1. What Does This Program Do? (BASIC)

When the following program is run, what is the final value of A?

```
10 A = 0
20 FOR J = 1 TO 10 STEP 3
30   FOR K = J TO 10 STEP 4
40     A = A + J * K
50   NEXT K
60 NEXT J
```

2. What Does This Program Do? (BASIC)

Variable x is a 5x5 array, initially set to all zeros. How many elements of the array will contain a non-zero after the following program is run?

```
for r = 2 to 4
  for c = 2 to 4
    if x(r-1,c-1)=0 then x(r-1,c-1)=1 else x(r-1,c-1)=0
    if x(r,  c-1)=0 then x(r,  c-1)=1 else x(r,  c-1)=0
    if x(r+1,c+1)=0 then x(r+1,c+1)=1 else x(r+1,c+1)=0
    if x(r,  c+1)=0 then x(r,  c+1)=1 else x(r,  c+1)=0
  next c
next r
```

3. Computer Number Systems

Solve for X:

$$X_8 = 3247_8 + 6435_8$$

4. Computer Number Systems

Which number is the largest? Express your answer as a letter, **a**, **b**, **c**, etc. If there's a tie, list all.

(**a**) 126_{10} (**b**) 1111111_2 (**c**) 175_8 (**d**) 81_{16}

5. Bit-String Flicking

Evaluate the following:

```
01011 OR 10110 AND 11010
```

6. Bit-String Flicking

Evaluate the following expression:

```
(LCIRC-3 (RSHIFT-2 (LCIRC-1 01101))) AND
            (LSHIFT-1 (RCIRC-2 (NOT 11100)))
```

7. Bit-String Flicking

Find all values of `x` (5-bits long) that make the following equation true.

```
x OR (LSHIFT-2 11011)) = (RCIRC-2 11011)
```

8. Boolean Algebra

Simplify the following expression as much as possible:

$$(\overline{A}B + AB)(\overline{AB})$$

9. Boolean Algebra

How many ordered triples make the following expression true?

$$AB(\overline{A} + C) + \overline{A}(B + C)$$

10. Boolean Algebra

Complete the label of the rightmost column in the following truth table.

A	B	C	A ☐ $\overline{B}$	$\overline{A}$ ☐ C	$(A$ ☐ $\overline{B})$ ☐ $(\overline{A}$ ☐ $C)$
0	0	0	0	1	1
0	0	1	0	1	1
0	1	0	0	1	1
0	1	1	0	1	1
1	0	0	1	0	1
1	0	1	1	1	1
1	1	0	0	0	0
1	1	1	0	1	1

$(A$ ☐ $\overline{B})$ ☐ $(\overline{A}$ ☐ $C)$

1. The following table shows the values the variables take when the program is run.

J	K	A
1	1	1
1	5	6
1	9	15
4	4	31
4	8	63
7	7	112
10	10	212

212

2. Each time through the loop, the values of four elements of the array are toggled. The following table shows the elements of x that change from 0 to 1, and from 1 to 0, each time through the inner loop:

r	c	0→1	1→0
2	2	(1,1) (2,1) (3,3) (2,3)	
2	3	(1,2) (2,2) (3,4) (2,4)	
2	4	(1,3) (3,5) (2,5)	(2,3)
3	2	(3,1) (4,3)	(2,1) (3,3)
3	3	(3,2) (4,4)	(2,2) (3,4)
3	4	(2,3) (3,3) (4,5)	(3,5)
4	2	(4,1) (5,3)	(3,1) (4,3)
4	3	(4,2) (5,4)	(3,2) (4,4)
4	4	(4,3) (5,5)	(3,3) (4,5)

12

3. Work from the right to the left, carrying as needed. All numbers are in octal, except where noted.

$$7 + 5 = 12_{10} = 14$$
$$carry + 4 + 3 = 8_{10} = 10$$
$$carry + 2 + 4 = 7$$
$$3 + 6 = 9_{10} = 11$$

11704

4. Compare the numbers by converting each of them into a common base. Since it's easy to convert among base 2, 8, and 16, the obvious approach to this problem is to choose one of those, and then convert the decimal number into that base. However, because these numbers are pretty small, converting them all into base 10 is easy. The values of the numbers are as follows:

(**b**) $1111111_2 = (10000000 - 1)_2 = (128 - 1) = 127$

(**c**) $175_8 = 1 \cdot 8^2 + 7 \cdot 8 + 5 = 64 + 56 + 5 = 125$

(**d**) $81_{16} = 8 \cdot 16 + 1 = 129$

(**d**)

5. The AND must be evaluated before the OR; the evaluation is as follows:

`01011 OR 10110 AND 11010` = `01011 OR 10010`

= `11011`

`11011`

6. Simplify the left operand of the AND

`(LCIRC-3 (RSHIFT-2 11010))` = `(LCIRC-3 00110)` = `10001`

and now the right operand:

`(LSHIFT-1 (RCIRC-2 00011))` = `(LSHIFT-1 11000)` = `10000`

Finally, `10001 AND 10000 = 10000`.

`10000`

7. Simplify the equation by evaluating the LSHIFT and RCIRC:

`x OR 01100 = 11110`

Now, rewrite `x` as `abcde` and consider each bit in turn.

`a OR 0` = 1 ⇒ `a` = 1
`b OR 1` = 1 ⇒ `b` = *anything*
`c OR 1` = 1 ⇒ `c` = *anything*
`d OR 0` = 1 ⇒ `d` = 1
`e OR 0` = 0 ⇒ `e` = 0

`10010, 10110, 11010, 11110`

Can be written as `1**10`.

8. The simplification is as follows:

$$\begin{aligned}(\overline{A}B + AB)(\overline{AB}) &= B(\overline{A} + A)(\overline{A} + \overline{B}) \\ &= B(\overline{A} + \overline{B}) \\ &= B\overline{A} + B\overline{B} \\ &= B\overline{A}\end{aligned}$$

The operands of the AND may be commuted. Thus, the following answer is also correct: $B\overline{A}$

$\overline{A}B$

9. The expression simplifies as follows:

$$\begin{aligned}AB(\overline{A} + C) + \overline{A}(B + C) &= AB\overline{A} + ABC \ + \ \overline{A}B + \overline{A}C \\ &= ABC + \overline{A}B + \overline{A}C\end{aligned}$$

The following truth table shows the inputs that make the expression true:

A	B	C	ABC	$\overline{A}B$	$\overline{A}C$	$ABC + \overline{A}B + \overline{A}C$
0	0	0	0	0	0	0
0	0	1	0	0	1	1
0	1	0	0	1	0	1
0	1	1	0	1	1	1
1	0	0	0	0	0	0
1	0	1	0	0	0	0
1	1	0	0	0	0	0
1	1	1	1	0	0	1

4

10. In the fourth column, the expression $A\boxed{\ \ }\overline{B}$ is true only when $A = 1$ and $B = 0$, so the operator must be an AND. In the fifth column, the expression $\overline{A}\boxed{\ \ }C$ is true except when $A = 1$ and $C = 0$, so the operator must be an OR. The rightmost column is the OR of the previous two columns.

$(A\boxed{*}\overline{B})\boxed{+}(\overline{A}\boxed{+}C)$