

一、

1.输出：1 2 4 3 2

2 和 3

语句 1 对应定义 1，因为这个 a 之前除了全局变量 a=1 没有其他定义

语句 2 对应定义 3，在 main 函数里面，把 a=1 覆盖掉了

语句 3 对应定义 2，括号里的 int a=4，所以输出 4

语句 4 对应定义 4，在 func 函数外面，所以输出 3

语句 5 对应定义 3，main 函数里面

二、

1. 在 main 函数中，使用了 using namespace x3;，这使得 x3 命名空间中的所有名称都在 main 函数中可见。然而 using namespace x3; 并不会覆盖全局变量，无法判断是全局 a 还是 x3 的 a，所以会报错

2.

```
cout << ::a << endl; (定义 0)
```

```
cout << x1::a << endl; (定义 1)
```

```
cout << x1::x2::a << endl; (定义 2)
```

三、

1.

#include < xxx > 编译时会直接在 include 文件夹查找 xxx 文件进行编译，如果找不到就会报错。

#include "xxx" 编译时首先查找用户文件所在目录，查找不到的情况下在 include 安装目录下继续查找。

(可能可以理解为#include < > 是包含标准库的头文件，而 #include "" 是包含用户自定义的头文件。)

2. (需要修改的见黄底)

```
// file1.cpp
#include <iostream>
#include "file.h"
extern int a;
int b = 1;
int main()
{
    std::cout << a << std::endl;
    func();
    return 0;
}

// file2.cpp
#include <iostream>
#include "file.h"
extern int b;
int a = 0;
void func()
{
    std::cout << b << std::endl;
}
```

3.完全错误的！头文件通常用于声明函数和类，而非定义变量。