**专题13：排序算法及其实现**

**一、学习目标**：

1. 掌握选择排序和冒泡排序的算法
2. 了解算法vb实现的基本方法和思路
3. 能用VB代码表示简单算法

二、

**三、学习过程：**

1.海盗的金币问题学习重难点：

1、选择和冒泡排序基本算法的理解

2、vb代码实现简单算法

算法描述：

**2：算法实现的基本思路：**

A:

B:

C:

2.冒泡排序算法的程序实现：

“总共进行n-1趟

”每趟从后向前依次比较

判断相邻两个元素大小

相邻两个元素交换

3.选择排序算法的程序实现：

“总共进行n-1趟

第一名选手上擂台

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_”各选手依次上台打擂

打擂比赛

新擂主上擂台

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_判断擂主在不在正确的位置

若不在正确位置，则交换

**真题巩固**

1. 5个学生的身高（单位：cm）依次为177，172，180，168，165，采用冒泡排序对这5个学生的身高数据进行从低到高排序，第二遍加工后的数据为

|  |  |
| --- | --- |
| A. 180，177，172，168，165 | B. 180，177，168，172，165 |
| C. 165，168，172，177，180 | D. 165，168，177，172，180 |

2.5个学生的身高（单位：cm）依次为177，172，180，168，165，采用冒泡排序对这5个学生的身高数据进行从低到高排序，总共需要交换的次数为

|  |  |  |  |
| --- | --- | --- | --- |
| A. 7 | B. 8 | C. 9 | D. 10 |

3.实现某排序算法的部分 VB 程序段如下：

For i = 1 To 4

For j = 8 To i + 1 Step - 1

If d（j）< d（j - 1） Then

t = d（j）: d（j）= d（j - 1）: d（j - 1）= t

End If

Next j

s = s + Str（d（i））

Next i

Text1.Text=s

若数组元素 d（1）到 d（8）的数据依次为“12，7，18，13，9，17，6，23”，运行该程序段后，文本框Text1中显示的内容是

|  |  |  |  |
| --- | --- | --- | --- |
| A. 6 7 9 12 | B. 23 18 17 13 | C.12 7 18 13 | D.9 17 6 23 |

4. 实现某排序算法的部分VB程序如下：

For i＝1 To 6

　k＝i

　For j＝i＋1 To 7

If a(j)＜a(k) Then k＝j

　Next j

　If i＜＞k Then

t＝a(i)：a(i)＝a(k)：a(k)＝t

　End If

Next i

在排序过程中，经过某一遍排序“加工”后，数组元素a(1)到a(7)的数据依次为“10，41，75，12，63，11，85”。下一遍排序“加工”后数组元素a(1)到a(7)的数据依次是

|  |  |
| --- | --- |
| A. 10，11，41，75，12，63，85 | B. 10，11，75，12，63，41，85 |
| C. 10，11，12，75，63，41，85 | D. 10，11，12，41，63，75，85 |

5.有以下选择排序算法的VB程序段：

k=0

For i = 1 to 4

For j = i+1 to 5

If a(j)<a(i) Then

t=a(j):a(j)=a(i):a(i)=t

k=k+1

Next j

Next i

已知数组元素a(1)到a(5)的值依次为“12，41，75，10，63”，经该程序段加工后，k的值为

|  |  |  |  |
| --- | --- | --- | --- |
| A.2 | B.3 | C.4 | D.5 |

6．某VB程序段如下：

i = 1

Do while i<=3

min＝i

j=i+1

Do while j<=5

If a(j)＜a(min) Then min＝j

　 j=j+1

Loop

t＝a(i)：a(i)＝a(min)：a(min)＝t

i=i+1

Loop

数组元素a(1)到a(5)的值依次为“7，8，3，9，6”，则程序段运行后，数组元素a(1)到a(5)的值依次为

|  |  |  |  |
| --- | --- | --- | --- |
| A.3,6,7,9,8 | B.9,8,7,3,6 | C.3,6,7,8,9 | D.9,8,7,6,3 |