APOLLO

B2 Programming
SL

Computer Science
First assessment 2027
2 questions | mark schemes

2. (a) *Award **[3 max]***

It is easier to work with one single structure than several individual ones;
Individual elements that have the same value can be quickly accessed via their indices;
Iteration through arrays is more efficient / Methods associated with arrays can make operations more efficient;

(b) *Award **[4 max]***

repeat for each entry/index in the `Name` array
    identify the position of the space in each name
    save the substring to the left as s1 and the substring to the right as s2
    save the new entry as s2+space+s1

(c) *Award **[1]***

```
MAT[1][2];
```

(d) *Award **[1]** each;*

The value of a is: 6
The value of b is: 17

(e) *Award **[5 max]** as follows:*

*Award **[1]** for initial variables correctly set;*
*Award **[1]** for both loops;*
*Award **[1]** for checking seat for* `False`*;*
*Award **[1]** for updating row value;*
*Award **[1]** for resetting* `tempTotal`*;*
*Award **[1]** for printing the correct row;*

```
public void least(boolean[][]seats)
{
    int falseTotal = 0;
    int leastRow = 0;

    for (int x = 0; x < 40; x++)
    {
        int tempTotal = 0;
        for (int y = 0; y < 30; y++)
        {
         if (seats[x][y] == false)       // if(!seats[x][y] )
            tempTotal = tempTotal + 1;
        }
        if (tempTotal > falseTotal)
        {
            leastRow = x;
            falseTotal = tempTotal;
        }
    }
    System.out.println ("Row with least seats sold is " + leastRow);
}
```

5. (a) *Award **[3 max]***

It is easier to work with one single structure than several individual ones;
Individual elements that have the same value can be quickly accessed via their —indices;
Iteration through arrays is more efficient / Methods associated with arrays can make operations more efficient;

(b) *Award **[4 max]** as follows:*

repeat for each entry/index in the `Name` list
identify the position of the space in each name
save the substring to the left as s1 and the substring to the right as s2
save the new entry as s2+space+s1

repeat for each entry/index in the `Name` list
use the `split()` function
to separate the 2 strings into 2 list values (s1 and s2)
save the new entry as s2+space+s1

(c) *Award **[1]***

```
MAT[1][2];
```

(d) *Award **[1]** for each:*

The value of a is: 6
The value of b is: 17

(e) *Award **[5 max]** as follows:*

*Award **[1]** for initial variables correctly set;*
*Award **[1]** for both loops;*
*Award **[1]** for checking seat for* `False`*;*
*Award **[1]** for updating row value;*
*Award **[1]** for resetting* `temp_total`*;*
*Award **[1]** for printing the correct row;*

```
def least(seats):

    false_total = 0
    least_row = 0

    for x in range(40):
        temp_total = 0
        for y in range (30):
            if seats[x][y] == False:
                temp_total = temp_total + 1
        if temp_total > false_total:
            least_row = x
            false_total = temp_total

    print ("Row with least seats sold is ",least_row)
```