

UIL TXCSOpen August Invitationals Written Test

1. Describe the action of `s = s.replaceAll("(^[\\W]{5})\\w+", "$1****");`

- A) Replaces all sequences of greater than five word characters with the first five word characters of that sequence followed by "****"
- B) Replaces all sequences greater than five word characters with "\$1****"
- C) Replaces all sequences of at least 6 word characters with that sequence followed by "****"
- D) Replaces all sequences of 5 than non whitespace characters followed that are followed by at least one word character with that said sequence followed by "****"
- E) Replaces all sequences greater than five non word characters or greater with that sequence followed by "****"

2. Describe the action of `s = s.replaceAll("(\\\\\\\\\\\\\\\\)+", "$1.*");`

- A) This will add on a .* after every instance of four backslashes in a row in s.
- B) This will replace any sequence of more than zero backslashes that contains an even number of backslashes with "\\.*"
- C) This will replace any sequence of four backslashes or greater with four backslashes, and then the rest of the backslashes in the sequence, effectively changing nothing.
- D) This will replace any sequence of two backslashes with "\\.*"
- E) Because this function uses the backreference \$1, literal instances of "(\\\\\\\\)+" are replaced with .*

3. What is the minimum number of directed edges needed to create a strongly connected graph of 12 nodes?

- A) 66
- B) More information is needed
- C) 12
- D) 22
- E) 11

4. What is approximately the number of comparisons needed for the worst case of a quick sort implementation (when selecting one element at random for the pivot)?

- A) $n(n+1)/2$
- B) $n(n-1)/2$
- C) n^2
- D) $n\log(n)$
- E) $n(n+1)/2-1$

5. What is the optimal average time complexity to find the median of an unsorted array?

- A) $O(n)$
- B) $O(n\log n)$
- C) $O(n(\log(\log(n))))$
- D) $O(n^2)$
- E) $O(1)$

<p>6. What is the optimal time complexity to iterate through all rectangles in an $n \times n$ grid</p> <p>A) $O(n^2)$ B) $O(n^3)$ C) $O(n^3 \log(n))$ D) $O(n^4)$ E) $O(n^2 \log(n))$</p>	
<p>7. What is the Big O of the following code in terms of the length of the input string, n?</p> <p>A) $O(\log(n))$ B) $O(n)$ C) $O(n \log(n))$ D) $O(n^2)$ E) $O(1)$</p>	<pre>public static String recurse(String s){ if(s.length() <= 1) return s; return recurse(s.substring(0,s.length()/2)) + s; }</pre>
<p>8) What is the best time complexity for a program that sorts java longs?</p> <p>A) $O(n)$ B) $O(\log(n))$ C) $O(n \log(n))$ D) $O(n + \log(n))$ E) $O(n * \log(\log(n)))$</p>	
<p>9) What sorting method does <code>Arrays.sort()</code> use on arrays of Objects?</p> <p>A) Merge Sort B) Quicksort C) Insertion Sort D) Radix Sort E) Timsort</p>	
<p>10) Which of the following correctly compares Quicksort and Merge Sort? (When using simple implementations)</p> <p>A) Merge Sort is stable, Quicksort is not B) Merge Sort requires transitive values, Quicksort does not C) Merge Sort and Quicksort have the same worst case Big O complexity D) Merge Sort is used by Java for objects while Quicksort is used for primitive values E) Neither Quicksort nor Merge Sort can sort in place</p>	
<p>11) After the following code is run, what is the internal size of the array within the <code>ArrayList</code> <code>arr</code>?</p> <p>A) 21 B) 22 C) 24 D) 27 E) 31</p>	<pre>ArrayList<Integer> arr = new ArrayList<>(); for(int i = 0; i < 21; i++){ arr.add(i); }</pre>
<p>12) Convert 1010.11011 base 2 to decimal</p> <p>A) 10.8125 B) 10.685 C) 10.84375 D) 10.72525 E) 347</p>	

13) Which of the following can this boolean equation change to by applying De Morgan's law once?

$$((!A)B+C)*A$$

- A) $((!A)!(!(B)*(C))))*A$
- B) $((!A)!(!(B)+(C))))+A$
- C) $!(!(A)B)*(C))*A$
- D) $!(((A)B+C)+(A))$
- E) $!(((A+(!B)))+C)*A$

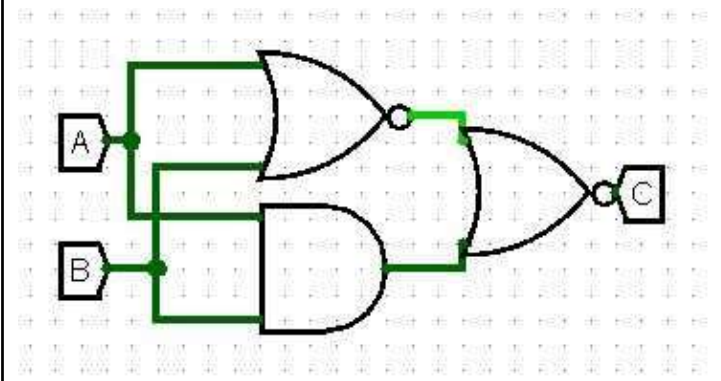
14) Which of these does the following boolean equation simplify to:

$$(A+B)*C + D*(A+B)*!C!D + (A*B)(!(A+B))*C*D + (A+B)*!C$$

- A) $A*B$
- B) $A+B$
- C) $(A+B)*!C$
- D) $!A * !B + !C +!D$
- E) $A \oplus B$

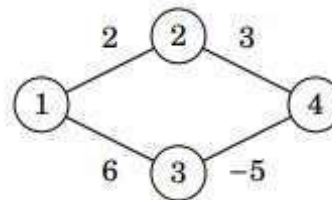
15) What is the best interpretation of the following circuit?

- A) $A*B$
- B) $A+B$
- C) $(A+B)*!C$
- D) $!A * !B + !C +!D$
- E) $A \oplus B$



16) Which of the following algorithms is best (Fastest and correct) to find the shortest paths from node 1 to all other nodes in the following graph?

- A) Dijkstra
- B) DFS
- C) BFS
- D) Bellman-Ford
- E) Floyd-Warshall



17) What is the minimum number of characters that need to be changed, added or removed to make this program compile?

- A) 3
- B) 5
- C) 6
- D) 7
- E) 8

```
public Class Test {
    public static int main(String[] ars){
        System.out.println(Hello World);
    }
}
```

18) Which of the following statements would round the double d to 2 decimal places and ensure that the length of the double is at least 10 characters long by appending the double with zeros?

- A) `printf("%010.2f",d)`
- B) `printf("%08.2f",d)`
- C) `printf("%07.2f",d)`
- D) `printf("%.2f010",d)`
- E) `printf("%0.2f010",d)`

19) What part of Java specifically allows for ArrayLists to accept multiple data types?

- A) Polymorphism
- B) Encapsulation
- C) Inheritance
- D) Generics
- E) Abstraction

20) Which of the following is equal to the number of ways to choose k elements from a set of size n?

- A) $n(n+1)/2k(k+1)$
- B) n choose k+1
- C) n choose k-1
- D) $(n-1)!/((n-k)!(k+1)!)$
- E) n choose n-k

21) What input to the following function generates the output: H5dMhThsaipEsitxeTc. This is a short answer question.

```
public static String recurse(String s){
    if(s.length()<=5) {
        String ret = "";
        for(int i = s.length()-1; i>=0; i--){
            ret+=s.charAt(i);
        }
        return ret;
    }
    return recurse(s.substring(0,s.length()/2)) +
        recurse(s.substring(s.length()/2));
}
```

22) What input to the following function generates the output: YNS!HR!HUR!NVO!HOWDSRD. This is a short answer question.

```
public static String foo(String s){
    String ret = "";
    for(char c : s.toCharArray()){
        ret+=(char) (c^1);
    }
    return ret;
}
```

```

public class LargeProblem {
    public static void main(String[] args){
        Foo obj = new Foo(4);
        obj.insert(1);
        obj.insert(1);
        obj.insert(2);
        obj.insert(3);
        obj.insert(5);
        obj.insert(4);
        obj.insert(-3);
        for(int i = 0; i < 3; i++){
            System.out.println(obj.remove());
        }
    }
}

class Foo {
    private int[] arr;
    private int i;
    private boolean red = false;
    public Foo(int a){
        arr = new int[1<<a];
        arr[0] = Integer.MAX_VALUE;
        i = 1;
    }
    public boolean insert(int a){
        if (i > arr.length) {
            return false; //Block B
        }
        arr[i] = a;
        int ind = i++;
        while(arr[ind] > arr[ind>>1]){
            arr[ind] = arr[ind>>1] ^ arr[ind]; //Block A
            arr[ind>>1] = arr[ind>>1] ^ arr[ind]; //Block A
            arr[ind] = arr[ind>>1] ^ arr[ind]; //Block A
            ind>>=1;
        }
        return true;
    }
    public int remove(){
        int ind = 1;
        int ret = arr[ind];
        boolean swapped = false;
        while(!swapped) {
            int l = ind << 1 >= i ? Integer.MIN_VALUE : arr[ind << 1]; //Block C
            int r = (ind << 1) + 1 >= i ? Integer.MIN_VALUE : arr[(ind << 1)+1]; //Block C
            if (l > r){
                if (arr[i-1] >= 1){
                    swapped = true;
                    arr[ind] = arr[i-1];
                }
            }
        }
    }
}

```

```

        else
            arr[ind] = 1;
            ind<<=1;
        } else {
            if (arr[i-1] >= r){
                swapped = true;
                arr[ind] = arr[i-1];
            }
            else
                arr[ind] = r;
            ind<<=1;
            ind++;
        }
    }
    --i;
    return ret;
}
}

```

The code above applies to problems 23 - 28

23) Which of the following is the output of the above code?

A)	B)	C)	D)	E)
-3	5	5	-3	-3
1	4	3	1	4
2	3	4	1	5

24) Which of the following is implemented by the class Foo?

- A) Max Heap
- B) Binary search tree
- C) Min Heap
- D) Red-Black Tree
- E) Priority Queue

25) What are the Big O values for the insert and remove methods in Foo respectively, with regard to the variable i?

- A) $O(i \cdot \log(i))$, $O(i)$
- B) $O(\log(i))$, $O(i)$
- C) $O(i)$, $O(\log(i))$
- D) $O(i \cdot \log(i))$, $O(i \cdot \log(i))$
- E) $O(\log(i))$, $O(\log(i))$

26) What is the function of the three lines marked by Block A?

- A) Swap arr[ind] and arr[ind>>1] in place
- B) Calculate the xor of arr[ind] and arr[ind>>1] and place them in their respective locations
- C) Perform a transfer operation between the pointers at ind>>1 and ind
- D) Perform a Gaussian-swap
- E) invert arr[ind] and arr[ind>>1]

27) Which of the following blocks of code can replace block b to expand the capacity of the data structure

- A)
- ```
int[] temp = new int[arr.length>>1];
i = 0;
for(int a : arr){
 temp[++i] = a;
}
arr = temp;
```
- B)
- ```
int[] temp = new int[arr.length>>1];
i = 0;
for(int a : arr){
    temp[i++] = a;
}
arr = temp;
```
- C)
- ```
Arrays.resize(arr,arr.length>>1)
```
- D)
- ```
this = new Foo(arr.length>>1);
```
- E)
- ```
int[] temp = new int[arr.length>>1];
for(int i = 0; i < arr.length; i++){
 temp[i] = arr[i];
}
arr = temp;
```

28) Which of the following can replace Block C for the code to function properly?

- A) `int l = ind << 1 >= i ? Integer.MIN_VALUE : arr[ind << 1]; r = (ind << 1) + 1 >= i ? Integer.MIN_VALUE : arr[(ind << 1)+1];`
- B) `int l = ind << 1 >= i ? Integer.MAX_VALUE : arr[ind << 1]; r = (ind << 1) + 1 >= i ? Integer.MAX_VALUE : arr[(ind << 1)+1];`
- C) `int l = (ind << 1) + 1 >= i ? arr[(ind << 1)+1] : Integer.MIN_VALUE; r = ind << 1 >= i ? arr[ind << 1] : Integer.MIN_VALUE;`
- D) `int l = ind << 1 >= i ? arr[ind << 1] : Integer.MAX_VALUE; r = (ind << 1) + 1 >= i ? arr[(ind << 1)+1] : Integer.MAX_VALUE;`
- E) `int l = (ind << 1) + 1 >= i ? arr[(ind << 1)+1] : Integer.MAX_VALUE; r = ind << 1 >= i ? arr[ind << 1] : Integer.MAX_VALUE;`

|                                                                                                                                                                                                                                                                                                                                                                          |                                                                                                                                                                                                                                                                                                                                    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |                                                                                                                                                                                                                                                                    |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|----|----|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>29) What input to the following function generates the output: fkjJKLQPas. This is a short answer question.</p>                                                                                                                                                                                                                                                       | <pre>public static String foo(String s){     String str = "";     for(char c: s.toCharArray()){         if (c &gt;= 97 &amp;&amp; c &lt; 97+26)             str+=(char)(c-32);         else if (c &gt;=65 &amp;&amp; c &lt; 65+26){             str+=(char)(c+32);         }         else return "";     }     return str; }</pre> |    |    |    |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |                                                                                                                                                                                                                                                                    |
| <p>30) Which of the following is not equivalent to <math>A \wedge B</math>?</p> <p>A) <math>A*(!B) + B*(!A)</math><br/>B) <math>(A  B) \&amp;\&amp; !(A\&amp;\&amp;B)</math><br/>C) <math>!(!(A+B) + (A*B))</math><br/>D) <math>(A + B + (A*B) + A^B)) * !(A*B)</math><br/>E) <math>!(!(A*B) + (A\&amp;B))</math></p>                                                    |                                                                                                                                                                                                                                                                                                                                    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |                                                                                                                                                                                                                                                                    |
| <p>31) What is the output of the following code?</p> <table><tr><td>A)</td><td>B)</td><td>C)</td><td>D)</td></tr><tr><td>1</td><td>2</td><td>1</td><td>1</td></tr><tr><td>4</td><td>4</td><td>5</td><td>6</td></tr><tr><td>5</td><td>5</td><td>4</td><td>4</td></tr><tr><td>4</td><td>4</td><td>1</td><td>1</td></tr></table> <p>E) Error: IndexOutOfBoundsException</p> | A)                                                                                                                                                                                                                                                                                                                                 | B) | C) | D) | 1 | 2 | 1 | 1 | 4 | 4 | 5 | 6 | 5 | 5 | 4 | 4 | 4 | 4 | 1 | 1 | <pre>public class test {     public static void main(String[] args) {         ArrayList&lt;Integer&gt; a = new ArrayList&lt;&gt;();         a.add(1);         a.add(2);         a.add(5);         a.remove(1);         a.add(4);         a.add(1,4);     } }</pre> |
| A)                                                                                                                                                                                                                                                                                                                                                                       | B)                                                                                                                                                                                                                                                                                                                                 | C) | D) |    |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |                                                                                                                                                                                                                                                                    |
| 1                                                                                                                                                                                                                                                                                                                                                                        | 2                                                                                                                                                                                                                                                                                                                                  | 1  | 1  |    |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |                                                                                                                                                                                                                                                                    |
| 4                                                                                                                                                                                                                                                                                                                                                                        | 4                                                                                                                                                                                                                                                                                                                                  | 5  | 6  |    |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |                                                                                                                                                                                                                                                                    |
| 5                                                                                                                                                                                                                                                                                                                                                                        | 5                                                                                                                                                                                                                                                                                                                                  | 4  | 4  |    |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |                                                                                                                                                                                                                                                                    |
| 4                                                                                                                                                                                                                                                                                                                                                                        | 4                                                                                                                                                                                                                                                                                                                                  | 1  | 1  |    |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |                                                                                                                                                                                                                                                                    |
| <p>32) Which of the following <b>Best</b> describes all red-black trees?</p> <p>A) Binary Tree<br/>B) Binary Search Tree<br/>C) Balanced Binary Search Tree<br/>D) Complete Balanced Binary Search Tree<br/>E) Full Complete Balanced Binary Search Tree</p>                                                                                                             |                                                                                                                                                                                                                                                                                                                                    |    |    |    |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |                                                                                                                                                                                                                                                                    |
| <p>33) What is the output of the following code?</p> <p>A) 0<br/>B) 4<br/>C) 5<br/>D) 6<br/>E) 7</p>                                                                                                                                                                                                                                                                     | <pre>String s = "T_A__B_S_!"; System.out.println(s.split("_").length);</pre>                                                                                                                                                                                                                                                       |    |    |    |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |                                                                                                                                                                                                                                                                    |



**34)** What is the maximum number of edges that you need to add to a tree of  $n$  nodes to turn the tree into a non-tree?

- A)  $n$
- B) 1
- C)  $n-1$
- D)  $n+1$
- E) 0

**35)** What is the main reason one would use a postfix expression instead of an infix expression when writing a program to evaluate a mathematical expression?

- A) Postfix expressions allow for math to be evaluated with a simple stack structure
- B) Postfix expressions allow for math to be evaluated with a simple queue structure
- C) Postfix expressions allow for math to be evaluated with a priority queue structure
- D) Postfix expressions lack parentheses, saving on space
- E) Postfix expressions have a greater value of mathematical simplicity.

**36)** Why might one want to use a singly LinkedList to implement a stack structure instead of an ArrayList?

- A) A LinkedList supports both  $O(1)$  write and reads to a stack, while ArrayLists always take  $O(n)$  for both
- B) A LinkedList supports  $O(1)$  reads to any value within the entire stack, while ArrayLists do not.
- C) A LinkedList supports  $O(1)$  writes to any value within the entire stack, while ArrayLists do not.
- D) A LinkedList supports both  $O(1)$  write and reads to a stack, while ArrayLists occasionally will take  $O(n)$  to write.
- E) A LinkedList will not run out of space while an ArrayList will.

**37)** If one were to try and divide an integer  $i$  by 2 in Java, should they use  $i/2$  or  $i>>1$  and why?

- A) It doesn't matter, as the java compiler automatically converts  $i/2$  and  $i>>1$  to the same expression in the byte code.
- B)  $i>>1$ , as this bitshift operation requires fewer clock cycles than the division instruction.
- C)  $i/2$ , as this division operation is greater optimised for division, as bitshifts can both divide and multiply.
- D)  $i>>1$  as shifting by  $n-1$  is a faster way to divide any integer  $i$  by  $n$ .
- E)  $i/2$  as the division in Java is turing complete, meaning that it runs faster amortized.

**38)** Evaluate the following postfix expression:

1 2 + 3 - 1 + 5 - 4 +

- A) -1
- B) 0
- C) 1
- D) 2
- E) 5

**39)** Describe the distribution of `(int) (Math.random()*Math.random()*100)`

- A) The maximum value is 100, the minimum 0, values are more likely to be less than 50 than greater than 50
- B) The maximum value is 99, the minimum 0, values are equally distributed
- C) The maximum value is 100, the minimum 1, values are more likely to be greater than 50 than less than 50
- D) The maximum value is 99, the minimum 0, values are more likely to be greater than 50 than less than 50.
- E) The maximum value is 99, the minimum 0, values are more likely to be less than 50 than greater than 50.

**40)** Which of the following is the best answer for why one would use dynamic programming?

- A) One uses dynamic programming to store calculated values so that they don't have to be recalculated, saving on time at the cost of storage.
- B) One uses dynamic programming to slow down programs by calculating more, saving greatly on storage.
- C) One uses dynamic programming to optimize recursive functions by storing their values for retrieval later, speeding up the program at a cost of storage.
- D) One uses dynamic programming to change the program itself as it runs, enabling for the program to take up less storage as it runs, optimising memory usage and increasing program speed.
- E) One uses dynamic programming for error handling, allowing for a program to dynamically react to extraneous input in an enterprise environment.