指针操作总结

1、new[] 申请的空间，需要delete[]来释放。

2、delete指针后，请一定要将指针置为NULL，因为在析构函数中常会根据指针是否为NULL来决定是否需要delete释放，如果之前已经被delete，但又没有被置NULL，在后面再delete的话会出错。

3、对指针数组，比如原来定义的时候是这样

unsigned char\* pSample[20];

在申请空间的时候，确保不要为pSample[20]、 pSample[21]……申请空间！否则出错，又很难检查出来！！！

4、对指针数组，要一次遍历释放，如：

for (int i=19;i>=0;i--)

{

delete pSample[i];

pSample[i]=NULL;

}

5、留意以下容易犯的错误：

TCHAR\* szBuff=new TCHAR[MAX\_PATH];

szBuff=(TCHAR\*)lpPath;

此处，程序定义了szBuff指针，然后为之分配内存，而后又把指针指向另一处，这导致原来分配的空间没被释放，因而会发生内存泄露！因而应用以下代码代替：

TCHAR\* szBuff=TCHAR\*)lpPath;

6、指向指针的指针的内存申请

PMatrixH=(int \*\*)calloc(GrayLayerNum,sizeof(int\*));

for (int i=0;i<GrayLayerNum;i++)

{

PMatrixH[i]=(int\*)calloc(GrayLayerNum,sizeof(int));

}

释放：

for (int i=0;i<GrayLayerNum;i++)

{

free(PMatrixH[i]);

}

free(PMatrixH);

这里要注意：calloc申请的空间由free释放，new申请的空间用delete释放.

７、指向指针的指针的操作

定义：char \*\*DeviceName

分配空间:

DeviceName=new char\*[256];

for(int i=0;i<256;i++)

DeviceName[i]=new char[2048];

销毁内存

for(int i=0;i<256;i++)

{

Delete[] DeviceName[i];

DeviceName[i]=NULL;

}