Communication

10. Provide the phrase or term indicated. /6

element	(a) The name for data located at a specific array index.
Array size	(b) The 'n' in Big-Oh notation stands for this.
Bin Sort	(c) Which sorting algorithm is not "in-place"?
j+1 < col	(d) The edge guard for moving right.
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.

11. Are instructions to fold an origami paper crane an algorithm? Provide 3 reasons why or why not. /3

Yes. ① it is a single task → make a paper crane
② it completes the task → if you follow to the end, you get a crane we are done
③ it is a series of steps → folds for crane all listed in order

12. What is the trade-off involved in bubble sort? /2

Positive: it is fast when the array is almost sorted
Negative: in all other cases, its many swaps make it slow

13. 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

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

1 Constant time
4 O(n log n)
7 O(n!)
5 O(n²)
2 O(log n)
3 O(n)
6 O(n³)

15. 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 $O(n)$

Unit 5 – ICS3U0 –Arrays & Algorithms

Sample Test – Thursday May 23, 2024

Name: Solutions

Total	%	Knowledge	Communication	Thinking	Application
(96)	%	(17)	(28)	(21)	(30)

Knowledge

0. This array holds the colours of the rainbow. /7

```
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 in rain[22]? N/A
(e) What is in element 3? green
(f) What is the index of indigo? 5
(g) What are rain's indices? 0 to 6

1. The picture names from one row from Swim Hippos is encoded in an int array as follows: /9

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

```
int swim[] = { 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< swim.length; i++)
    System.out.println(swim[i] + ".jpg");
```

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

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

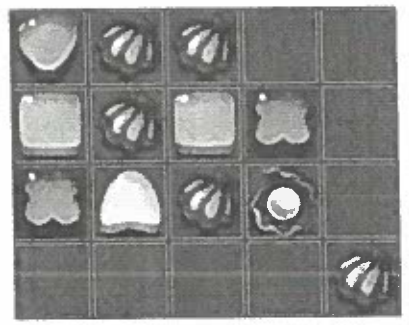
```
int temp = swim[0];
swim[0] = swim[2];
swim[2] = temp;
```

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

Good hardware cannot compensate for a slow algorithm.

Application

3. This is the first level in the Swim Hippo Game. /7



(a) What is in ocean[0][0]? 1

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

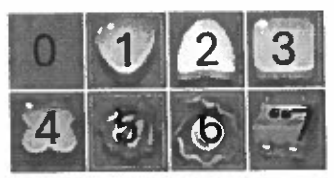
(c) What is in ocean[1][2]? 3

(d) What is in ocean[4][2]? error

(e) Where is the open oyster (6)?

ocean[2][3]

Picture information:



(f) How many rows? 4

(g) How many columns? 5

4. Write the code for the init method in the Swim Hippo Game.
The Hippo's opening position is shown below. /6

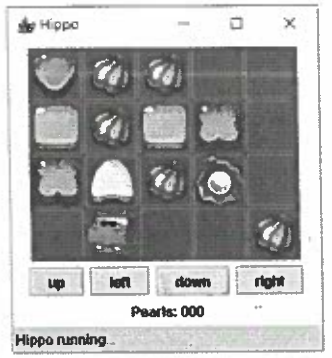
```
int ocean [][]={{1, 5, 5, 0, 0},
                {3, 5, 3, 0, 0},
                {4, 2, 5, 6, 0},
                {0, 0, 0, 0, 5}};
int row = 4;    int col = 5;
int levelTotal = 3;
//Initial hippo position (on screen to right)
int x = 3;    int y = 1;
//Collected pearls
int pearlCount = 0;

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

5. Fill in the neighbours chart for the ocean array. /4

	Up ocean[x-1][y]	
Left ocean[x][y-1]	Clicked Element ocean[x][y]	Right ocean[x][y+1]
	Down ocean[x+1][y]	

6. The rules of the Swim Hippo follow:
- The hippo moves around the screen.
- When the hippo collects 3 pearls, and their pearl count is 3, they win.
- 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 (5).

(a) Code the up portion of actionPerformed. /7




```
if (e.getActionCommand().equals("up")) {
   //edge guard for 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 water
     else if (ocean[x-1][y] == 0)
         showStatus("Just keep swimming!");
     else
         showStatus("Don't swim there hippo!");

     //Move character on screen
     pics[x*col+y].setIcon(createImageIcon(ocean[x][y]+".png"));
     x--;
     //Change to hippo picture
     pics[x*col+y].setIcon(createImageIcon("7.png"));
     }
}
```

(b) Code the win portion of actionPerformed. /1

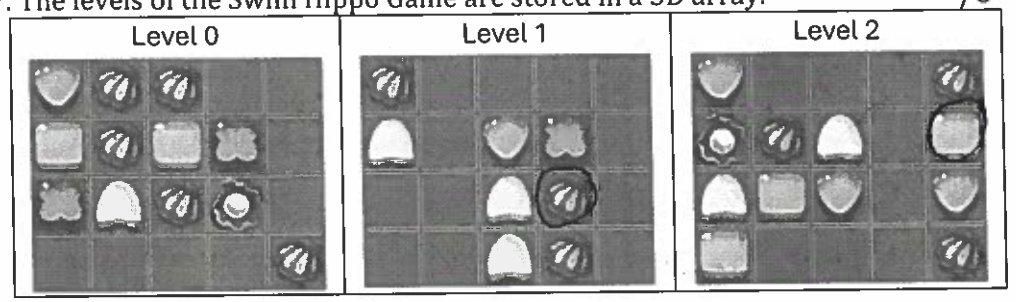
```
if (pearlCount == 3)
    showStatus("You win!!");
```

(c) Write code that would shuffle the values in the ocean array. /5

```
for(int i=0; i < BIGNum; i++){
  int a = (int) Math.random()*row;
  int b = (int) Math.random()*col;
  int c = (int) Math.random()*row;
  int d = (int) Math.random()*col;
  int temp = ocean[a][b];
  ocean[a][b] = ocean[c][d];
  ocean[c][d] = temp
```

Thinking

7. The levels of the Swim Hippo Game are stored in a 3D array. /6



(a) The 1st dimension is **level number**; its indices will go from: 0 to 2

(b) The 2nd dimension is the **rows**; its indices will go from: 0 to 3

(c) The 3rd dimension is the **columns**; its indices will go from: 0 to 4

(d) What is in level [1] [2] [3]? 5

(e) What is in level [2] [1] [4]? 3

(f) Using three nested loops, count the total closed clams (5). /4

```
int count = 0;
for(int k = 0; k < levelTotal; k++){
   for(int i=0; i<row; i++){
      for(int j=0; j<col; j++){
         if(level[k][i][j]==5)
            count++;
      }
   }
}
System.out.println("The clam count is "+ count);
```

8. Which sorting algorithm would be the best choice? /6

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", "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 array is all integers between 8 and 21.

9. 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) > 0)
        max = a [i];
}
System.out.println ("The largest value is: " + a[2]);
```

int

; ; ;