(a)

in constructor,nCharArray=0

in constructor,nCharArray=1

in constructor,nCharArray=1

in oprator[],nCharArray=2

abc

pbc

in operator=,nCharArray=2

in operator=,nCharArray=1

pbc

pbc

in destructor,nCharArray=1

in destructor,nCharArray=1

in destructor,nCharArray=0

(b)

**nCharArray** stands for the number of different strings in main function.

No. It isn’t equal to the number of **String** objects. It is less than **String** objects.

(c)

On the first constructor, static members nCharArray isn’t updated.

From (b) we know**nCharArray** stands for the number of different strings in main function. So the copy of same strings can only add the number of **String** objects, but the value of **nCharArray** won’t change.

(d)

str + 1

(e)

析构函数清除内存的时候，输出了两句in destructor,nCharArray=1，也就是析构函数的前两次隐式调用完成后内存才完全清除，if (!str[0])中的条件才为真，即执行两次清除内存后str才为空，所以两个String objects共享一个存储空间。

(I am sorry that my English is not good enough to answer all the questions!)