Unit 5 – ICS3U0 –Arrays & Algorithms

Sample Test – Wednesday December 11, 2024

Name: Gorski

Total	%	Knowledge	Communication	Thinking	Application
(100)	%	(17)	(30)	(23)	(30)

Knowledge

0. This array holds the possible values in the game 2048 /6

```
int nums[] = {2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048};
```

(a) Fill in the array memory diagram.

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]
2	4	8	16	32	64	128	256	512	1024	2048

(b) What is nums.length? ...11.....

(c) What is in nums[1]? ...4.....

(d) What is in nums[11]? .error...

(e) What is in element 4? ...32...

(f) What is the index of 1024? ...9.....

(g) What are num's indices? .0... to .10..

1. A list of a student's marks is recorded in an array as follows: /10

```
89, 94, 95, 82, 90
```

(a) Declare an array for these integers. Put the marks in it.

```
int marks[] = {89, 94, 95, 82, 90};
```

(b) Use your array from part a and the one below to write the loop needed to print the table shown.

```
String assign[]= {"Test 1", "Project 1", "Test 2", "Presentation", "Test 3"};
```

Format your table exactly as shown.

```
for (int i=0; i < marks.length; i++)
{
  System.out.println(assign[i]+": " + marks[i] + "%");
}
```

```
Test 1: 89%
Project 1: 94%
Test 2: 95%
Presentation: 82%
Test 3: 90%
```

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

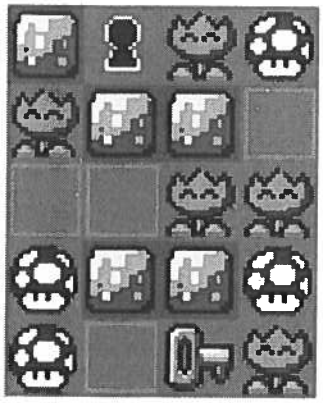
```
int temp = marks[0];
marks[0] = marks[2];
marks[2] = temp;
```

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

fast hardware cannot compensate for a slow algorithm.

Application

3. This is the first level in the Mario Flower Game. /7



(a) What is in land[0][0]? ...1.....

(b) What is in land[1][3]? ...0.....

(c) What is in land[3][1]? ...5.....

(d) What is in land[2][3]? ...2.....

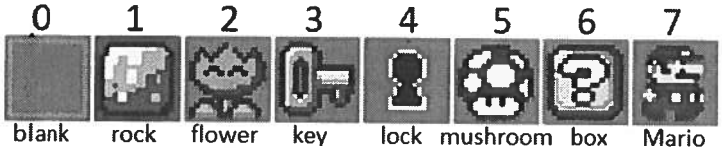
(e) Where is the key (3)?

land[..4..][..2..]

(f) How many rows? ...5.....

(g) How many columns? ...4.....

Picture information:



4. Write the code for the init method in the Mario Flower Game.
Mario's opening position is shown on the screen to the right. /6

```
int land [][]={{1, 4, 2, 6},
               {2, 1, 1, 0},
               {0, 0, 2, 2},
               {6, 1, 1, 6},
               {6, 0, 3, 2}};
int row = 5;     int col = 4;
int levelTotal = 4;
//Initial Mario position (on screen to right)
int x = 2;    int y = 1;
//Collected flowers & key variables
int flowers = 0;    boolean key = false;

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(land[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 land array (above). /4

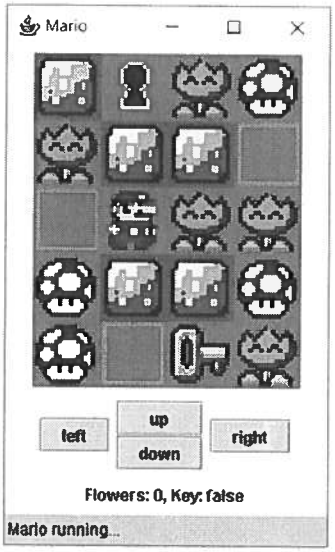
	Up land[x-1][y]	
Left land[x][y-1]	Clicked Element land[x][y]	Right land[x][y+1]
	Down land[x+1][y]	

6. The rules of the Mario Flower Game follow:

- Mario moves around the screen collecting flowers (2), getting a point each time.
- When Mario steps on a mushroom (5), he loses a flower point.
- When Mario goes to the key (3), he picks it up. The key variable becomes true.
- To win, Mario goes to the lock (4). He must have a key (true) and two flowers.

(a) Code the down portion of actionPerformed. /8

```
if (e.getActionCommand ().equals ("down")) {
   //edge guard for down
    if (  x+1 >= row  )
        showStatus ("Off the board!!");
    //down-one space has a rock
    else if (land [x + 1] [y] ==  1  )
         showStatus ("Rock!");
    else {
      showStatus("Good move. ");
      //down-one has a flower
      if (land [x + 1] [y] ==  2  ) {
          flowers++;
          //Change down-one to nothing
          land [ x+1 ] [  y  ] = 0 ;
      }
      //down-one space has a mushroom
      else if (land [x + 1] [y] ==  5  ) {
          flowers--;
          //Change down-one space to nothing
          land [ x+1 ] [  y  ] =  0 ;
      }
      //down-one space has a key
      else if (land [ x+1 ] [  y  ] ==  3  ) {
          key =  true  ;
          //Change the down space to nothing
          land [ x+1 ] [  y  ] =  0 ;
          showStatus ("You found the key!");
      }
      //down-one space has a lock, check for win
      else if (land [ x+1 ] [  y  ] ==  4  ) {
          if(flowers >=  2  && key ==  true  )
              showStatus ("You win!!");
          else
              showStatus ("Need two flowers and a key to win");
      }
      score.setText("Flower: "+ flowers +", Key: "+key);
      pics[x*col+y].setIcon(createImageIcon(land[x][y]+".png"));
      x++;
      //change to Mario picture
      pics[x*col+y].setIcon(createImageIcon("  7  .png"));
      }
   }
```



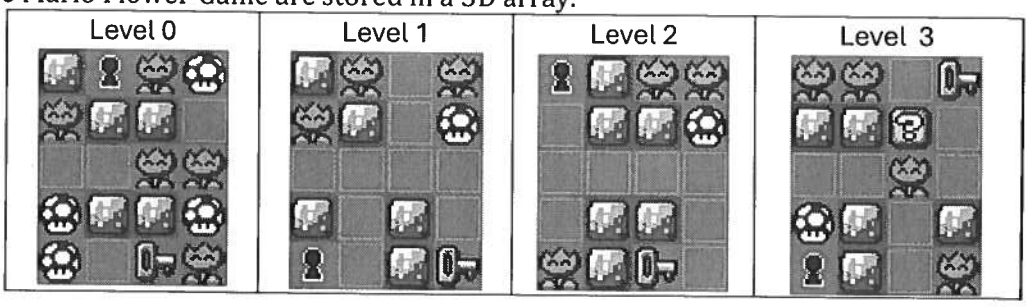
(b) Write a replace function. It looks for all of the instances of "find" in the `land` array and replaces them with whatever is in the "replace" variable. /5

```
int find = IO.inputInt("Find what? ");
int replace = IO.inputInt("Replace with what? ");

for( int i=0; i<row; i++ ){
   for( int j=0; j<col; j++ ){
     if( land [ i ] [ j ] == find )
         land [ i ] [ j ] = replace ;
   }
}
```

Thinking

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



(a) The 1st dimension is **level number**; its indices will go from: ..0.. to ..3..

(b) The 2nd dimension is the **rows;** its indices will go from: ..0.. to ..4..

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

(d) What is in level [2] [4] [0]? ..2..

(e) Where is the ? level[..3..][..1..][..2..]

(f) Using three nested loops, print the co-ordinates of all the locks (4). /4

```
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 ]== 4 )
          System.out.println("["+k+","+ i +","+ j +"]");
    }
  }
}
```

8. Which sorting algorithm would be the best choice? /7

..bubble... (a) String a[] = {"apple", "banana", "cherry", "eggplant", "dragonfruit", "fig"};

..bin..... (b) int b[] = {1, 3, 3, 2, 3, 0, 0, 2, 1, 3, 2, 0 , 2, 2, 1, 0, 3, 0, 1, 2, 1};

..bubble... (c) int c[] = {34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 45, 44, 46, 47, 48, 49, 50};

.selection. (d) char d[] = {'1', '3', '3', '2', '3', '0', '0', '2', '1', '3', '2', '2', '0'};

.....bin..... (e) You want to sort 9000 shoes (sized 2, 3, 4 and 5) by size.

.selection. (f) You want to sort 9000 turnips that weigh between 110.56 and 189.23 grams. You want them sorted by weight.

..bubble.. (g) A class list is sorted and one student is added to the front.

9. Circle **and correct** 6 errors in this code.
It should print the values over 5 in the array named 'a'.. /6

```
int a[] = {10, 2, 8, 4, 5, 6, 7};

System.out.println ("Values Over 5");

for (i = 0 ; i < a.length ; i++)
{
    if (a [i] >= 5)
        System.out.println (a + " ");
```

(3)

Communication

10. Provide the phrase or term indicated. /6

algorithm	(a) Term for a series of steps that complete a task.
order	(b) The 'O' in Big-Oh notation stands for this.
array size	(c) The 'n' in Big-Oh notation stands for this.
sorting	(d) Term for putting elements in order.
selection	(e) The algorithm that repeatedly swaps the max into place.
trade-off	(f) A compromise. You must give up one thing to get another.

11. What is an array? Why is it useful? (At least 3 sentences, provide specific details/evidence) /3

An array is a variable that can store a group of values. They are useful because you can loop through the indices to visit all of the elements (this reduces line of code). Arrays are also needed to sort a list.

12. What is the trade-off involved in bin sort? (Point form is fine, provide details/evidence) /2

Positive: It is very fast (O(n)), faster even than Quicksort!

Negative: It only works in a very specialized situation (ints, small range, repeated values) and it can rarely be used.

13. Trace the sorting of this array using each sorting method. /9

Bubble Sort

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

Selection Sort

9	6	3	7	8	5	4
4	6	3	7	8	5	9
4	6	3	7	5	8	9
4	6	3	5	7	8	9
4	5	3	6	7	8	9
4	3	5	6	7	8	9
3	4	5	6	7	8	9

Bin Sort

The Original Array:

0	3	1	0	3	2	1	2	1	0	0

Bins:

0	1	2	3
4	3	2	2

Final Array:

0	0	0	0	1	1	1	2	2	3	3

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

2. O(log n)
7. O(n!)
5. O(n^2)
4. O(n log n)
6. O(n^3)
3. O(n)
1. O(1)

15. In Big-Oh, what is the speed of each? /10

(a) Swapping two values O(1)
(b) Bin Sort O(n)
(c) Finding the average O(n)
(d) Printing out an array O(n)
(e) Selection Sort O(n^2)
(f) Bubble Sort (average case) O(n^2)
(g) Min O(n)