指针和引用在C++中很常用，但是对于它们之间的区别很多初学者都不是太熟悉，下面来谈谈他们2者之间的区别和用法。

1.指针和引用的定义和性质区别：

(1)指针：指针是一个变量，只不过这个变量存储的是一个地址，指向内存的一个存储单元；而引用跟原来的变量实质上是同一个东西，只不过是原变量的一个别名而已。如：

int a=1;int \*p=&a;

int a=1;int &b=a;

上面定义了一个整形变量和一个指针变量p，该指针变量指向a的存储单元，即p的值是a存储单元的地址。

而下面2句定义了一个整形变量a和这个整形a的引用b，事实上a和b是同一个东西，在内存占有同一个存储单元。

(2)可以有const指针，但是没有const引用；

(3)指针可以有多级，但是引用只能是一级（int \*\*p；合法 而 int &&a是不合法的）

(4)指针的值可以为空，但是引用的值不能为NULL，并且引用在定义的时候必须初始化；

(5)指针的值在初始化后可以改变，即指向其它的存储单元，而引用在进行初始化后就不会再改变了。

(6)"sizeof引用"得到的是所指向的变量(对象)的大小，而"sizeof指针"得到的是指针本身的大小；

(7)指针和引用的自增(++)运算意义不一样；

2.指针和引用作为函数参数进行传递时的区别。

(1)指针作为参数进行传递：

[复制代码](javascript:void(0);)

#include<iostream>

using namespace std;

void swap(int \*a,int \*b)

{

　　int temp=\*a;

　　\*a=\*b;

　　\*b=temp;

}

int main(void)

{

　　int a=1,b=2;

　　swap(&a,&b);

　　cout<<a<<" "<<b<<endl;

　　system("pause");

　　return 0;

}

[复制代码](javascript:void(0);)

结果为2 1；

用指针传递参数，可以实现对实参进行改变的目的，是因为传递过来的是实参的地址，因此使用\*a实际上是取存储实参的内存单元里的数据，即是对实参进行改变，因此可以达到目的。

再看一个程序;

[复制代码](javascript:void(0);)

#include<iostream>

using namespace std;

void test(int \*p)

{

　　int a=1;

　　p=&a;

　　cout<<p<<" "<<\*p<<endl;

}

int main(void)

{

int \*p=NULL;

test(p);

if(p==NULL)

cout<<"指针p为NULL"<<endl;

system("pause");

return 0;

}

[复制代码](javascript:void(0);)

运行结果为：

0x22ff44 1

指针p为NULL

大家可能会感到奇怪，怎么回事，不是传递的是地址么，怎么p回事NULL？事实上，在main函数中声明了一个指针p，并赋值为NULL，当调用test函数时，事实上传递的也是地址，只不过传递的是指地址。也就是说将指针作为参数进行传递时，事实上也是值传递，只不过传递的是地址。当把指针作为参数进行传递时，也是将实参的一个拷贝传递给形参，即上面程序main函数中的p何test函数中使用的p不是同一个变量，存储2个变量p的单元也不相同（只是2个p指向同一个存储单元），那么在test函数中对p进行修改，并不会影响到main函数中的p的值。

如果要想达到也同时修改的目的的话，就得使用引用了。

2.将引用作为函数的参数进行传递。

在讲引用作为函数参数进行传递时，实质上传递的是实参本身，即传递进来的不是实参的一个拷贝，因此对形参的修改其实是对实参的修改，所以在用引用进行参数传递时，不仅节约时间，而且可以节约空间。

看下面这个程序：

[复制代码](javascript:void(0);)

#include<iostream>

using namespace std;

void test(int &a)

{

　　cout<<&a<<" "<<a<<endl;

}

int main(void)

{

int a=1;

cout<<&a<<" "<<a<<endl;

test(a);

system("pause");

return 0;

}

[复制代码](javascript:void(0);)

输出结果为： 0x22ff44 1

          0x22ff44 1

再看下这个程序：

这足以说明用引用进行参数传递时，事实上传递的是实参本身，而不是拷贝。

所以在上述要达到同时修改指针的目的的话，就得使用引用了。

[复制代码](javascript:void(0);)

#include<iostream>

using namespace std;

void test(int \*&p)

{

　　int a=1;

　　p=&a;

　　cout<<p<<" "<<\*p<<endl;

}

int main(void)

{

int \*p=NULL;

test(p);

if(p!=NULL)

cout<<"指针p不为NULL"<<endl;

system("pause");

return 0;

}

[复制代码](javascript:void(0);)

输出结果为：0x22ff44 1

         指针p不为NULL