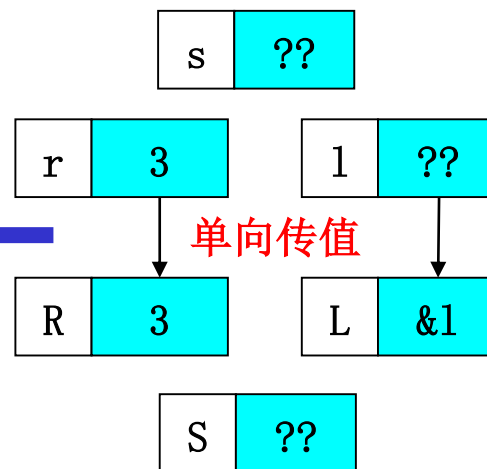
函数执行后同时得到周长及面积

周长：指针变量做形参方式

面积：函数返回值方式

注：函数的return只能带一个返回值!!

main



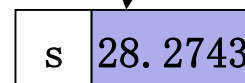
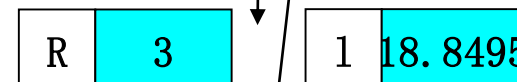
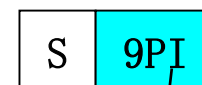
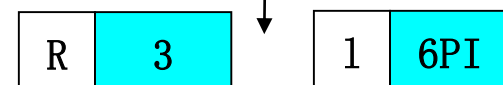
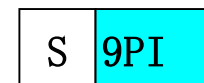
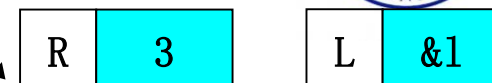
单向传值

SL

初始内存分配如图所示
请自行画出SL中三句话
执行时内存的变化
理解最后的输出结果

$*L = 2*PI*R$

$S = PI*R*R$



函数传值给SL后 $S=PI*R*R$
使得变量S的值为9PI，之后，
 $*L$ 中L中为&l即 $*L$ 指向l，
 $*L$ 指向l被赋值为6PI，返回
值S赋值给s，s，l输出

$s=28.2743$

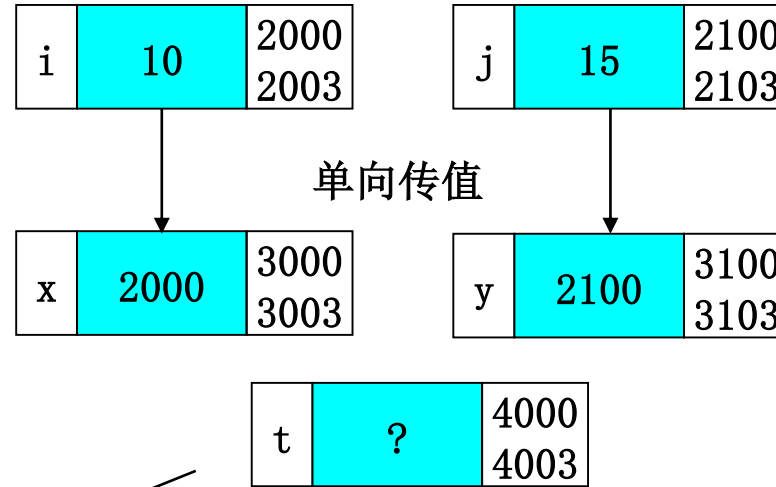
$l=18.8495$



§ 6. 指针基础 – 画内存图并分析

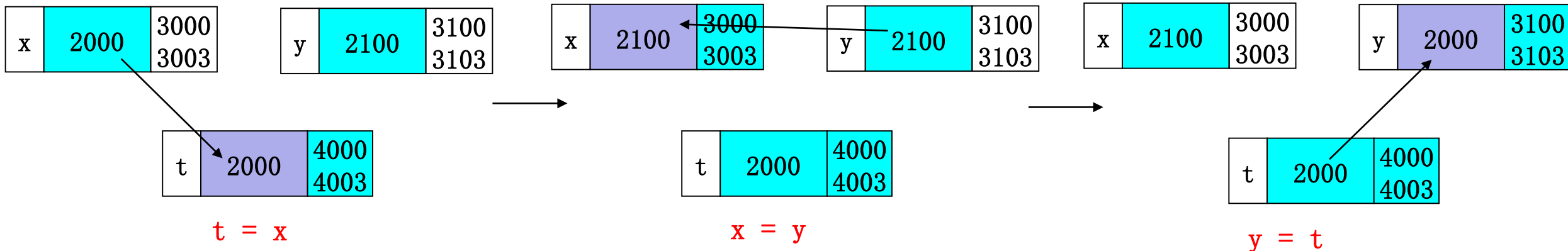
```
void swap(int *x, int *y)
{
    int *t;
    t = x;
    x = y;
    y = t;
}
```

```
int main()
{
    int i=10, j=15;
    cout << "i=" << i << " j=" << j << endl;
    swap(&i, &j);
    cout << "i=" << i << " j=" << j << endl;
}
```



i的地址被储存在x里，j的地址被储存在y里，之后，x赋给t，y赋给x，t赋，x与y存储的值互换，又因为函数单向传值，所以对i，j没有影响

初始内存分配如图所示
请自行画出swap中三句话
执行时内存的变化
理解为什么无法交换





§ 6. 指针基础 – 画内存图并分析

```
void swap(int *x, int *y)
```

```
{
    int *t;
    *t = *x;
    *x = *y;
    *y = *t;
}
```

VS2019编译报错

-使用了未初始化的局部变量t

其它编译器可能可以运行

初始内存分配如图所示，请自行画出swap中三句话执行时内存的变化，理解为什么出现严重错误

另1：哪句是错误的键？

另2：int *t 改为 int tt, *t;

t = &tt;

为什么就正确了？

```
int main()
```

```
{
    int i=10, j=15;
    cout << "i=" << i << " j=" << j << endl;
    swap(&i, &j);
    cout << "i=" << i << " j=" << j << endl;
}
```

i	10	2000
		2003

j	15	2100
		2103

单向传值

x	2000	3000
		3003

y	2100	3100
		3103

t	(假设5000)	4000
		4003

?		5000
		5003

i=10 j=15

提示：5000-5003系统是否分配给了程序？

i=15 j=10

或 死机或其它非正常现象

另1：*t=*x是关键性的错误。

另2：int tt会在程序内开辟一部分内存空间给tt，所以此时t中存储的地址即tt的地址一定在程序里面。

错误原因：在执行过程中，i中的值要赋值给t存储的地址中的值，但t存储的地址不一定分配给了程序，因此在程序里可能找不到要赋值的对象，因此报错

