UIL COMPUTER SCIENCE WRITTEN TEST

2026 INVITATIONAL B

FEBRUARY 2026

General Directions (Please read carefully!)

1. DO NOT OPEN THE EXAM UNTIL TOLD TO DO SO.
2. There are 40 questions on this contest exam. You will have 45 minutes to complete this contest.
3. All answers must be legibly written on the answer sheet provided. Indicate your answers in the appropriate blanks provided on the answer sheet. Clean erasures are necessary for accurate grading.
4. You may write on the test packet or any additional scratch paper provided by the contest director, but NOT on the answer sheet, which is reserved for answers only.
5. All questions have ONE and only ONE correct answer. There is a 2-point penalty for all incorrect answers.
6. Tests may not be turned in until 45 minutes have elapsed. If you finish the test before the end of the allotted time, remain at your seat and retain your test until told to do otherwise. You may use this time to check your answers.
7. If you are in the process of actually writing an answer when the signal to stop is given, you may finish writing that answer.
8. All provided code segments are intended to be syntactically correct, unless otherwise stated. You may also assume that any undefined variables are defined as used.
9. A reference to many commonly used Java classes is provided with the test, and you may use this reference sheet during the contest. AFTER THE CONTEST BEGINS, you may detach the reference sheet from the test booklet if you wish.
10. Assume that any necessary import statements for standard Java SE packages and classes (e.g., `java.util`, `System`, etc.) are included in any programs or code segments that refer to methods from these classes and packages.
11. NO CALCULATORS of any kind may be used during this contest.

Scoring

1. Correct answers will receive **6 points**.
2. Incorrect answers will lose **2 points**.
3. Unanswered questions will neither receive nor lose any points.
4. In the event of a tie, the student with the highest percentage of attempted questions correct shall win the tie.

STANDARD CLASSES AND INTERFACES – SUPPLEMENTAL REFERENCE

package java.lang

```
class Object
  boolean equals(Object anotherObject)
  String toString()
  int hashCode()

interface Comparable<T>
  int compareTo(T anotherObject)
```
Returns a value < 0 if `this` is less than `anotherObject`.
Returns a value = 0 if `this` is equal to `anotherObject`.
Returns a value > 0 if `this` is greater than `anotherObject`.

```
class Integer implements Comparable<Integer>
  Integer(int value)
  int intValue()
  boolean equals(Object anotherObject)
  String toString()
  String toString(int i, int radix)
  int compareTo(Integer anotherInteger)
  static int parseInt(String s)

class Double implements Comparable<Double>
  Double(double value)
  double doubleValue()
  boolean equals(Object anotherObject)
  String toString()
  int compareTo(Double anotherDouble)
  static double parseDouble(String s)

class String implements Comparable<String>
  int compareTo(String anotherString)
  boolean equals(Object anotherObject)
  int length()
  String substring(int begin)
```
Returns `substring(begin, length())`.
```
  String substring(int begin, int end)
```
Returns the substring from index `begin` through index `(end - 1)`.
```
  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 `fromIndex`. Returns `-1` if `str` is not found.
```
  int indexOf(int ch)
  int indexOf(int ch, int fromIndex)
  char charAt(int index)
  String toLowerCase()
  String toUpperCase()
  String[] split(String regex)
  boolean matches(String regex)
  String replaceAll(String regex, String str)

class 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 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, int b)
  static int max(int a, int b)
  static long round(double a)
  static double random()
```
Returns a `double` greater than or equal to `0.0` and less than `1.0`.

package java.util

```
interface List<E>
class ArrayList<E> implements List<E>
  boolean add(E item)
  int size()
  Iterator<E> iterator()
  ListIterator<E> listIterator()
  E get(int index)
  E set(int index, E item)
  void add(int index, E item)
  E remove(int index)

class LinkedList<E> implements List<E>, Queue<E>
  void addFirst(E item)
  void addLast(E item)
  E getFirst()
  E getLast()
  E removeFirst()
  E removeLast()

class Stack<E>
  boolean isEmpty()
  E peek()
  E pop()
  E push(E item)

interface Queue<E>
class PriorityQueue<E>
  boolean add(E item)
  boolean isEmpty()
  E peek()
  E remove()

interface Set<E>
class HashSet<E> implements Set<E>
class TreeSet<E> implements Set<E>
  boolean add(E item)
  boolean contains(Object item)
  boolean remove(Object item)
  int size()
  Iterator<E> iterator()
  boolean addAll(Collection<? extends E> c)
  boolean removeAll(Collection<?> c)
  boolean retainAll(Collection<?> c)

interface Map<K,V>
class HashMap<K,V> implements Map<K,V>
class TreeMap<K,V> implements 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()

interface Iterator<E>
  boolean hasNext()
  E next()
  void remove()

interface ListIterator<E> extends Iterator<E>
  void add(E item)
  void set(E item)

class Scanner
  Scanner(InputStream source)
  Scanner(String str)
  boolean hasNext()
  boolean hasNextInt()
  boolean hasNextDouble()
  String next()
  int nextInt()
  double nextDouble()
  String nextLine()
  Scanner useDelimiter(String regex)
```

Standard Classes and Interfaces – Supplemental Reference

Package java.util.function

Interface BiConsumer<T,U>
```
  void accept(T t, U u)
```

Interface BiFunction<T,U,R>
```
  R apply(T t, U u)
```

Interface BiPredicate<T,U>
```
  boolean test(T t, U u)
```

Interface Consumer<T>
```
  void accept(T t)
```

Interface Function<T,R>
```
  R apply(T t)
```

Interface Predicate<T>
```
  boolean test(T t)
```

Interface Supplier<T>
```
  T get()
```

UIL COMPUTER SCIENCE WRITTEN TEST – 2026 INVITATIONAL B

Note: Correct responses are based on **Java SE Development Kit 22 (JDK 22)** from Oracle, Inc. All provided code segments are intended to be syntactically correct, unless otherwise stated (e.g., "error" is an answer choice) and any necessary Java SE 22 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

Find a number M in base 10 such that when it is written in base 7 and then interpreted as base 5, it results in 345_{10}. Write your answer in the blank provided on your answer sheet for this question.

Question 2

What is output by the code to the right?

A) 15.5
B) 15
C) 15.0
D) NaN
E) Infinity

```
int a = 5;
double b = 2.5;
int c = 4;
double result = a++ + b * --c / (a % 3)
  + +c - -b;
out.println(result);
```

Question 3

What is output by the code to the right?

A) `"2\n2.5\"` **B)** `"2\n2.5\\"`
C) `"2\n2.5\\""` **D)** `"2\n2.5\\"\"`
E) Compile-time error

```
out.print("\"");
out.printf("%d", 10/4);
out.print("\\n");
out.printf("%.1f", 10/4.0);
out.print("\\\\");
out.printf("%s", "\"");
```

Question 4

What is output by the code to the right?

A) `truetruetruetrue`
B) `falsetruetruefalse`
C) `truetruetruefalse`
D) `falsetruefalsefalse`
E) `falsefalsefalsefalse`

```
String s = "Comp Sci Rocks!";
String s2 = new String("Comp Sci Rocks!");
String s3 = new String("Comp Sci Rocks!").intern();

out.print(s == s2);
out.print(s == s3);
out.print(s2 == s3);
out.print(s.equals(s2)^s2.equals(s3));
```

Question 5

What is output by the code to the right?

A) `true false` **B)** `true true`
C) `false false` **D)** `false true`
E) There is no output due to an error.

```
boolean a = false;
boolean b = true;
boolean c = false;
boolean result = a || b && (c = true) ^
  (c = false);
out.println(result + " " + c);
```

Question 6

What is output by the code to the right?

A) `2147483648|5`
B) `-2147483648|-2`
C) `-2147483648|5`
D) `2147483647|5`

```
int m = Integer.MIN_VALUE;
out.println(Math.abs(m) + "|" +
Math.floorMod(m, 7));
```

Question 7

What is output by the code to the right?

A) `5.0` **B)** `6.0` **C)** `7.0` **D)** `8.0` **E)** `6`

```
int a = 4, b = 7, c = 3;
double r = a++ * --b / (double)(c += a) +
  (a % 3) - (b-- - ++c);
out.println(r);
```

Question 8

What is output by the code to the right?

A) `AC4`

B) `AD4`

C) `BC4`

D) `BD4`

E) `AE4`

```
int x = 1;
String s = "";

switch (x++) {
  case 1:
    if (++x == 3) s += "A";
    else s += "B";
  case 2:
    if (x++ == 3) s += "C";
    else s += "D";
    break;
  default:
    s += "E";
}

out.println(s + x);
```

Question 9

What is output by the code to the right?

A) `021|` **B)** `0201|`

C) `0210|` **D)** `021|10`

E) `0211`

```
int i = 0, j = 3;
while (i < j) {
    out.print(i++ + "" + --j);
    if (i == j) out.print("|");
}
```

Question 10

What is output by the code to the right?

A) `30202` **B)** `30002` **C)** `30022` **D)** `20002`

```
int[] a = {2, 0, 2, 0, 2};
int[] b = a;
int i = 0;

b[a[i]++] = a[++i] + a[i--];

out.println(a[0] + "" + a[1] + "" +
a[2] + "" + a[3] + "" + a[4]);
```

Question 11

Assume a file exists at path **`dir/nums.txt`** and it contains exactly: `7 010 8 09`

What is output by the code on the next page?

A) `dir|nums.txt|30`

B) `dir|nums.txt|31`

C) `dir|nums.txt|32`

D) `dir|nums.txt|33`

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

Question 12

What is output by the code to the right?

A) `12|54`

B) `11|54`

C) `12|45`

D) `13|54`

E) `12|60`

```
int n = 508;
int sum = 0, prod = 1;

while (n > 0) {
  int d = n % 10;
  sum += d;

  if (sum % 2 == 0) prod *= (d + 1);
  else              prod += n % 7;

  n /= 10;
  if (d == 0) sum--;
}
out.println(sum + "|" + prod);
```

```
/* Code for Q11 */

File f = new File("dir" + File.separator + "nums.txt");
Scanner sc = new Scanner(f);

sc.useRadix(8);

int sum = sc.nextInt() + sc.nextInt() + sc.nextInt(10);

int last;
if (sc.hasNextInt()) last = sc.nextInt();
else last = sc.nextInt(10);

sum += last;

out.println(f.getParent() + "|" + f.getName() + "|" + sum);
sc.close();
```

Question 13

What is output by the code to the right?

A) 27

B) 29

C) 31

D) 39

E) Compile-time error

```
int x = 1, y = 2, z = 3;
boolean p = false;

out.println((x + y << z) ^ ((x | y & z)
    + (p || x++ == 1 && --y > 0
    ? 4 : 8)));
```

Question 14

Which statement is **always true** in Java for any int x?

A) `~x == -x`

B) `~x == -x - 1`

C) `~x == x ^ 1`

D) `~x == x + 1`

E) `~x == x - 1`

Question 15

What is output by the code to the right?

A) 27

B) 29

C) 31

D) 39

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

```
ArrayList<Integer> a = new
ArrayList<>();
a.add(0); a.add(1); a.add(2);
a.add(1); a.add(0);

List<Integer> v = a.subList(1, 4);

out.print(v.remove(1));
a.remove((Integer)1);
out.print(v.size());
```

Question 16

What is output by the code to the right?

A) 23

B) 13

C) 123

D) There is no output due to a compile error.

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

```
try {
    String s = "apple";
    char c = s.charAt(5);
    out.print(1);
} catch (ArrayIndexOutOfBoundsException e) {
    out.print(2);
} finally {
    out.print(3);
}
```

Question 17

What could replace **<1*>** in the code to the right so that it will compile and run without error.

A) pop **B)** remove **C)** poll

D) Both A and B. **E)** All of the above.

Question 18

Assuming **<1*>** is filled in correctly, which of the following could not be found in the output by the code to the right?

A) Johnny **B)** Test

C) Appleseed **D)** Quick

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

Question 19

What is the big $\mathcal{O}$ runtime of the code to the right

A) $\mathcal{O}(N \log \mathrm{N})$ **B)** $\mathcal{O}(N)$

C) $\mathcal{O}(N^2)$ **D)** $\mathcal{O}(log\, N)$

E) $\mathcal{O}(1)$

```
Stack<String> st = new Stack<String>();

st.add("Johnny");
st.add("Cash");
st.add("Test");
st.<1*>();
st.<1*>();
st.add("Appleseed");
st.add("Quick");
st.<1*>();

for(int i = 0; i < 12; i++)
     st.add(st.<1*>());

out.println(st.<1*>());
```

Question 20

Which of the lines in the code to the right first causes an error?

A) //1 **B)** //2

C) //3 **D)** //4

E) None of the above. All lines will be error-free.

```
String s = "Banana";
int i = s.charAt(0);    //1
s.substring(2);         //2
s.charAt(s.length());   //3
i = (int)(s);           //4
```

Question 21

What is output if the following call is made to the function `mystery` to the right? `mystery(10,6);`

A) 31 **B)** 33

C) 39 **D)** 27

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

Question 22

What is output if the following call is made to the function `mystery` to the right? `mystery(25,17);`

A) 35 **B)** 21

C) 25 **D)** 39

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

```
int mystery(int a, int b) {
  if(a == b)
    return 1;
  if(a < b)
    return 2 * mystery(b, a);
  if(b < 0)
    return 0;
  return 3 + mystery(a - b, b - 1);
}
```

Question 23

What can replace **<1*>** in the code to the right so that the `iPhone` class compiles and runs without error?

A) `implements IOS, Phone`

B) `extends IOS, Phone`

C) `extends IOS implements Phone`

D) `implements Phone extends IOS`

E) More than one of the above.

Question 24

Which of the following methods must be initialized by the `iPhone` class to the right?

A) `addApp(String a)` **B)** `numApps()`

C) iPhone constructor **D)** `call()`

E) None are required. **F)** A, B, and D.

G) C and D. **H)** All are required.

Question 25

What can replace **<2*>** in the code to the right so that the `iPhone` class compiles and runs without error?

A) `protected` **B)** `public`

C) Nothing is required **D)** `private`

E) A or B. **F)** A, B, or C.

G) Any of the above.

Question 26

What can replace **<3*>** in the code to the right so that the `iPhone` class compiles and runs without error?

A) `protected` **B)** `public`

C) Nothing is required **D)** `private`

E) A or B. **F)** A, B, or C.

G) Any of the above.

Question 27

Assuming **<1*>**, **<2*>**, and **<3*>** are filled in correctly, what is output by the code to the right?

A) `8328328321 4`

B) `null 5`

C) `8328328321 5`

D) `null 4`

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

```
interface Phone {
    String call();
}

abstract class IOS {
    String number;
    ArrayList<String> apps;

    public IOS(String n) {
        number = n;
        apps = new ArrayList<>();
    }

    abstract void addApp(String a);
    abstract int numApps();
}

class iPhone <1*> {
    public iPhone(String n) {
        super(n);
    }

    <2*> void addApp(String a) {
        apps.add(a);
    }

    <2*> int numApps() {
        return apps.size();
    }

    <3*> String call() {
        return number;
    }
}

////////////client code///////////
iPhone i = new
iPhone("8328328321");
i.addApp("Facetime");
i.addApp("Notes");
i.addApp("Spotify");
i.addApp("Netflix");
i.addApp("Notes");
String n = i.call() + " ";
out.print(n + i.numApps());
```

Question 28

What is output by the code to the right?

A) `false false` **B)** `false true`

C) `true false` **D)** `true true`

E) There is no output due to a compile error.

```
String str = "Back Up Down Forwards";
String reg =
    "([A-Z][a-z]*)|((\\W|\\D)+)";
out.print(str.matches(reg) + " ");
str = "NobodyToldMe212";
out.println(str.matches(reg));
```

Question 29

Which of the following is not a legal Java instantiation, i.e. causes no compile errors (possible runtime errors should be ignored)?

A) `Collection<BigInteger> c = new Stack<BigInteger>();`

B) `List l = new LinkedList<LinkedList>();`

C) `Queue<LinkedList<String>> q = new PriorityQueue<LinkedList<String>>();`

D) `LinkedHashMap<TreeMap, Object> lhm = new LinkedHashMap<>();`

E) More than one of the above. **F)** None of the above.

Question 30

If a selection sort algorithm takes 16 seconds to sort a list of 4,000 items, how long will it take to sort a list of 2,000 items (assuming average-case for both)?

A) 4 seconds **B)** 8 seconds **C)** 2 seconds **D)** 5 seconds **E)** 10 seconds

Question 31

Which of the following changes to the program to the right will cause the program to run without error, disregarding whatever output it may produce, and is the most accurate answer choice?

A) Do nothing. The code will run without error as is.

B) Remove the underscores from `L1`.

C) Change `L2` to `System.out.println(5);`.

D) Wrap `L2` in a `static` block (i.e., `static { … }`) instead of just the non-static block it is currently contained within.

E) Remove `L3`.

F) More than one of the answer choices above are valid and will cause the program to run without error.

G) None of the answer choices above are accurate, and the program will still error.

```
import static java.lang.System.*;

public class Q_31_32_33 {
    public static final int MOD =
        1_000_000_009; // L1

    {
        out.println(5); // L2
    }

    public static void main(String[] args) {
        out.println(MOD); // L3
    }
}
```

Question 32

Suppose no changes are made to the program to the right. What is output by the code to the right?

A)
```
5
```
B)
```
1000000009
```
C)
```
5
1000000009
```
D)
```
1000000009
5
```
E) No output is produced, but the program runs without error.

Question 33

Suppose the changes described by answer choice D to question 31 are made to the program to the right. What is output by the now modified code to the right?

A)
```
5
```
B)
```
1000000009
```
C)
```
5
1000000009
```
D)
```
1000000009
5
```
E) No output is produced, but the program runs without error.

Question 34

What is order of precedence of the operators to the right?

```
I.   ++ (post)
II.  -- (pre)
III. -> (lambda)
IV.  :: (method reference)
```

A) `II, I, IV, III` **B)** `I, II. IV, III`

C) `III, II, I, IV` **D)** `III, I, II, IV`

E) `IV, II, I, III` **F)** `IV, I, II, III`

Question 35

What is output by the code to the right?

A) `10 8 10 3.14 10.0 3.141592653589793`

B) `10 8 10 3.14 10.0 3.141592653589793101`

C) `10 8 8 3.14 3.1 3.141592653589793`

D) `10 8 8 3.14 3.1 3.141592653589793101`

E) There is no output due to an error.

```
out.printf("%d %d %<d %.2f %<.1f %s%n",
    7 + 3,
    (int) Math.pow(2, 3),
    3.14159,
    Math.PI,
    "Hello".charAt(1)
);
```

Question 36

Consider the randomized algorithm to the right, which takes as input a set of unique vertices V of size n and outputs a graph. Alice claims that this algorithm is guaranteed to generate a tree on the vertex set V. Bob argues that the algorithm is not guaranteed to generate a tree. Each student has provided two formal proofs supporting their claim (options A and B by Alice; options C and D by Bob). Which of the following proofs is the most accurate?

A) *Direct proof*: Since the algorithm adds exactly $n - 1$ edges and removes exactly one vertex at each iteration, the resulting graph must be a tree. Therefore, the algorithm always generates a tree.

B) *Proof by contradiction*: Assume for contradiction that the algorithm produces a graph that is not a tree. Since the graph is undirected and has exactly $n - 1$ edges, this can only occur if the graph contains a cycle. Assume for contradiction that a cycle is created when adding some edge (u, v). Then u and v must already be connected by a path using previously added edges. However, in this algorithm, after every edge insertion one of its endpoints is removed from V and never used again, so each connected component is always represented by exactly one remaining vertex in V. Therefore, two distinct vertices chosen from V can never already be connected, making it impossible for the new edge to close a cycle. This is a contradiction, thus the algorithm always generates a tree.

C) *Proof by induction*: Assume for induction that the graph formed after k iterations of the for-loop is a tree. At iteration $k + 1$, the algorithm adds an edge between two random vertices and removes one vertex. Since random choices may create cycles, the resulting graph is not guaranteed to remain a tree. Therefore, by induction, the algorithm does not guarantee a tree.

D) *Proof by counter-example*: Consider $V = \{a, b, c, d\}$. Suppose the algorithm performs the following sequence of choices: (i) Picks edge (a, b); removes vertex d. (ii) Picks edge (b, c); removes vertex b. (iii) Picks edge (c, a); removes vertex a. The resulting graph contains the cycle $a \to b \to c \to a$, which cannot exist in a tree, and therefore, the algorithm is not guaranteed to generate a tree.

```
private static final Random RAND =
    new Random();

private <T extends Comparable<T>>
    HashMap<T, HashSet<T>>
    generateTree(HashSet<T> V, int n)
{
    HashMap<T, HashSet<T>> graph =
        new HashMap<T, HashSet<T>>();

    for (int i = 0; i < n - 1; i++) {
        int idx1 = RAND.nextInt(V.size());
        int idx2;
        do {
            idx2 = RAND.nextInt(V.size());
        } while (idx1 == idx2);

        T u = null, v = null;
        int count = 0;
        Iterator<T> it = V.iterator();
        while (it.hasNext()) {
            T val = it.next();
            if (count == idx1) {
                u = val;
            } else if (count == idx2) {
                v = val;
            }
            count++;
        }

        if (!graph.containsKey(u)) {
            graph.put(u, new HashSet<T>());
        }
        graph.get(u).add(v);

        if (!graph.containsKey(v)) {
            graph.put(v, new HashSet<T>());
        }
        graph.get(v).add(u);

        if (RAND.nextBoolean()) {
            V.remove(v);
        } else {
            V.remove(u);
        }
    }
    return graph;
}
```

Question 37

Alice additionally wrote the following proof but decided to scrap it because she thought there was an issue with it. Please identify the issue, if any…

Proof by contradiction: Assume for contradiction that the algorithm produces a graph that is not a tree. Since the graph is undirected and has $n - 1$ edges, the only way this can occur is if the graph contains a cycle. Without loss of generality, let the cycle be $u \rightarrow v \rightarrow w \rightarrow \cdots \rightarrow u$, with length at least three. For vertex u to be in a cycle, it must have been selected as a candidate vertex in at least two different iterations. However, after an edge is formed between u and another vertex, either u or that vertex is removed from the candidate set. Thus, u cannot appear in two different edges that are a part of a cycle. This contradicts the assumption that a cycle exists. Therefore, the algorithm must generate a tree.

A) Since the algorithm is not guaranteed to generate a tree, her proof is wrong as she is arguing the wrong thing.

B) There are no issues – the proof is perfectly valid as is.

C) A cycle existing in the graph is not necessary and/or sufficient for $n - 1$ undirected edges to be unable to form a tree, thus her initial assumption is wrong.

D) Her reasoning that "u cannot appear in two different edges that are a part of a cycle" is false. Since this fact is used as the reason there is a contradiction, her proof is invalid.

E) Assuming "without loss of generality" that the graph takes the form of "$u \rightarrow v \rightarrow w \rightarrow \cdots \rightarrow u$, with length at least three" is an unrealistic assumption and makes her proof too general, potentially leaving out important counter-examples.

Question 38

Pictured to the right is a Deterministic Finite Automata (DFA) over the language of bit strings (i.e., alphabet $\Sigma = \{0,1\}$). What is the regular expression corresponding to the set of all bit strings that can be generated by said DFA?

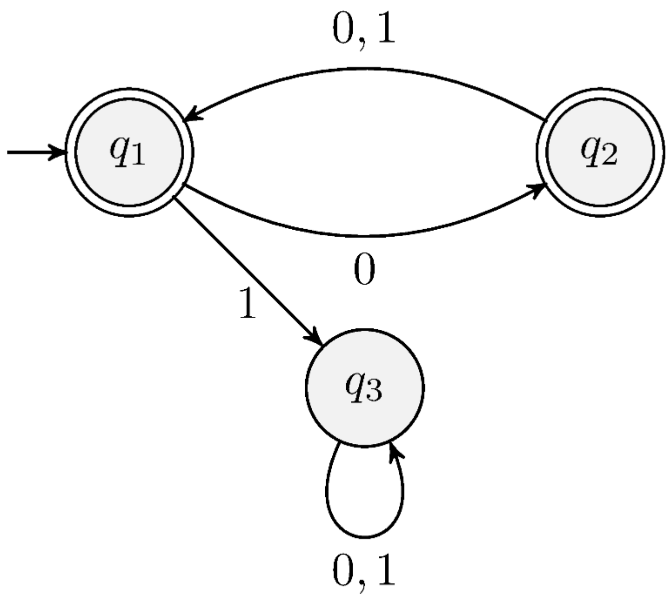
A) `(0(0+1))*`

B) `((0(0+1))*)+((0(0+1))*1(0+1)*)`

C) `((0(0+1))*)+((0(0+1))*0)`

D) `(0(0+1))*1(0+1)*`

E) More than one of the above.



Question 39

What is the 8-bit two's complement representation of the following base 10 number? Write your answer in the blank provided on your answer sheet for this question.

`-67`

Question 40

How many 0s are present in the truth table created from the following boolean expression? Write your answer in the blank provided on your answer sheet for this question.

```
(A && !B || E) && (C ^ !D) && (!E || !C && A) || (B ^ D ^ C)
```