

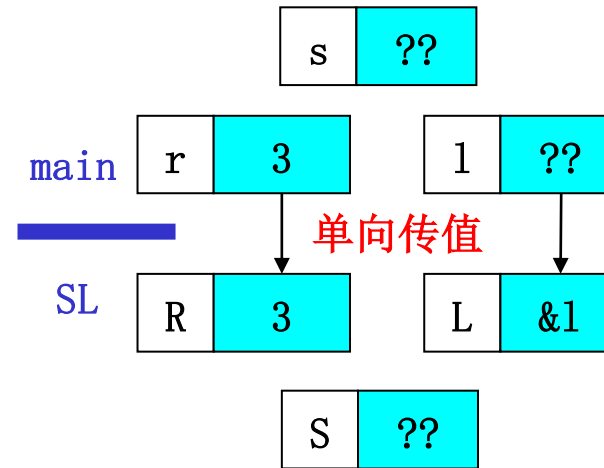


§ 12. 指针进阶 – 基础知识题

```
#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!!



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



§ 12. 指针进阶 – 基础知识题

```
#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; s=28.2743
```

```
  cout << "l=" << l << endl; l=18.8495
```

```
}
```

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

r	3
---	---

l	??
---	----

SL

R	3
---	---

L	&l
---	----

s	??
---	----

S	28.27431
---	----------

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



§ 12. 指针进阶 – 基础知识题

```
#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; s=28.2743
```

```
  cout << "l=" << l << endl; l=18.8495
```

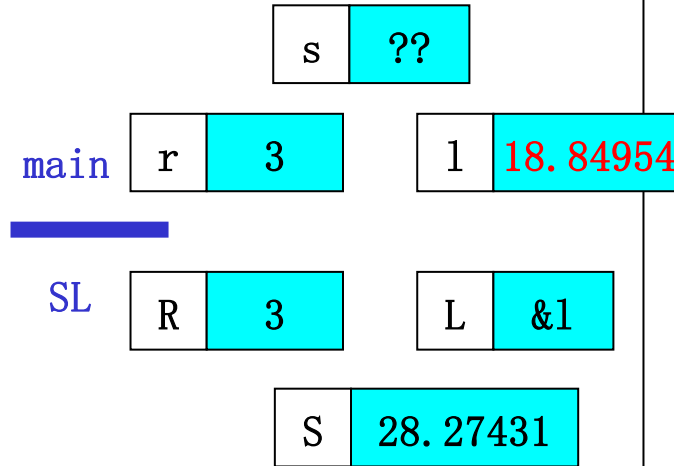
```
}
```

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!!



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



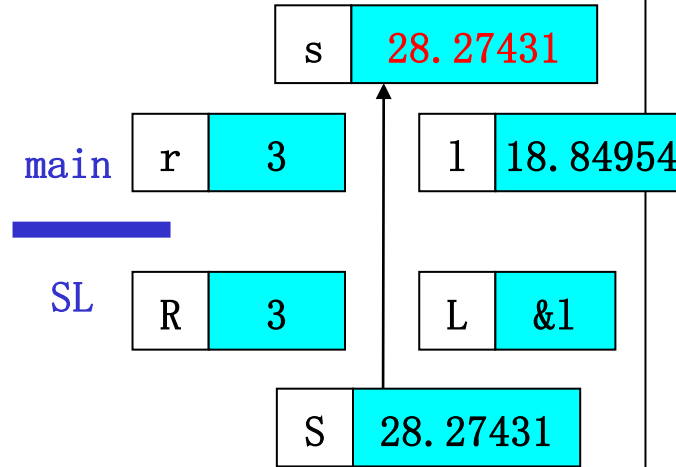
§ 12. 指针进阶 – 基础知识题

```
#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！！



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



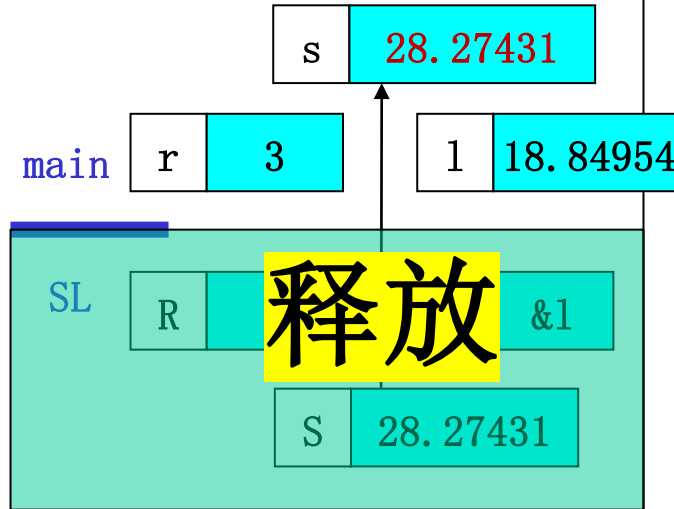
§ 12. 指针进阶 – 基础知识题

```
#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!!



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

分析：通过double R单向传递半径信息，通过指针 *L 修改main中的 l 以得到周长信息，通过函数返回值来得到面积信息。

Microsoft Visual Studio 调试控制台

```
s=28.2743
l=18.8495
```

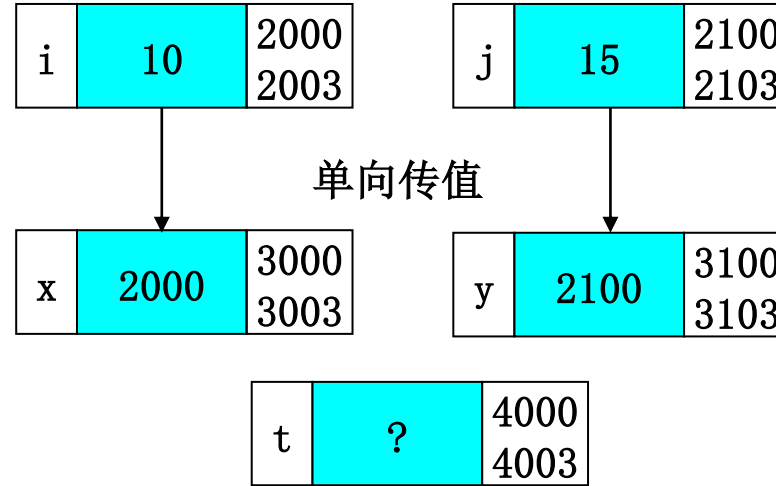




§ 12. 指针进阶 – 基础知识题

```
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;
}
```



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



§ 12. 指针进阶 – 基础知识题

```
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	10	2000
		2003

j	15	2100
		2103

x	2000	3000
		3003

y	2100	3100
		3103

t	2000	4000
		4003

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



§ 12. 指针进阶 – 基础知识题

```
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	10	2000
		2003

j	15	2100
		2103

x	2100	3000
		3003

y	2100	3100
		3103

t	2000	4000
		4003

i=10 j=15

i=10 j=15

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



§ 12. 指针进阶 – 基础知识题

```
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	10	2000
		2003

j	15	2100
		2103

x	2100	3000
		3003

y	2000	3100
		3103

t	2000	4000
		4003

i=10 j=15

i=10 j=15

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



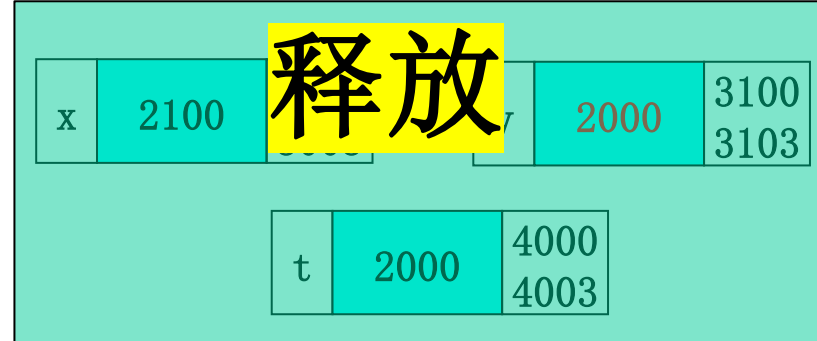
§ 12. 指针进阶 – 基础知识题

```
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;
}
```

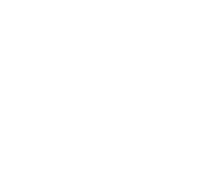
i	10	2000
		2003

j	15	2100
		2103



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

分析：因为 main 中 i,j 的值实际上并没有改变，改变的只是函数中自动指针向量。





§ 12. 指针进阶 – 基础知识题

```
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: 哪句是错误的关键?

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;    i=10 j=15
```

```
    swap(&i, &j);
```

```
    cout << "i=" << i << " j=" << j << endl;    i=15 j=10
```

或 死机或其它非正常现象

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

提示: 5000-5003系统
是否分配给了程序?



§ 12. 指针进阶 – 基础知识题

```
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;    i=10 j=15
```

```
    swap(&i, &j);
```

```
    cout << "i=" << i << " j=" << j << endl;    i=15 j=10
```

```
}
```

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：哪句是错误的关键？

另2：int *t 改为 int tt, *t;

t = &tt;

为什么就正确了？

i	10	2000
		2003

j	15	2100
		2103

x	2000	3000
		3003

y	2100	3100
		3103

t	(假设5000)	4000
		4003

	10	5000
		5003

提示：5000-5003系统
是否分配给了程序？

或 死机或其它非正常现象



§ 12. 指针进阶 – 基础知识题

```
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;
```

```
}
```

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：哪句是错误的关键？

另2：int *t 改为 int tt, *t;
t = &tt;

为什么就正确了？

i	15	2000
		2003

j	15	2100
		2103

x	2000	3000
		3003

y	2100	3100
		3103

t	(假设5000)	4000
		4003

	10	5000
		5003

i=10 j=15

i=15 j=10

或 死机或其它非正常现象

提示：5000-5003系统
是否分配给了程序？



§ 12. 指针进阶 – 基础知识题

```
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: 哪句是错误的关键?

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;    i=10 j=15
```

```
    swap(&i, &j);
```

```
    cout << "i=" << i << " j=" << j << endl;    i=15 j=10
```

```
}
```

或 死机或其它非正常现象

i	15	2000
		2003

j	10	2100
		2103

x	2000	3000
		3003

y	2100	3100
		3103

t	(假设5000)	4000
		4003

	10	5000
		5003

提示: 5000-5003系统
是否分配给了程序?



§ 12. 指针进阶 – 基础知识题

```
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;    i=10 j=15
```

```
    swap(&i, &j);
```

```
    cout << "i=" << i << " j=" << j << endl;    i=15 j=10
```

```
}
```

或 死机或其它非正常现象

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

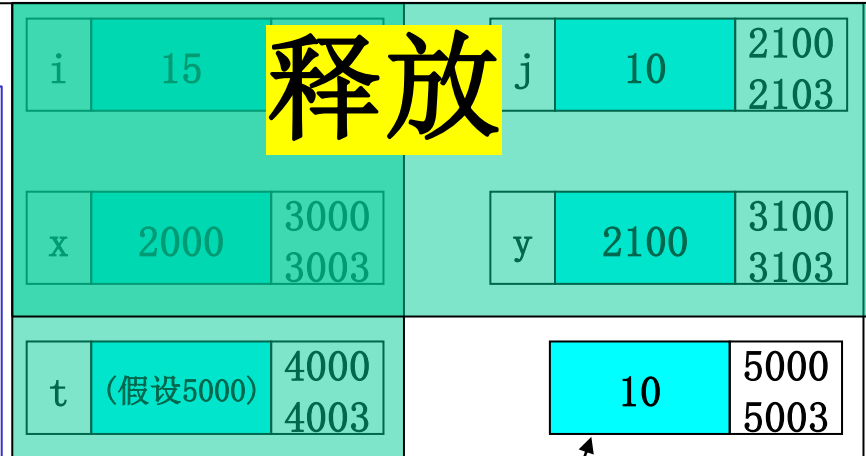
初始内存分配如图所示，请自行画出
swap中三句话执行时内存的变化，理解为什么出现严重错误

另1：哪句是错误的关键？

另2：int *t 改为 int tt, *t;
t = &tt;

为什么就正确了？

释放



提示：5000-5003系统
是否分配给了程序？

另1：int *t;

另2：因为此时 *t 对应的地方是正常分配了的空间。