THE VIRTUAL MEET EXPERIENCE

2023-2024

HS VIRTUAL CHALLENGE MEET #6 - THE SQ MEET

COMPUTER SCIENCE

DO NOT OPEN TEST UNTIL TOLD TO DO SO

The Virtual Challenge Meets™

2023-2024 High School Virtual Challenge Meet #6 The SQ Meet

General Directions:

1) DO NOT OPEN EXAM UNTIL TOLD TO DO SO.
2) **NO CALCULATORS of any kind may be used.**
3) You have 45 minutes to complete this contest. If you are in the process of actually writing an answer when the signal to stop is given, you may finish writing that answer.
4) Papers may not be turned in until forty-five minutes have elapsed. If you finish the test before the end of the allotted time, remain at your seat and retain your paper until told to do otherwise. You may use this time to check your answers.
5) All answers must be written on the answer sheet/Scantron card provided. Indicate your answers in the appropriate blanks provided on the answer sheet or on the Scantron card. Clean erasures are necessary for accurate Scantron grading.
6) You may place as many notations as you desire anywhere on the test paper except on the answer sheet or Scantron card which is reserved for answers only.
7) You may use additional scratch paper provided by the contest director.
8) All questions have ONE and only ONE correct (BEST) answer. There is a penalty for all incorrect answers. **All provided code segments are intended to be syntactically correct, unless otherwise stated (i.e. `error` is an answer choice). Ignore any typographical errors and assume any undefined variables are defined as used.**
9) A reference to commonly used Java classes is provided with the test and you may use this reference during the contest. You may detach the reference sheets from the test booklet but DO NOT DO SO UNTIL THE CONTEST BEGINS.
10) Assume that any necessary import statements for **Standard Java 20 Packages** and classes (e.g. .lang, .util, System, Math, Double, etc.) are included in any programs or code segments that refer to methods from these classes and/or packages.

Scoring:

1) All questions will receive 6 points if answered correctly; no points will be given or subtracted if unanswered; 2 points will be deducted for each incorrect answer.

For Computer Science practice tests and hands-on materials, go to www.apluscompsci.com

Standard Classes and Interfaces — Supplemental Reference

class java.lang.Object
- `boolean equals(Object other)`
- `String toString()`
- `int hashCode()`

interface java.lang.Comparable<T>
- `int compareTo(T other)`
  Return value < 0 if `this` is less than `other`.
  Return value = 0 if `this` is equal to `other`.
  Return value > 0 if `this` is greater than `other`.

class java.lang.Integer implements Comparable<Integer>
- `Integer(int value)`
- `int intValue()`
- `boolean equals(Object obj)`
- `String toString()`
- `int compareTo(Integer anotherInteger)`
- `static int parseInt(String s)`

class java.lang.Double implements Comparable<Double>
- `Double(double value)`
- `double doubleValue()`
- `boolean equals(Object obj)`
- `String toString()`
- `int compareTo(Double anotherDouble)`
- `static double parseDouble(String s)`

class java.lang.String implements Comparable<String>
- `int compareTo(String anotherString)`
- `boolean equals(Object obj)`
- `int length()`
- `String substring(int begin, int end)`
  Returns the substring starting at index `begin` and ending at index `(end - 1)`.
- `String substring(int begin)`
  Returns `substring(from, length())`.
- `int indexOf(String str)`
  Returns the index within this string of the first occurrence of `str`. Returns `-1` if `str` is not found.
- `int indexOf(String str, int fromIndex)`
  Returns the index within this string of the first occurrence of `str`, starting the search at the specified index.. Returns `-1` if `str` is not found.
- `charAt(int index)`
- `int indexOf(int ch)`
- `int indexOf(int ch, int fromIndex)`
- `String toLowerCase()`
- `String toUpperCase()`
- `String[] split(String regex)`
- `boolean matches(String regex)`

class java.lang.Character
- `static boolean isDigit(char ch)`
- `static boolean isLetter(char ch)`
- `static boolean isLetterOrDigit(char ch)`
- `static boolean isLowerCase(char ch)`
- `static boolean isUpperCase(char ch)`
- `static char toUpperCase(char ch)`
- `static char toLowerCase(char ch)`

class java.lang.Math
- `static int abs(int a)`
- `static double abs(double a)`
- `static double pow(double base, double exponent)`
- `static double sqrt(double a)`
- `static double ceil(double a)`
- `static double floor(double a)`
- `static double min(double a, double b)`
- `static double max(double a, double b)`
- `static int min(int a, in b)`
- `static int max(int a, int b)`
- `static long round(double a)`
- `static double random()`
  Returns a `double` value with a positive sign, greater than or equal to `0.0` and less than `1.0`.

interface java.util.List<E>
- `boolean add(E e)`
- `int size()`
- `Iterator<E> iterator()`
- `ListIterator<E> listIterator()`
- `E get(int index)`
- `E set(int index, E e)`
  Replaces the element at `index` with the object `e`.
- `void add(int index, E e)`
  Inserts the object `e` at position `index`, sliding elements at position `index` and higher to the right (adds 1 to their indices) and adjusts `size`.
- `E remove(int index)`
  Removes element from position `index`, sliding elements at position `(index + 1)` and higher to the left (subtracts 1 from their indices) and adjusts `size`.

class java.util.ArrayList<E> implements List<E>

class java.util.LinkedList<E> implements List<E>, Queue<E>

Methods in addition to the `List` methods:
- `void addFirst(E e)`
- `void addLast(E e)`
- `E getFirst()`
- `E getLast()`
- `E removeFirst()`
- `E removeLast()`

class java.util.Stack<E>
- `boolean isEmpty()`
- `E peek()`
- `E pop()`
- `E push(E item)`

interface java.util.Queue<E>
- `boolean add(E e)`
- `boolean isEmpty()`
- `E peek()`
- `E remove()`

class java.util.PriorityQueue<E>
- `boolean add(E e)`
- `boolean isEmpty()`
- `E peek()`
- `E remove()`

interface java.util.Set<E>
- `boolean add(E e)`
- `boolean contains(Object obj)`
- `boolean remove(Object obj)`
- `int size()`
- `Iterator<E> iterator()`
- `boolean addAll(Collection<? extends E> c)`
- `boolean removeAll(Collection<?> c)`
- `boolean retainAll(Collection<?> c)`

class java.util.HashSet<E> implements Set<E>

class java.util.TreeSet<E> implements Set<E>

interface java.util.Map<K,V>
- `Object put(K key, V value)`
- `V get(Object key)`
- `boolean containsKey(Object key)`
- `int size()`
- `Set<K> keySet()`
- `Set<Map.Entry<K, V>> entrySet()`

class java.util.HashMap<K,V> implements Map<K,V>

class java.util.TreeMap<K,V> implements Map<K,V>

interface java.util.Map.Entry<K,V>
- `K getKey()`
- `V getValue()`
- `V setValue(V value)`

interface java.util.Iterator<E>
- `boolean hasNext()`
- `E next()`
- `void remove()`

interface java.util.ListIterator<E> extends java.util.Iterator<E>

Methods in addition to the `Iterator` methods:
- `void add(E e)`
- `void set(E e)`

class java.lang.Exception
- `Exception()`
- `Exception(String message)`

class java.util.Scanner
- `Scanner(InputStream source)`
- `boolean hasNext()`
- `boolean hasNextInt()`
- `boolean hasNextDouble()`
- `String next()`
- `int nextInt()`
- `double nextDouble()`
- `String nextLine()`
- `Scanner useDelimiter(String pattern)`

2023-2024 Virtual Challenge Meet #6 – The SQ Meet - Computer Science

Note: Correct responses are based on **Java SE Development Kit 20 (JDK 20)**, from Sun Microsystems, Inc. All provided code segments are intended to be syntactically correct, unless otherwise stated (i. e. `error` is an answer choice) and any necessary Java SE 20 Standard Packages have been imported. Ignore any typographical errors and assume any undefined variables are defined as used. **For all output statements, assume that the System class has been statically imported using:** ***import static java.lang.System.*;***

QUESTION 1

Which of these is **NOT** equivalent to $204_8 + 100010_2$?

A. 10101110_2 B. 166_{10} C. 246_8 D. $A6_{16}$ E. more than one

QUESTION 2

What is output by the code to the right?

A. 26 B. 852
C. 18 D. 21
E. 15

```
out.print( 8 + 5 * 2 );
```

QUESTION 3

What is output by the code to the right?

A. \ B. \\
C. \\\\ D. 0
E. There is no output due to a syntax error.

```
out.print( '\\\\' );
```

QUESTION 4

What is output by the code to the right?

A. _com B. co
C. com D. i_com
E. m

```
String s = "www_apluscompsci_com";
out.println(s.substring(17,20));
```

QUESTION 5

What is output by the code to the right?

A. true B. false

```
boolean a = false;
a = false || false ^ true^ true ;
out.println( a );
```

QUESTION 6

What is output by the code to the right?

A. 81 B. 81.0
C. 12 D. 12.0
E. There is no output due to a syntax error.

```
float w = (float)Math.pow(3,4);
out.println( w );
```

QUESTION 7

What is output by the code to the right?

A. 243.0 B. 36.0
C. 27.0 D. 12.0
E. There is no output due to a syntax error.

```
double x = 9;
double b = 3;
out.println( x * b * x );
```

QUESTION 8

What is output by the code to the right?

A. `45` B. `345`

C. `34` D. `34`

E. `35`

```
int x = 0b1010;
if( x%2 != 0 )
   System.out.print("3");
else
   System.out.print("4");
System.out.println( 5 );
```

QUESTION 9

What is output by the code to the right?

A. `2` B. `4`

C. `21` D. `11`

E. The output cannot be determined due to an infinite loop.

```
int x = 0;
int y = 50;
while( x < y ){
   x = x + 8;
   y -= x;
}
out.println(y);
```

QUESTION 10

What is output by the code to the right?

A. `51` B. `92`

C. `17` D. `12`

E. There is no output due to a syntax error.

```
int []a = {-2,12,51,17,92,33};
out.println( a[a.length/2] );
```

QUESTION 11

What is output by the code to the right?

A. `www` B. `aplus`

C. `comp` D. `sci`

E. There is no output due to a syntax error.

```
String s = "www aplus comp sci";
Scanner aplus = new Scanner(s);
aplus.next();
out.println( aplus.next() );
```

QUESTION 12

What is output by the code to the right?

A. `19` B. `22`

C. `18` D. `25`

E. There is no output due to a runtime error.

```
int k = 0, z = 30;
for( int i = 0; i < 10; i+=3){
  if( i < z ){
    k += i;
  }
}
out.println( k );
```

QUESTION 13

What is output by the code to the right?

A. `7` B. `4`

C. `6` D. `8`

E. There is no output due to a runtime error.

```
int[][] m = {{1,2,3},{4},{5,6,7,8}};
out.println( m[2][2] );
```

QUESTION 14

What is output by the code to the right?

A. `2.9` B. `1.5`

C. `7.5` D. `0.3`

E. There is no output due to a syntax error.

```
TreeSet<Double> s = new TreeSet<>();
s.add( 1.5 );
s.add( 7.5 );
s.add( 0.3 );
s.add( 3.3 );
s.add( 2.9 );
out.println( s.floor( 3.0 ) );
```

QUESTION 15

What is output by the code to the right?

A. `2 16 256`

B. `2 4 16`

C. `2 4 16 256 512`

D. `2 4 16 256`

E. `2 16`

```
for (int i=2; i <= 256; i *= i)
{
  out.print(i + " ");
}
```

QUESTION 16

What is output by the code to the right?

A. `icspomcsulpa`

B. `apluscompsci`

C. `icspmocsulpa`

D. `compsciaplus`

E. There is no output due to a runtime exception.

```
String s = "apluscompsci";
String o = "";
int i;
for(i=s.length() - 1; i >= 0; --i)
{
   o += s.charAt(i);
}
out.println(o);
```

QUESTION 17

What is output by the code to the right?

A. `18` B. `9` C. `72` D. `64` E. `144`

```
out.println(36 << 2);
```

QUESTION 18

What is output by the code to the right?

A. `306`

B. `389`

C. `0`

D. `null`

E. There is no output due to a runtime exception.

```
String s = "APLUS";
int h = 0;
int r = s.length();
while( r-- > 0)
{
   h += s.charAt(r);
}
out.println(h);
```

QUESTION 19

What is output by the code to the right?

A. `0` B. `22`

C. `11` D. `33`

E. `true`

```
ArrayList<Integer> U;
U = new ArrayList<>();
U.add(11);
U.add(22);
U.add(33);
out.println(U.remove(2));
```

QUESTION 20

What is output by the code to the right?

A. `58` B. `44`

C. `8` D. `9`

E. `31`

```
String s = "www.apluscompsci.com";
s = "www.practice.apluscompsci.com";
s = "The BEST UIL Prep!!!!";
int x = s.indexOf("UIL");
out.println(x);
```

QUESTION 21

Which sort is being used in the code to the right ?

A. selection B. merge

C. insertion D. quick

E. radix

```
int[] p = {3,-5,7,999,0,-32736};
Arrays.sort(p);
```

QUESTION 22

What is output by the code to the right?

A. `truetruetruetrue`

B. `truefalsetruetrue`

C. `truetruefalsetrue`

D. `falsetruetruefalse`

E. `falsefalsetruetrue`

```
public static boolean y( boolean b )
{
 if (b || !b)
    return true;
 else if (b && !b)
    return false;
 return true;
}

////////////////////////////////////
// client code  ////////////////////
out.print( y(true) );
out.print( y(false) );
out.print( y(true || false) );
out.print( y(false && true) );
```

QUESTION 23

What is output by `//line 1` in the code to the right?

A. `tweet` B. `Chicken`

C. `Bird` D. `cluck`

E. There is no output due to a runtime error.

QUESTION 24

What is output by `//line 2` in the code to the right?

A. `tweet` B. `Chicken`

C. `Bird` D. `cluck`

E. There is no output due to a runtime error.

```
class Bird {
  public void talk() {
     out.println("tweet");
  }
}
class Chicken extends Bird {
  public void talk() {
     out.println("cluck");
  }
}

////////////////////////////////////
// client code  ////////////////////
Bird b = new Bird();
b.talk();   //line 1

b = new Chicken();
b.talk();   //line 2
```

QUESTION 25

What is output by the code to the right?

A. `110`

B. `104`

C. `106`

D. `108`

E. `102`

```
public static int funFour(int i) {
  return funFive(i + 5);
}

public static int funFive(int i) {
  if (i < 100)
     return funFour(++i);
  return i;
}

////////////////////////////////////
// client code  ////////////////////
out.println(funFour(3));
```

QUESTION 26

What is output by the code to the right?

A. `2147483648` B. `Integer.MAX_VALUE`

C. `2147483647` D. `-2147483648`

E. There is no output due to a runtime error.

```
int x = Integer.MIN_VALUE;
out.println( Math.abs(x) );
```

QUESTION 27

What is output by the code to the right?

A. `[1, 2, 3, 5, 7, 9, 10]`

B. `[1, 7, 2, 5, 9, 10, 3]`

C. `[10, 9, 7, 5, 3, 2, 1]`

D. `[1, 9, 2, 5, 7, 10, 3]`

E. There is no output due to a runtime error.

```
int[] ray = {1, 7, 2, 5, 9, 10, 3};
Arrays.sort(ray, 2, 5);
out.println(Arrays.toString(ray));
```

QUESTION 28

What is output by the code to the right?

A. `21` B. `26`

C. `32` D. `29`

E. There is no output due to a runtime error.

```
out.println( 7 + 4 << 3 - 2 ^ 12 );
```

QUESTION 29

What is output by the code to the right?

A. `2` B. `-1`

C. `0` D. `-2`

E. There is no output due to a runtime error.

```
String A = "APLUS";
String B = "COMPSCI";
int C = A.compareTo(B);
out.println(C);
```

QUESTION 30

What is output by the code to the right?

A. `[0, 4, 12, 24]`

B. `[0, 16, 6, 8]`

C. `[0, 16, 72, 192]`

D. `[4, 20, 48, 88]`

E. There is no output due to runtime error.

```
int[] ray = {2,4,6,8};
for (int i=0; i<ray.length; i++)
{
  ray[i] *= i * ray[i];
}
out.println(Arrays.toString(ray));
```

QUESTION 31

What is output by the code to the right?

A. `15` B. `21` C. `16` D. `010101` E. `12`

```
int i = 0b010101;
out.println(Integer.toString(i, 16));
```

QUESTION 32

What is output by the code to the right?

A. `105.0` B. `115.0`

C. `116.0` D. `120.0`

E. There is no output due to infinite recursion.

```
public static double z(int i) {
  if (i < 100)
     return z(i + 10);
  return i + 15;
}

////////////////////////////////
// client code ////////////////
out.println( z(1) );
```

QUESTION 33

What is output by the code to the right?

A. `22` B. `55`

C. `87` D. `38`

E. `-1`

```
out.println( '0' - 10 );
```

QUESTION 34

What is output by the code to the right?

A. `Howdy`

B. `2`

C. `Chicken`

D. `Doody`

E. There is no output due to a runtime error.

```
Stack<String> s = new Stack<>();
s.push("Howdy");
s.push("Doody");
s.peek();
s.push("2");
s.push("Chicken");
s.pop();
out.println(s.pop());
```

QUESTION 35

What is output by the code to the right?

A. `A+212!!!`

B. `+++212!!!`

C. `A+A+A+212!!!`

D. `A212`

E. There is no output due to an infinite loop.

```
String s = "A+A+CSA+BEST212!!!";
s = s.replaceAll("[A-Z]{2}","");
out.println(s);
```

QUESTION 36

What is output by the code to the right?

A. `3`

B. `null`

C. `8`

D. `!`

E. There is no output due to a runtime error.

```
HashMap<String, Integer> mp;
mp = new HashMap<>();
mp.put("H", 1);
mp.put("I", 2);
mp.put("!", 3);
mp.put("A", 5);
mp.put("!", 8);
mp.put("G", 9);
out.println(mp.get("!"));
```

QUESTION 37

What is output by the code to the right?

A. `[3, 3, 5, 7]`

B. `[3, 5, 7, 3]`

C. `[3, 7, 3, 5]`

D. `[3, 7, 5, 3]`

E. `[3, 5, 3, 7]`

```
PriorityQueue<Integer> pq;
pq = new PriorityQueue<>();
pq.add(3);
pq.add(7);
pq.add(2);
pq.poll();
pq.add(3);
pq.add(5);
pq.add(-4);
pq.poll();
out.println(pq);
```

QUESTION 38

Which sort has a worst and best case runtime of N * logN ?

QUESTION 39

If the following values are inserted into a binary search tree in the order shown, how many leaves would the tree contain?

10 30 5 90 12 45 2 900

QUESTION 40

Given the tree created in question 39, how many levels would the tree contain?

Name _____

School _____

Division: 1A 2A 3A 4A 5A 6A Grade: 9 10 11 12

Final Score _____ _____

2nd Grading _____ _____

1st Grading _____ _____
score initials

The Virtual Challenge Meets
Computer Science - Student Answer Sheet

1) _____
2) _____
3) _____
4) _____
5) _____
6) _____
7) _____
8) _____
9) _____
10) _____

11) _____
12) _____
13) _____
14) _____
15) _____
16) _____
17) _____
18) _____
19) _____
20) _____

21) _____
22) _____
23) _____
24) _____
25) _____
26) _____
27) _____
28) _____
29) _____
30) _____

31) _____
32) _____
33) _____
34) _____
35) _____
36) _____
37) _____
38) _____
39) _____
40) _____

Total Number Correct _____ x 6 = _____

Total Number Incorrect _____ x -2 = _____

Final Score (record here and top right) _____

2023-2024
Virtual Challenge Meet #6 - The SQ Meet
Computer Science - KEY

1) A
2) C
3) E
4) C
5) B
6) B
7) A
8) A
9) A
10) C

11) B
12) C
13) A
14) A
15) D
16) C
17) E
18) B
19) D
20) D

21) D
22) A
23) A
24) D
25) B
26) D
27) B
28) B
29) D
30) C

31) A
32) C
33) D
34) B
35) C
36) C
37) E
38) merge sort or heap sort
39) 4
40) 4

Note to Graders:

- All provided code segments are intended to be syntactically correct, unless otherwise stated (e.g. error is an answer). **Ignore any typographical errors.**
- Any necessary Standard Java 20 Packages are assumed to have been imported as needed. ssume any undefined (undeclared) variables have been defined as used.