1. 杰尼龟五场数学考试的成绩依次是“97，89，111，70，90”，如果用冒泡排序作降序排序，那整个排序过程中，97被交换的次数是 （ ）

A.1次

B.4次

C.3次

D.2次

1. 采用冒泡排序对“16,11,8,9,21,32,28”升序排序，共进行数据交换的次数为

A.3次 B.4次

C.5次 D.6次

1. 有如下Python 程序段：

a=[3,1,9,7,6,3]

n=len( a )

for i in range(1,n)：

for j in range(n-2,i-2,-1):

if a[ j ]<a[ j+1 ]:

a[ j ],a[ j+1]=a[ j+1 ]，a[ j ]

则程序运行结束后，数组a的值为 （ ）

1. [9,3,1,7,6,3] B.[9,7,6,3,3,1] C.[1,3,3,9,7,6] D.[1,3,3,6,7,9]
2. b=[56,78,11,31,24,52,66,49]

count = len(b)

for i in range(1,3):

for j in range(0,count-i):

if b[ j ]>b[ j+1 ]:

b[ j ],b[ j+1 ]=b[ j+1 ],b[ j ]

b最后的结果是\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. 使用如下程序段对数组a进行升序排序：

a = [23, 86,98,65,2]

for I in range( 4 ):

k=i

for j in range(i+1,5):

if a[ j ]<a[ k ]:

k=j

if k!=i:

a[ i ],a[ k ]=a[ k ],a[ i ]

1234567——排序

1.冒泡排序，选择排序

2.深刻理解冒泡循环，能正着排，反着排，花式排

3.试了解两种排序的时间复杂度

整个过程中，数组a数据的比较次数和交换次数分别是 （ ）

A.10 3 B.10 4 C.15 3 D.15 10

6.有如下Python程序段：

a=[2,1,9,8,6,3]

cnt=0

for i in range( len(a)-1,0,-1 ):

flag = False

for j in range( i ):

if a[ j ]>a[ j+1 ]:

a[ j ] ,a[ j+1 ] = a[ j+1 ] ,a[ j ]

flag=True

cnt=cnt+1

if not flag:

break

则程序运行结束后，变量cnt的值为 （ ）

A.3 B.4 C.5 D.6

7.使用冒泡排序算法对列表b中的数据进行降序排序，Python程序代码如下，请回答下列问题：

B = [8,21,13,17,25,10]

n = len( b )

for i in range(0,n):

for j in range ( \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ) :

if \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_:

b[ j ],b[ j-1 ] = b[ j-1 ],b[ j ]

print( b )

8. 在经典的选择排序算法中，一趟扫描过程中只能找出最大（或最小）值，将其交换到最左端（或右端）。那么，能否在一趟扫描过程中，同时选出最小值和最大值呢？

可以在参加排序数组的所有元素中同时选出最小值和最大值，分别交换到左、右边界，再缩小左、右边界。然后在余下的元素成为一个有序的序列。这种排序方法被成为双向选择排序算法，效率比普通的选择排序算法要高一些。自定义函数selection\_sort（ a ）能够实现双向选择排序算法功能。

1. 使用双向选择排序算法对数组a排序，已知a的初始值为[4,5,2,6,9,7,1,8],经过第1趟排序后，a 的值变为[1,5,2,6,8,7,4,9]，则经过第2趟排序后，a的值为\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
2. 请在划线处填入合适的代码
3. 若删去加框处的代码，能否正确排序？

def selection\_sort ( a ):

L,R=0,len( a )-1

While L<R:

imin = imax = L

for i in range(L+1,R+1):

if a[ i ]>a[ imax ]:

imax=i

elif a[ i ]<a[ imin ]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

a[ imin ],a[ L ]=a[ L ],a[ min ] 100 1 22 1 1 1 1 1 1 0 1

if imax == L:

imax=imin

a[imax],a[R]=a[R],a[imax]

print(f”第{L+1}趟：“，a)

L=\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

R=\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_