Unit 5 – ICS3U0 –Arrays & Algorithms

Sample Test – December 14, 2022

Name: Solutions

Total	%	Knowledge	Communication	Thinking	Application
(106)	%	(24)	(29)	(22)	(25)

Knowledge

0. This array holds the colours of the rainbow. /6

```
String rain[]={"red", "orange", "yellow", "green", "blue", "indigo", "violet"};
```

(a) Fill in the array memory diagram.

[0]	[1]	[2]	[3]	[4]	[5]	[6]
red	orange	yellow	green	blue	indigo	violet

(b) What is rain.length? ...7.....

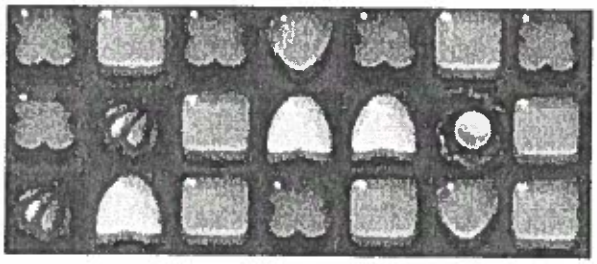
(c) What is in rain[1]? .orange...

(d) What is the index of indigo? ...5.....

(e) What is in element 3? green...

(f) Why would rain[22] produce an error? .index..off..end..of..array.....

1. Consider the game of Scrubby Dubby. /7



These numbers were assigned to the pictures.

(a) What is in ocean[0][0]? ..4.....

(b) What is in ocean[1][4]? .2.....

(c) What is in ocean[2][5]? ..1.....

(d) What is in ocean[5][2]? .error..

(e) Where is the open oyster (6)? ocean[.1..][5.]

(f) How many rows? ..3....

(g) How many columns? ..7.....

2. Fill in the neighbours chart for the center element in the ocean array. /4

	Up ocean[i-1][j]	
Left ocean[i][j-1]	Clicked Element ocean[i][j]	Right ocean[i][j+1]
	Down ocean[i+1][j]	

3. Convert each method so they work with an int array. /6

```
for (int i = 0 ; i < a.length ; i++)
    System.out.print (a [i] + " ");

System.out.println ();
```

```
String findMe = IO.inputString("Find what? ");
int pos = -1;
for (int i = 0 ; i < a.length ; i++)
{
    if (a [i].equals(findMe))
       pos = i;
}
if(pos==-1)
    System.out.println("Didn't find it");
else
    System.out.println("In position "+ pos);
```

```
public void bubble(String a[])
{
  for (int i = 0 ; i < a.length-1 ; i++)
  {
    for (int j = 0 ; j < a.length-1-i ; j++)
    {
      if (a [j + 1].compareTo(a [j]) < 0)
      {
        String temp = a [j];
        a [j] = a [j + 1];
        a [j + 1] = temp;
      }
    }
  }
}}
```

Communication

4. Provide the phrase or term indicated. /6

Order	(a) The 'O' in Big-Oh notation stands for this.
Array Size	(b) The 'n' in Big-Oh notation stands for this.
Bin Sort	(c) Which sorting algorithm is not "in-place"?
Bogo Sort	(d) The worst sorting algorithm. By far. No contest.
Trade off	(e) A compromise. You must give up one thing to get another.
Bubble Sort	(f) The algorithm named after the CO_2 rising in glass of pop.

5. Trace the sorting of this array using each sorting method. /8

Bubble Sort

8	5	3	0	4	7	1
5	8	3	0	4	7	1
5	3	8	0	4	7	1
5	3	0	8	4	7	1
5	3	0	4	8	7	1
5	3	0	4	7	8	1
5	3	0	4	7	1	8
3	5	0	4	7	1	8
3	0	5	4	7	1	8
3	0	4	5	7	1	8
3	0	4	5	1	7	8
0	3	4	5	1	7	8
0	3	4	1	5	7	8
0	3	1	4	5	7	8
0	1	3	4	5	7	8

Selection Sort

8	5	3	0	4	7	1
1	5	3	0	4	7	8
1	4	3	0	5	7	8
1	0	3	4	5	7	8
0	1	3	4	5	7	8

Bin Sort

The Original Array:

4	4	1	0	3	2	1	2	1	4	0

Bins:

0	1	2	3	4
2	3	2	1	3

Final Array:

0	0	1	1	1	2	2	3	4	4	4

6. What is the moral of Bentley's example? /1

Good hardware cannot compensate for a bad algorithm.

7. Put these algorithm speeds in order.
(1 is fastest, 7 is slowest)

1. Constant time
4. O(n log n)
7. $O(2^n)$
5. $O(n^2)$
2. O(log n)
3. O(n)
6. $O(n^3)$

8. In Big-Oh, what is the speed of each? /9

(a) Selection Sort $O(n^2)$
(b) Swapping two values $O(1)$
(c) Bubble Sort (average case) $O(n^2)$
(d) Bin Sort $O(n)$
(e) Finding the maximum $O(n)$
(f) Finding the average $O(n)$

9. What is an array? Why are they useful? /2

What? An array is a group of variables all stored under one name.

Why useful? Arrays can be looped through to save many lines of code. Also, they can be searched and sorted. Single variables can not easily (or really at all for large amount).

10. What is the trade-off involved in bin sort? /2

Positive: It is extremely fast → O(n). That's even faster than Quicksort.

Negative: It only works for ①repeated ②integer ③that fall in a small range. It's speed is because it is very specialized.

Thinking

11. Which sorting algorithm would be the best choice? (bubble, selection, bin) /7

Selection (a) `char a[] = {'t', 'f', 't', 't', 'f', 't', 't', 'f', 't', 'f', 't'};`
bin (b) `int b[] = {6, 7, 4, 7, 8, 5, 4, 4, 6, 6, 7, 7, 7, 8, 4, 4, 6, 4, 4, 4, 7};`
bubble (c) `int c[] = {67, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17};`
Selection (d) `String d[] = {"frog", "dog", "bog", "log", "agog", "cog", "blog", "nog", "jog"};`
Selection (e) `double e[] = {1, 2, 0, 3, 2, 2, 0, 0, 2, 3, 3, 1, 0, 0, 0, 1, 0, 0, 1.2, 0};`
bin (f) The arrays is all integers between 8 and 21.
bubble (g) The array was sorted, you added one thing to the end.

12. Circle **and correct** 5 errors in this code. It should find the maximum value in a String array. /5

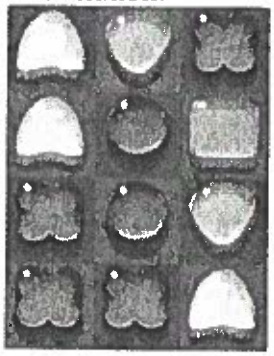
```
String max = a [0];
for (i = 0 ; i < a.length ; i++)
{
    if (a [i].compareTo (max))
       max = a [i];
}
System.out.println ("The largest value is: " + a[2]);
```

int ;) ; >0 ; max

13. Fill in the code for this connect 3 game called Scrubby Dubby. /7

The picture information:

The grid information:



The code:

```
int soap[][]={{3, 1, 2},
              {3, 4, 5},
              {2, 4, 1},
              {2, 2, 3}};

JButton pics[]=new JButton[row*col];   (or 12)
Panel grid=new Panel(new GridLayout(row, col));
int m=0;                               (or 4)
for(int i=0; i< row ; i++){            (or 4)
   for(int j=0; j< col ; j++){         (or 3)
       pics[m]=new JButton
        (createImageIcon( soap [i][j]+".jpg"));
       pics[m].addActionListener(this);
       pics[m].setActionCommand(m+"");
       grid.add(pics[m]);
       m++;
   }
}
add(grid);
```

int row=4;
int col=3;

14. In Scrubby Dubby, the series of the user's moves are recorded in a 3D array. /6
Answer the questions about the array memory diagram shown below.

Move 0		
1	3	4
2	3	1
3	2	1
4	5	1

Move 1		
5	3	4
5	3	1
3	2	1
3	3	2

Move 2		
4	2	4
2	3	1
2	3	1
5	1	2

Move 3		
1	3	4
2	3	3
2	3	1
5	3	2

Move 4		
2	2	4
5	3	3
2	4	1
5	3	2

Move 5		
5	2	3
2	1	4
5	3	1
2	3	1

(a) The first dimension is the **move number**; thus the indices will go from: ...0... to ...5...

(b) The second dimension is the **number of rows** on a game board; thus the indices will go from: ...0... to ...3...

(c) The third dimension is the **number of columns** on a game board; thus the indices will go from: ...0... to ...2...

(d) The array would be declared like this: `int move[][][]=new int[6][4][3];`

(e) What is in `move[1][2][0]`? ...3...

(f) What is in `move[5][3][2]`? ...1...

(g) Print the array on the screen as shown on the right. You will need three nested loops. /5

```
for(int m=0; m<6; m++){
   System.out.println("Move "+m+":");
   for(int i=0; i<4; i++){
      for(int j=0; j<3; j++){
         System.out.print(move[m][i][j]+" ");
      }
      System.out.println();
   }
}
```

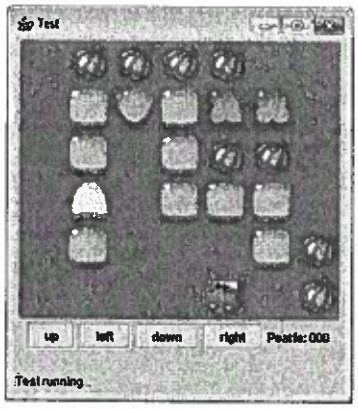
```
Move 0:
1 3 4
2 3 1
3 2 1
4 5 1
Move 1:
5 3 4
5 3 1
3 2 1
3 3 2
Move 2:
4 2 4
2 3 1
2 3 1
5 1 2
Move 3:
1 3 4
2 3 3
2 3 1
5 3 2
Move 4:
2 2 4
5 3 3
2 4 1
5 3 2
Move 5:
5 2 3
2 1 4
5 3 1
2 3 1
```

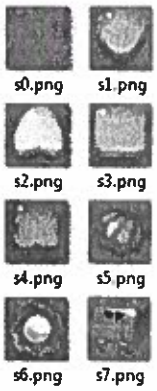
Application

15. Consider this Swimming Hippo Game.

The rules follow:

- The hippo moves around the screen.
- When the hippo passes over a clam (5), the clam opens to reveal a pearl (6).
- When the hippo passes over a pearl (6), the hippo collects it and gains a point. The clam closes.
- When the hippo has 3 pearls, they win.





/8

(a) Fill in the initial values of the global variables.

```
int ocean[] [] = {{0, 5, 5, 5, 5, 0, 0 },
                  {0, 3, 1, 3, 4, 4, 0 },
                  {0, 3, 0, 3, 5, 5, 0 },
                  {0, 2, 0, 3, 3, 3, 0 },
                  {0, 3, 0, 0, 0, 3, 5 },
                  {0, 0, 0, 0, 0, 0, 5 }};
```

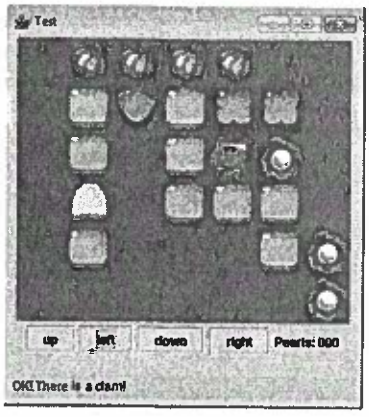
```
int row = 6 ;
int col = 7 ;

//Initial hippo position
int x = 5 ;
int y = 4 ;

//Collected pearls
int pearlCount = 0;
```

(b) Code the up portion of actionPerformed.

```
if (e.getActionCommand ().equals ("up"))
{
    if (x - 1 < 0)
        showStatus ("Off the board!!");
    else
    {
      //if there is a pearl
      if (ocean [x - 1] [y] == 6 )
      {
          showStatus ("OK! There is pearl!");
          pearlCount++;
          //Change the space to have a clam
          ocean [x - 1] [y] = 5 ;
          pearls.setText ("Pearls: " + pearlCount);
      }
      //else if there is a clam
      else if (ocean [x - 1] [y] == 5 )
      {
          showStatus ("OK! There is a clam!");
          //Change the space to have pearl
          ocean [x - 1] [y] = 6 ;
      }
      //else if it is open water
      else if (ocean [x-1] [y] == 0 )
          showStatus ("Just keep swimming!");
      else
          showStatus ("Scrubby Dubby");
      pics [x * col + y].setIcon (createImageIcon ("s" + ocean [x] [y] + ".png"));
      x--;
      //Change to hippo picture
      pics [x * col + y].setIcon (createImageIcon ("s7.png"));
    }
}
```



(c) Code the win portion of actionPerformed.

```
if (pearlCount == 3 )
    showStatus ("You win!!");
```

16. The picture names from one row from Scrubby Dubby is encoded in an int array as follows: /12

(a) Declare an array for these **integers**. Put the picture numbers in it.

```
int a[] = {2,4,4,1,3,3,2,2,4,2};
```

(b) Write the code needed to print the pictures names on the screen, using a loop.
The first four are shown on the side, your printed list should be formatted in exactly this way.

```
for (int i=0;  i<a.length;  i++)
   System.out.println (a[i]+ ".jpg");
```

```
2.jpg
4.jpg
4.jpg
1.jpg
```

(c) Write the code needed to find the average of the array and print it on the screen when finished.

```
int avg =0;
for (int i=0; i<a.length; i++)
   avg += a[i];
System.out.println ( avg /a.length );
```

(d) Write the code that would swap the zeroth and second values in the array.

```
int temp= a[0];
a[0] = a[2];
a[2] = temp;
```

17. Assume you have an unsorted array with repeated values. For example, it might be:

```
int a [] = {2, 4, 0, 3, 4, 0, 1, 0, 7, 8, 7, 8};
```

Write a program to print out all of the indices of the elements that contain the minimum. In this case, the minimum is 0 and the indices that would be printed are 2 5 7, because they all contain the minimum.

```
int min = a[0];
for(int i=0; i<a.length; i++){
   if (min>a[i])
      min = a[i];
}

for(int i=0; i<a.length; i++) {
   if (a[i]== min)
      System.out.print( i+" ");
}
System.out. println ();
```

A better way (with only 1 loop) /5

```
String list = "";
int min = a[0];
for(int i=0; i<a.length; i++){
   if(min >a[i]) {
      min = a[i];
      list = i+" ";
   }
   else if (min==a[i])
      list += i+" ";
}
System.out.println (list);
```