PROJECT STEM

Name: _____

Block: _____ **Date:**_____

Unit 10 Exam – Online Version

1. Which of the following is not a step in the merge-sort algorithm?
   a. Split the initial list in half
   b. Merge all the smaller arrays together simultaneously into one large array
   c. Merge all the smaller arrays in pairs recursively to make progressively larger arrays
   d. Split each smaller list in half recursively until each list has only 1 element
   e. All of the above are steps of the merge sort algorithm

2. When you pass an int variable to a method, the method receives _____.
   a. a copy of the variable
   b. a copy of the memory address
   c. the reference of the variable
   d. the length of the variable
   e. binary of the memory address

3. Consider the following method.

```
public static void recurMethod(String str)
{
  if (str.length() > 1)
  {
    recurMethod(str.substring(1, str.length() - 1));
  }
  System.out.println(str);
}
```

Which of the following is printed as a result of the call `recurMethod("program")`?

a.
g
ogr
rogra
progra
m

b.
progra
m
rogra
ogr
g

c.
g
ogr
rogra

d.
rogra
ogr
g

e.
m
ma
mar
marg
margo
margor
margor
p

4. Consider the following recursive method.

```
public static void wackyPrinter(String str)
{
  if (str.length() < 10)
  {
    wackyPrinter("!" + str + "!");
  }
  else
  {
    System.out.println(str);
  }
 }
```

What will be printed as a result of the call `wackyPrinter("cpu")`?

a. cpu
b. !!!!cpu!!!
c. !!!!cpu!!!!
d. cpu !cpu! !!cpu!! !!!cpu!!! !!!!cpu!!!!
e. !!!!cpu!!!! !!!cpu!!! !!cpu!! !cpu! cpu

5. Which method(s) would produce the following output if they were passed the parameter, "hamster"?

hamster
hamste
hamst
hams
ham
ha
h

I.
```
public static void mystery(String wo)
{
  System.out.println(wo);
  if (wo.length() > 1)
  {
    mystery(wo.substring(0, wo.length() - 1));
  }
}
```

II.
```
public static void mystery(String wo)
{
  if (wo.length() > 1)
  {
    mystery(wo.substring(0, wo.length() - 1));
  {
  System.out.println(wo);
}
```

III.
```
public static void mystery(String wo)
{
  if (wo.length() > 1)
  {
    mystery(wo.substring( wo.length() - 1));
  }
  System.out.println(wo);
}
```

a. I only
b. II only
c. III only
d. I and III only
e. I, II and III

6. You need to search an unordered list of fifteen students for the student id #1124. Which search would be most effective?

a. binary
b. insertion
c. linear
d. merge
e. selection

7. Consider the following code:

```
int list [] = /* missing code */;
int val = /* missing code */;
int n = -1;
for (int i = 0; i < list.length; i++)
{
  if (val == list[i])
  {
    n = i;
    break;
  }
}
```

What algorithm is shown?

a. Binary Search
b. Insertion Sort
c. Merge Sort
d. Selection Sort
e. Linear Search

8. Consider the following method.

```
public static int goRound(int num)
{
  if (num < 100)
  {
    return 10 * ((num + 5) / 10);
  }
  return 10 * goRound(num / 10);
}
```

What is returned as a result of the call `goRound(56348)`?

a. 61348
b. 60000
c. 50000
d. 61
e. 60

9. Consider the following recursive method:

```
public static int recur(int x)
{
  if (x >= 0)
  {
    return x + recur(x - 1);
  }
  return 0;
}
```

What is returned by the method call `recur(9)`?

a. 0
b. 44
c. 45
d. 46
e. 47

Questions 10-11 refer to the following information.

Consider the following instance variable and methods. You may assume that word has been initialized as a String with length greater than 0. The methods are intended to return true if a character appears twice in a row in word; false otherwise.

```
private String word;

public boolean hasRepeat()
{
  return hasRepeatHelper(word.length() - 1);
}

private boolean hasRepeatHelper(int pos)
{
  // Missing line

  String letter1 = word.substring(pos-1,pos);
  String letter2 = word.substring(pos, pos+1);

  if (letter1.equals(letter2))
  {
    return true;
  }
  else
  {
    return hasRepeatHelper(pos - 1);
  }
}
```

10. For which of the following values of word, will the call `hasRepeat` result in an error?

I. "aadvark"
II. "mississippi"
III. "repeat"

a. I only
b. II only
c. III only
d. II and III only
e. I, II and III

11. Which of the following should be used to replace `// Missing line` in `hasRepeatHelper` so that `hasRepeat` will work as intended?

a.
```
if (pos < 1)
{
  return false;
}
```

b.
```
if (pos < 2)
{
  return false;
}
```

c.
```
if (pos < 0)
{
  return false;
}
```

d.
```
if (pos > word.length() - 1)
{
  return false;
}
```

e.
```
if (pos > word.length())
{
  return false;
}
```

12. Which of the following are real advantages of using the merge-sort algorithm?

I. It is very easy to code
II. It is very fast for large data sets
III. It uses no memory if coded correctly

a. I only
b. II only
c. I and II only
d. II and III only
e. I, II and III

13. Consider the following recursive method.

```
public static String recur(int val)
{
  if (val == 0)
  {
    return "";
  }

  if (val % 2 == 0)
  {
    return recur(val / 2) + "f";
  }
  else
  {
    return recur(val / 2) + "t";
  }
}
```

What is printed as a result of executing the statement `System.out.println(recur(13))`?

a. `t`
b. `f`
c. `ttft`
d. `tftt`
e. Nothing is printed due to infinite recursion

14. Consider the following recursive method.

```
/** Precondition: num > 0 */
public static int mystery(int num)
{
  if (num % 2 == 1)
  {
    return 1;
  }
  else
  {
    return 2 * mystery(num / 2);
  }
}
```

Assume that `int x` has been declared and initialized with a value that satisfies the precondition of the method. Which of the following best describes the value returned by the call `mystery(x)`?

a. The largest multiple of 2 which divides x evenly
b. The largest power of 2 which divides x evenly
c. Nothing is returned. A run-time error occurs because of infinite recursion.
d. The value 1 is returned
e. The value x/2 is returned

15. Consider the following code:

```
public static int mystery( int [] list, int val)
{
  int n = -1;
  int max = list.length - 1;
  int min = 0;
  int mid = 0;

  while (max > min)
  {
    mid = (max + min) / 2;

    if (list[mid] == val)
    {
      n = mid;
      break;
    }

    if (list[mid] > val)
    {
      max = mid-1;
    }
    else
    {
      min = mid+1;
    }
  }
  return n;
}
```

What algorithm is this an implementation of?

a. Binary Search
b. Insertion Sort
c. Merge Sort
d. Selection Sort
e. Sequential Search

16. Consider the following recursive method.

```
public static void doWhat(int num)
{
  if(num < 15)
  {
    doWhat(num + 4);
    System.out.print(num + " ");
  }
}
```

What is printed as a result of the call `doWhat(2)`?

a. 14
b. 2 6 10 14
c. 14 10 6 2
d. 2 6 10 14 18
e. 18 14 10 6 2

17. The two methods below are intended to implement the merge sort algorithm when used in conjunction.

```
/** Main method to sort an array of ints.
 *  @param arr the array to be sorted
 *  @return an array with the same contents as arr, sorted in
 *  increasing order
 */
public static int[] mergeSort(int[] arr)
{
  // Base case: if list length is 1 then it is already sorted
  if (arr.length <= 1)
  {
    return arr;
  }

  // Split list into two half-lists
  int[] lowerHalf = new int[arr.length / 2];
  int[] upperHalf = new int[(arr.length + 1) / 2];
  for (int i = 0; i < lowerHalf.length; i++)
  {
    lowerHalf[i] = arr[i];
  }
  for (int i = 0; i < upperHalf.length; i++)
  {
    upperHalf[i] = arr[i + arr.length / 2];
  }
  return /* missing expression */;
}

/** Helper method which merges two ordered arrays of ints
 *  @param arr1 the first ordered array
 *  @param arr2 the second ordered array
 *  @return an array containing contents of both arr1 and arr2,
 *  sorted into increasing order
 */
private static int[] merge(int[] arr1, int[] arr2)
{
  /* implementation not shown */
}
```

Which of the following calls should replace `/* missing expression */` in the return statement for mergeSort so that the method works as intended?

a. `merge(lowerHalf, upperHalf)`
b. `mergeSort(lowerHalf) + mergeSort(upperHalf)`
c. `merge(mergeSort(lowerHalf), mergeSort(upperHalf))`
d. `mergeSort(merge(lowerHalf, upperHalf))`
e. `mergeSort(upperHalf, lowerHalf)`

18. Consider the following recursive method.

```
public int recur(int x)
{
  if (x > 10)
  {
    return 2 * recur(x / 2);
  }
  if (x < 10)
  {
    return recur(x + 2) / 2;
  }
  return 10;
}
```

What value is returned as a result of the call `recur(12)`?

a. 2
b. 4
c. 5
d. 10
e. Nothing is returned: an error is caused by infinite recursion.

19. Consider the following recursive method.

```
public static void mystery(String s)
{
  if(s.length() > 2)
  {
    mystery(s.substring(0,s.length() / 2));
    mystery(s.substring(s.length() / 2));
  }
  else
  {
    System.out.println(s);
  }
}
```

What is printed as a result of the call `mystery("method")`?

a.
```
me
```

b.
```
me
t
ho
d
```

c.
```
m
et
h
od
```

d.
```
me
th
od
```

e.
```
et
m
od
h
```

20. Consider the following instance variable and method.

```
private ArrayList<Integer> vals;
// vals is sorted into increasing order

public void binaryInsert(int num)
{
  int low = 0;
  int high = vals.size();
  while (high > low)
  {
    int mid = (low + high) / 2;         /* calculate midpoint */
    if (vals.get(mid).intValue() < num)
    {
      low = mid + 1;
    }
    else
    {
      high = mid;
    }
  }
  vals.add(low,new Integer(num)); /* insert new number */
}
```

The binaryInsert method creates and inserts an Integer element into the correct position of the list vals which is sorted into increasing order. Suppose vals contains the following Integer values: [2, 3, 4, 4, 4, 8], and that the following command is executed:

```
binaryInsert(4);
```

What will be the value of low when the line marked `/* insert new number */` in the binaryInsert method is executed?

a. 0
b. 2
c. 3
d. 4
e. 5