§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.16-19/P.27-30)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.31（如何同时得到周长和面积）

★ PDF课件的P.32（为什么无法进行交换）

★ PDF课件的P.33（为什么会出现错误，导致错误的关键语句是哪一句）

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"文档作业"中提交（11.28前）

友情提醒：大家尽快把VS2022调试的文档作业做完(除"字符指针"/"引用"知识点外，其它已经全部讲过了)，千万不要卡DDL!!!

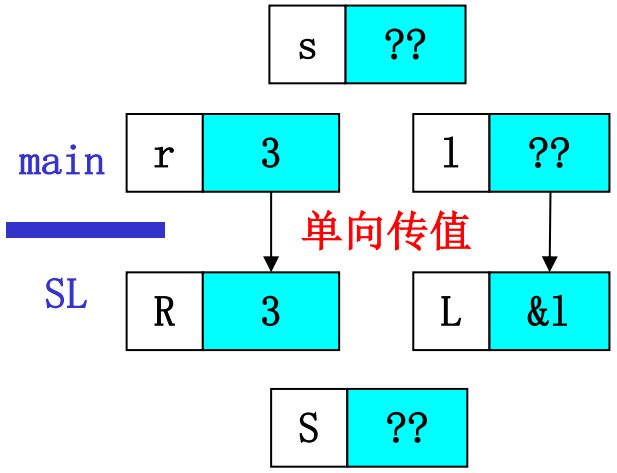
§6.指针基础 - 画内存图并分析

★ PDF课件的P.31（如何同时得到周长和面积）

```
#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!!

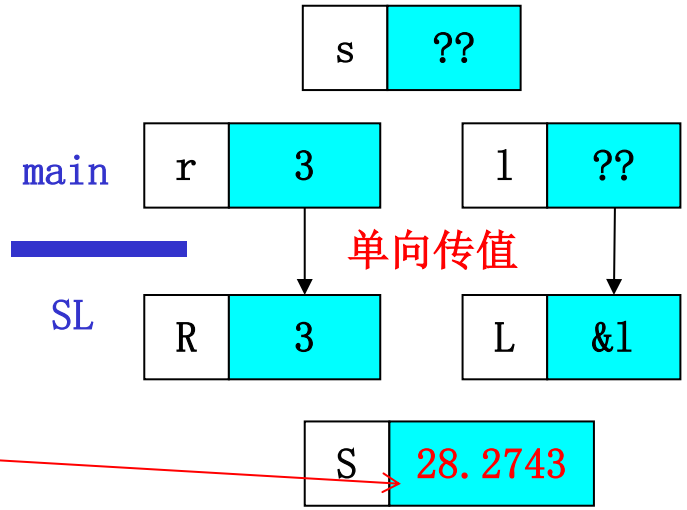
s=28.2743
l=18.8495



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

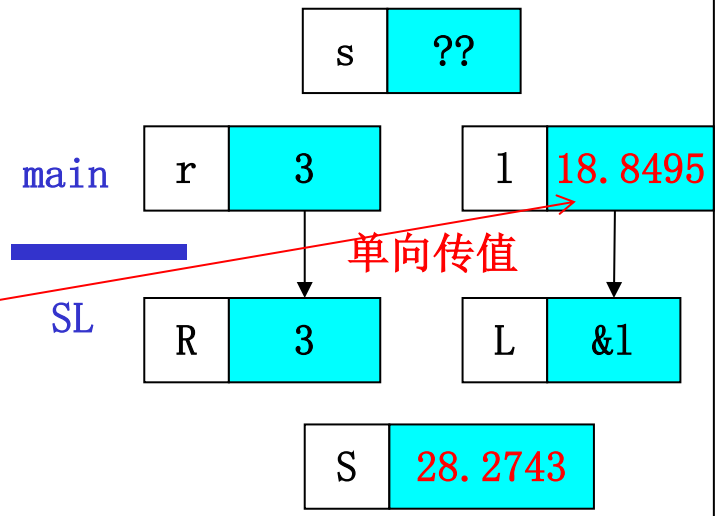
```
#include <iostream>
using namespace std;

#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;
}
```

s=28.2743
l=18.8495



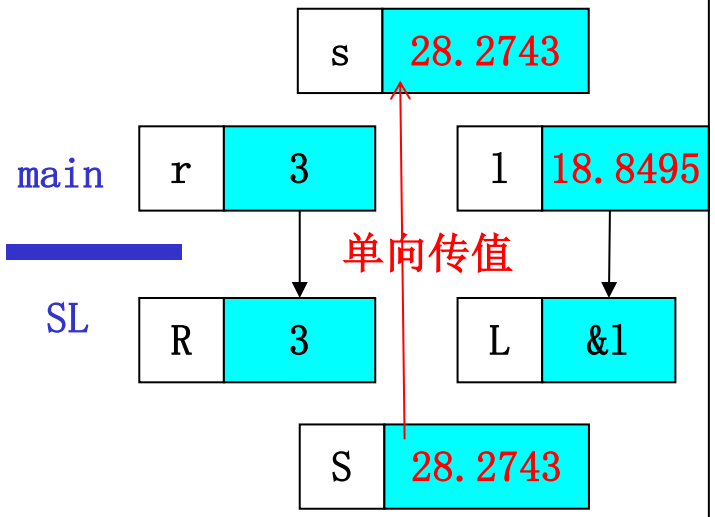
```
#include <iostream>
using namespace std;

#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;
}
```

s=28.2743
l=18.8495



```
#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;
}
```

s=28.2743
l=18.8495

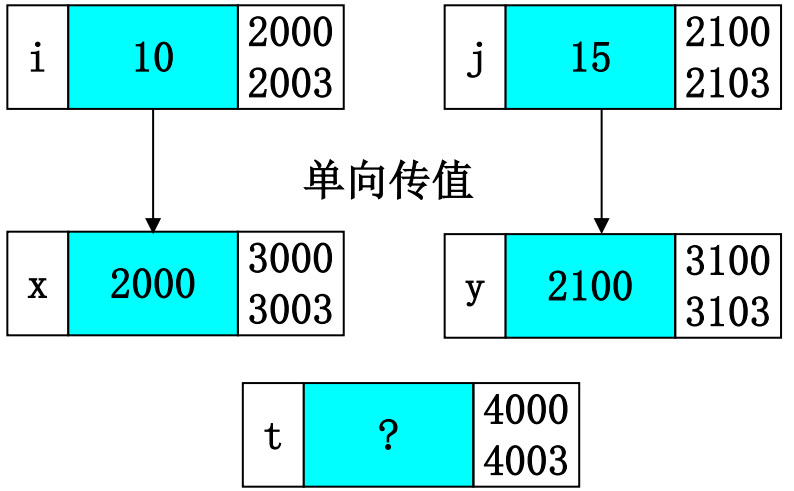


分析：
1.s的值是通过形参值S回传得到的；
2.l的值则是通过将地址传给形参，通过形参来间接访问实参，从而改变实参值。

§6. 指针基础 – 画内存图并分析

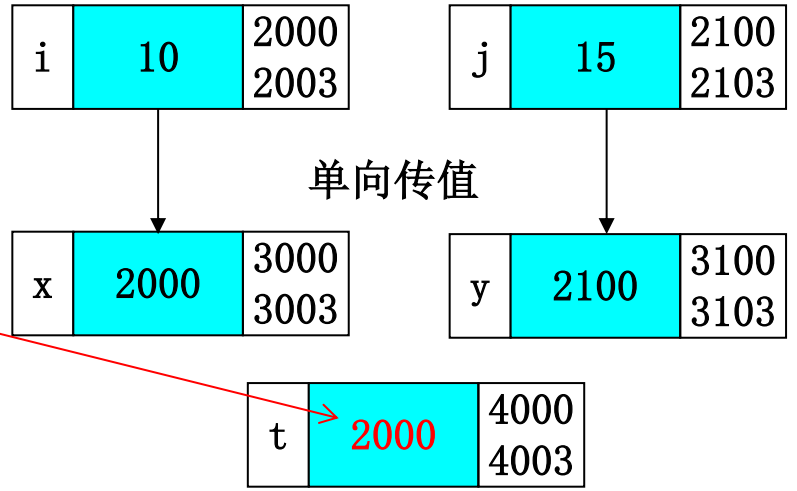
★ PDF课件的P.32（为什么无法进行交换）

```
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=10 j=15
}
```

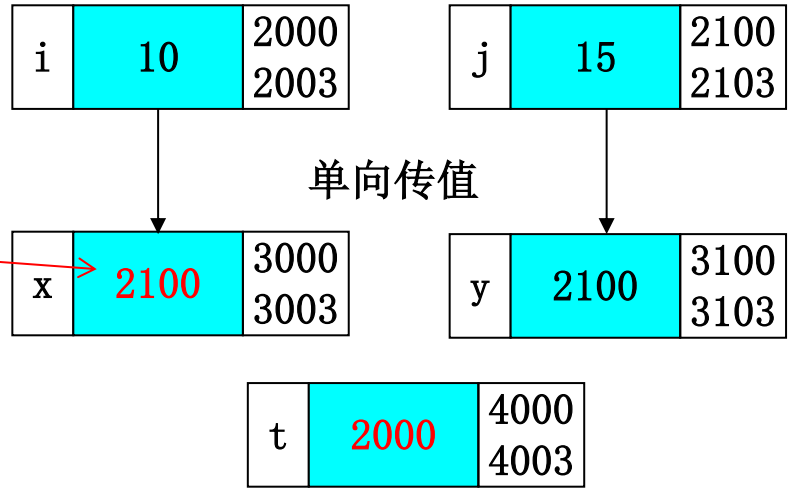


初始内存分配如图所示
请自行画出swap中三句话
执行时内存的变化
理解为什么无法交换

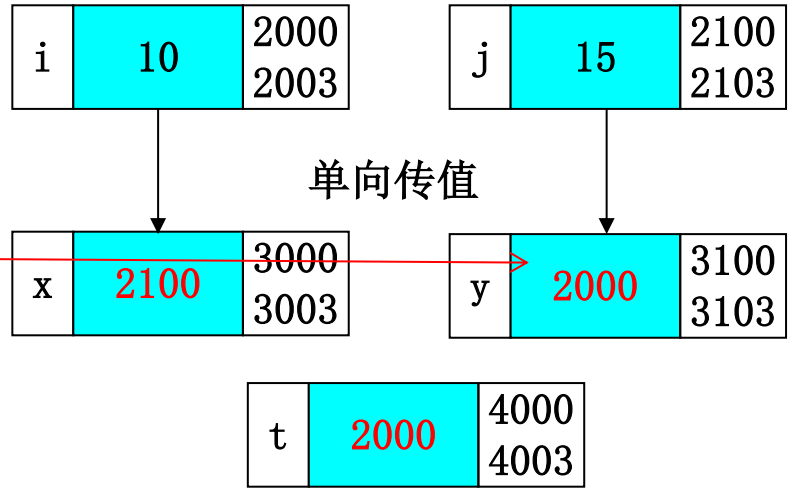
```
void swap(int *x, int *y)
{    int *t;
     t = x;
     x = y;
     y = t;
}

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=10 j=15
}
```



```
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=10 j=15
}
```



```
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=10 j=15
}
```



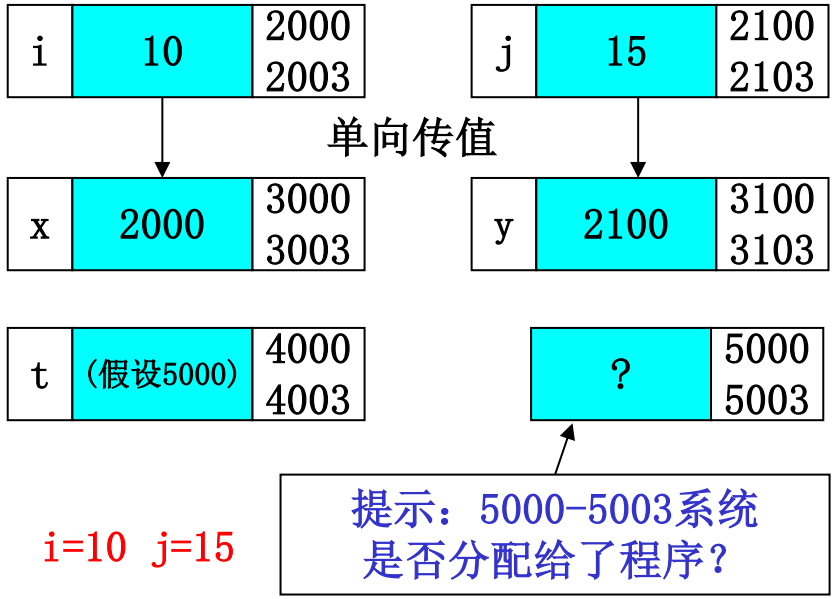
分析：
根据内存状态图的显示，可以发现，在swap函数中只是三个地址互相传递，并没有通过地址访问并修改其对应的实参值，因此没有对实参值进行改变，故无法交换。

§6.指针基础 – 画内存图并分析

★ PDF课件的P.33（为什么会出现错误，导致错误的关键语句是哪一句）

```
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;
}
```

VS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？



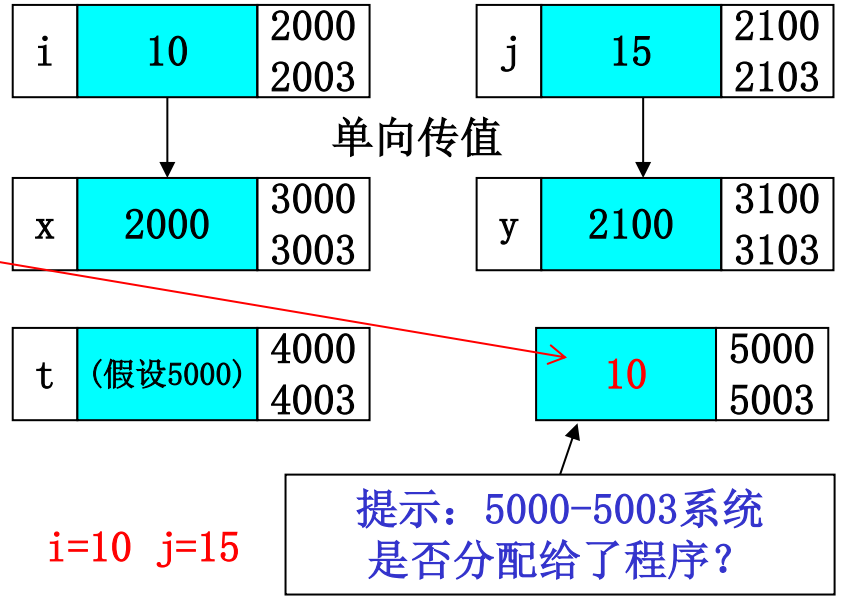
i=10 j=15

i=15 j=10

或 死机或其它非正常现象

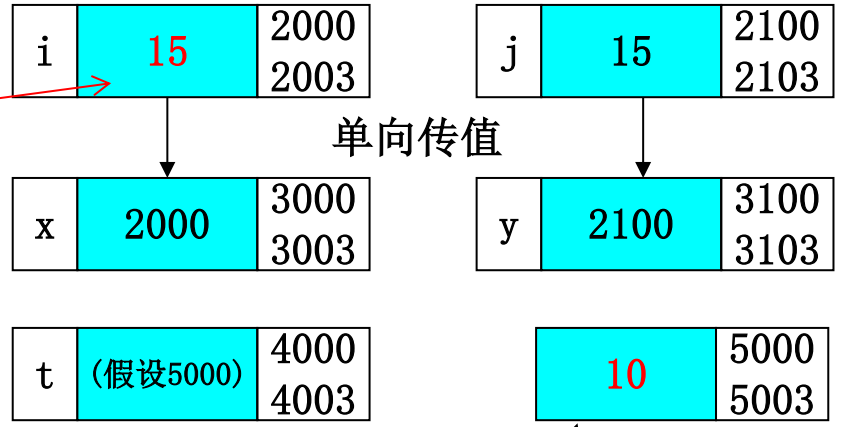
```
void swap(int *x, int *y)
{    int *t;
     *t = *x;
     *x = *y;
     *y = *t;
}

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



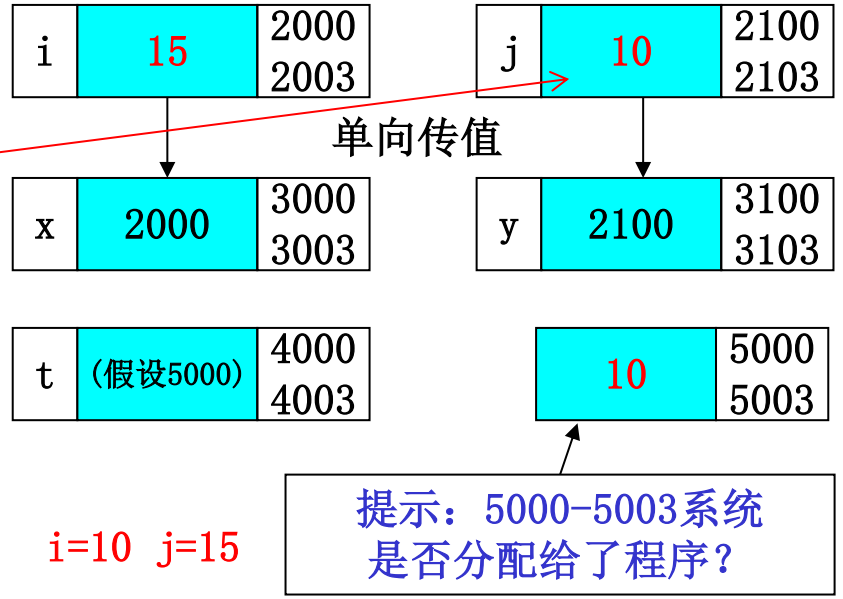
```
void swap(int *x, int *y)
{    int *t;
     *t = *x;
     *x = *y;
     *y = *t;
}

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



提示：5000-5003系统
是否分配给了程序？

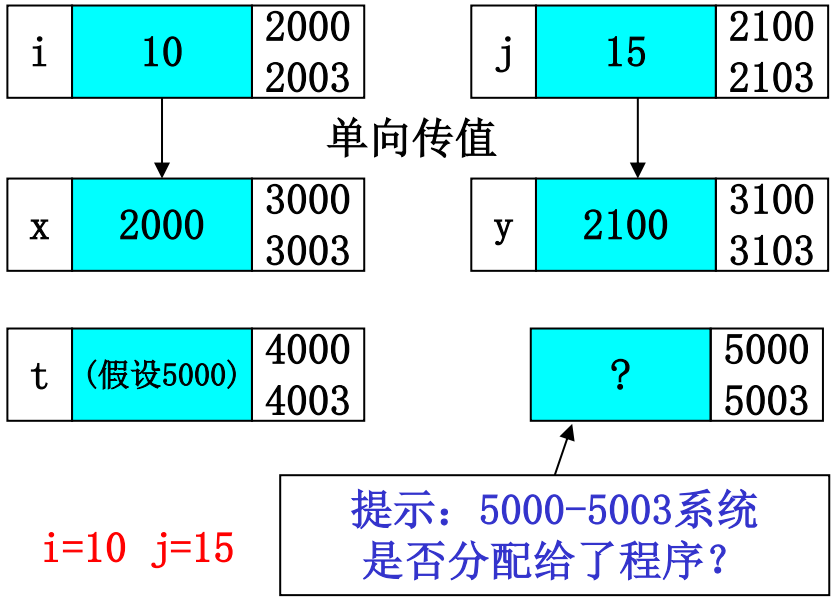
```
void swap(int *x, int *y)
{    int *t;
     *t = *x;
     *x = *y;
     *y = *t;
}

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

```
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
```

VS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？



分析：

另1：出现错误的关键就是--t内存储的某地址指向的内存并没有分配给程序，t内的地址也是随机分配的，*t相当于随机访问了其他内存，并对其进行了私自更改。这个问题可以和数组的越界非法读写一起理解，可能会导致严重错误，在程序运行时可能会弹窗或者出现其他非正常现象。（所以 *t=*x是错误的关键）

另2：改为int tt, *t; t = &tt;正确的原因是，tt这个变量的内存空间分配给了程序，可以对其进行读写操作。