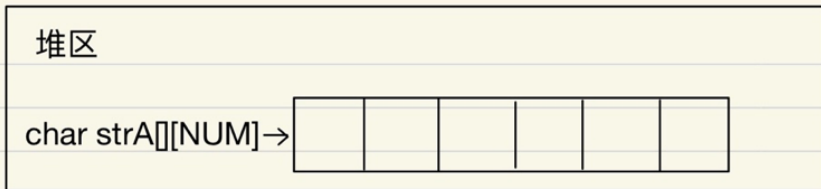


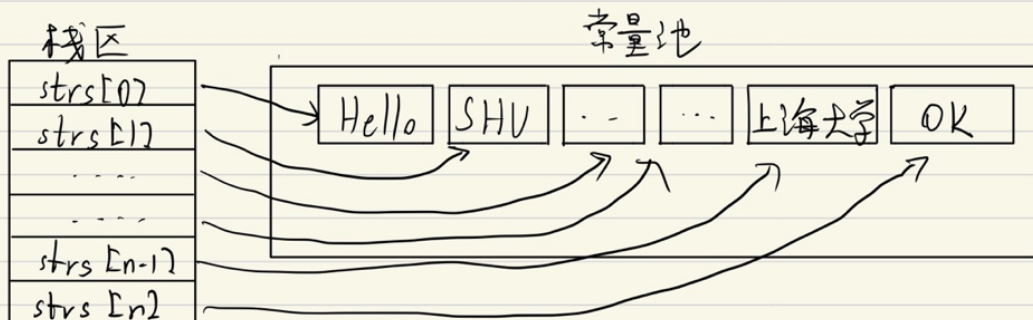
OOP第一周作业

1. 当while遍历source的指针为"\0"时跳出循环；执行函数过程中，传入的参数dest与source的指针同时后移(sizeof(dest)<sizeof(source)将导致dest越界？)
2. 不能，当循环结束时dest已经不再指向字符串的头
3. "/0"
4. str1为常指针，不能用于遍历
5. *p 或者 *q为"/0"，以及当 *p 与 *q 指向的字符不同时相等时
6. "\0" = 0，代表字符串结束
7. 不能，c与cpp不支持连续比较
8. 将字符型数字转化为含义相同的数值型数字
9. 因为要对传入的外部参数直接做出改动，若不使用引用，做出的改动只能以return的方式返回
10. 防止产生野指针
11. 二级：指向多个字符串；引用：将申请的地址返回dest
12. 将二维指针dest申请的堆空间首尾相连，形成连续的内存
13. 将source中一定长度的字符串复制到dest中，防止越界
14. 防止读取字符串越界
15. 二级：指向多个字符串；引用：将传入的strs指针释放
16. 找到以GetString0读取的字符串数组的头指针
17. 释放char** str申请的char *类型空间
18. 防止产生野指针
19. 变化
20. sizeof(strA) : 300 Bytes, sizeof(*strA) : 20 Bytes, n1 = 15
sizeof(strB) : 120 Bytes, sizeof(*strB) : 8 Bytes, n2 = 15
- 21.

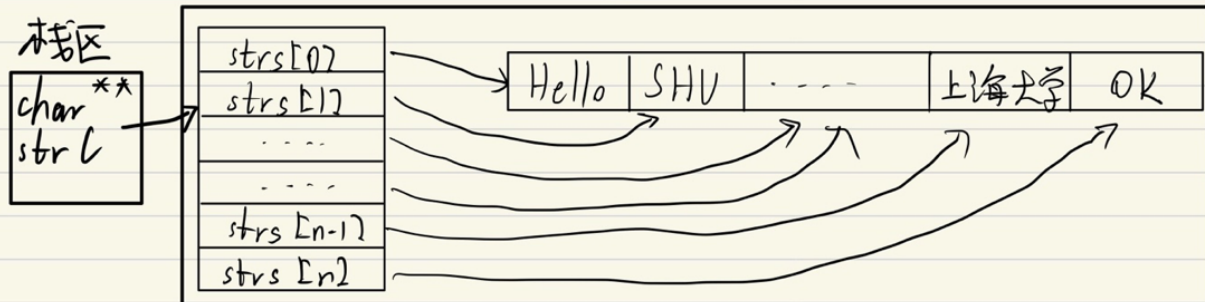
str A:



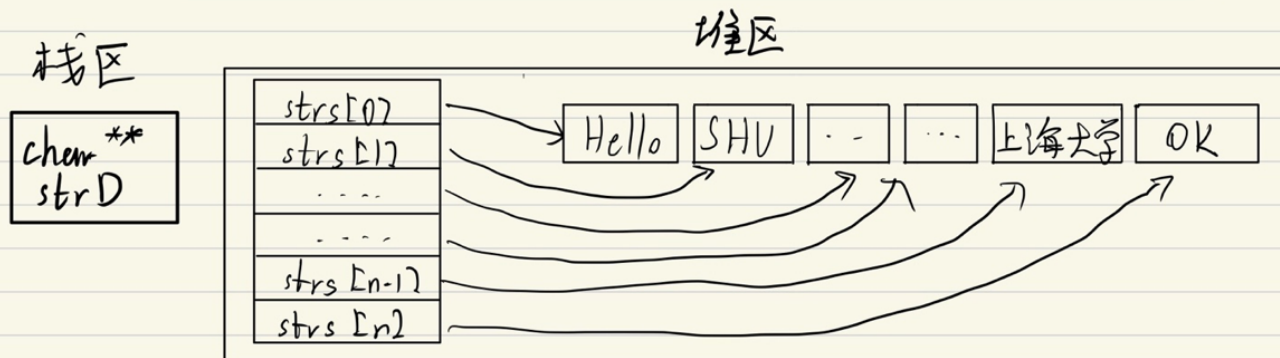
str B:



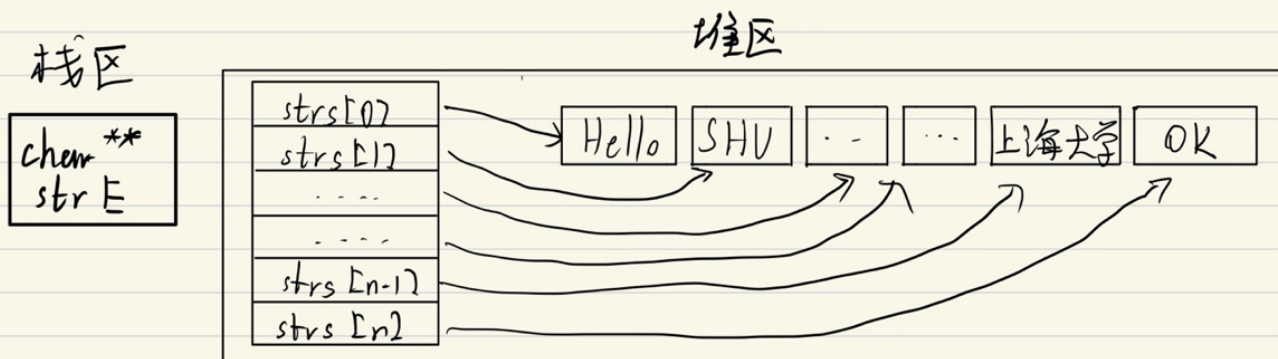
str C:



str D:



str E:



22. strA为二维数组，内存地址连续，BubbleA中交换字符串使用strcpy，而BubbleB仅交换字符串首地址，二维数组不能通过交换strA[i]的方式交换
23. 编译时不检查指针指向的空间是否可更改，因此编译通过；由于Bubble修改strB的内容，而strB指向常量，因此运行出错
24. 虽然strC在内存中连续，但本质上不是二维数组，因此可以通过交换指针的方式排序
25. 因为strC由GetStrings0构造，在内存中连续存放，因此只需要两次delete，而由GetStrings1，GetStrings2构造的变量不连续，需要n+1次delete
26. strC[0]不变，strC[0]的内容改变
27. strD是由GetStrings1构造的，指向位于堆区的指针，因此既可以直接交换指针，也可以交换指针指向的内容
28. strE是由GetStrings2构造的，其字符串的长度与source一一对应，如果使用BubbleA交换指针指向的内容，将会导致内存溢出
29. 可以，不输入参数默认为NULL，与false等价
30. 可以，strC只需要使用两次delete，与n0无关