1. What Does This Program Do? (BASIC)

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

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

2. What Does This Program Do? (Pascal)

If input to the following program are the numbers 4, 5, and 6, what is printed when the program is run?

```
program int3 (input, output);
  var a,b,c: integer;
  procedure one (var a:integer; b,c: integer);
    begin a:=b+c; b:=c+2; c:=a+b; end;
  procedure two (var a,b: integer; c: integer);
    begin a:=b+c; b:=c+2; c:=a+b; one(a,b,c); end;
  procedure three (var a,b,c: integer);
    begin a:=b+c; b:=c+2; c:=a+b; two(a,b,c); end;
  begin {mainline}
  read (a,b,c);
  three (a,b,c);
  writeln('a=',a, 'b=',b, 'c=',c);
  end.
```

a= ☐ b= ☐ c= ☐

3. What Does This Program Do? (Pascal)

If input to the following program are the numbers 12 and 10, what is printed when the program is run?

```
program sr3 (input, output);
  var a,b,c: integer;
  procedure two(var a,b: integer);
    begin a := a-2; b := b-3; end;
  function one (a,b: integer): integer;
    begin
    if a>5 then
      one := one(a-1,b)+b
    else if (a<=5) and (a>0) then
      begin two(a,b); one := one(a-1, b-1)+5; end
    else if (a<=0) then
      one := 4;
    end;
  begin {mainline}
  read(a,b);
  writeln(one(a,b));
  end.
```

1996–97

American Computer Science League

Contest #3

Classroom Division — PROBLEMS

4. LISP

Find all of the following expressions that have a value 10?

- **(a)** `(set 'a (mult 5 2))`
- **(b)** `(set 'a '(mult 5 2))`
- **(c)** `(setq a (mult 5 2))`
- **(d)** `(setq a '(mult 5 2))`
- **(e)** `(mult 5 2)`

5. LISP

Evaluate the following expression:

```
(ADD 1 (MULT 2 (SUB 3 (DIV 3 2)) 1))
```

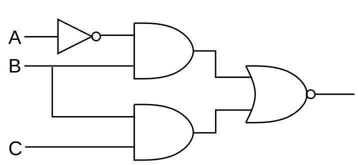
6. LISP

Evaluate the following expression:

```
(CDR (CDR (CAR (CDR '(2 ((3 (4 5) 6) 7))))))
```

7. Digital Electronics

List all of following expressions that are exact translations of the circuit at the right.



- **(a)** $\overline{\overline{A}B + BC}$
- **(b)** $(A + \overline{B})(\overline{B} + \overline{C})$
- **(c)** $\overline{(\overline{A} + B)(B + C)}$
- **(d)** $\overline{CB + \overline{A}B}$
- **(e)** $A\overline{B} + \overline{BC}$

1996–97 # American Computer Science League **Contest #3**

Classroom Division — PROBLEMS

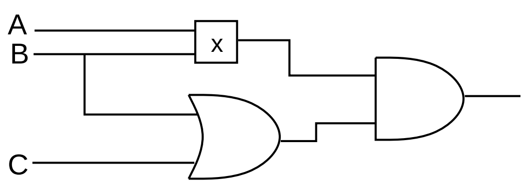
8. Digital Electronics

Draw a digital circuit that is equivalent to the following expression:

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

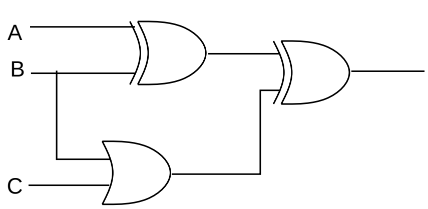
9. Digital Electronics

Find all gates (AND, OR, NAND, NOR, XOR, or XNOR) that can replace the "x" in the circuit below so that (1,1,0) and (1,1,1) are the only tuples that makes the circuit true.



10. Digital Electronics

Find all ordered triples that make the following circuit true.



1. The following table shows which elements of `x` are set to 1 each time through the inner loop:

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

Note that `x(3,3)` appears twice.

17

2. The following table shows how the values of `a`, `b`, and `c` change:

	a	b	c
before call to three	4	5	6
in three, just before call to two	11	8	19
in two, just before call to one	27	21	48
in one, just before exit	69	50	119
in two, just before exit	69	21	48
in three, just before exit	69	21	19
in mainline, just before call to writeln	69	21	19

a= 69 b= 21 c= 19

3. The evaluation of this recursive program is as follows:

$$\texttt{one}(12,10) = \texttt{one}(11,10) + 10$$
$$\texttt{one}(11,10) = \texttt{one}(10,10) + 10$$
$$\texttt{one}(10,10) = \texttt{one}(9,10) + 10$$
$$\texttt{one}(9,10) = \texttt{one}(8,10) + 10$$
$$\texttt{one}(8,10) = \texttt{one}(7,10) + 10$$
$$\texttt{one}(7,10) = \texttt{one}(6,10) + 10$$
$$\texttt{one}(6,10) = \texttt{one}(5,10) + 10$$
$$\texttt{one}(5,10) = \texttt{two}(5,10) => \texttt{a} = 3, \texttt{b} = 7 => \texttt{one}(2,6) + 5$$
$$\texttt{one}(2,6) = \texttt{two}(2,6) => \texttt{a} = 0, \texttt{b} = 4 => \texttt{one}(-1,3) + 5$$
$$\texttt{one}(-1,3) = 4$$

84

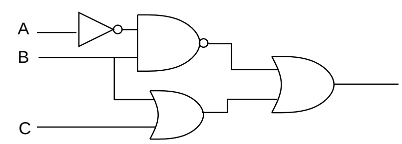
4. Choices (**a**) and (**c**) are identical; they have a value of 10. Choices (**b**) and (**d**) are also identical; they have a value of the list `(mult 5 2)`. Choice (**e**) is straightforward; it has a value of 10.	(**a**), (**c**) *and* (**e**)

5. Evaluate the expressions by working from the inside to the outside:
`(ADD 1 (MULT 2 (SUB 3 (DIV 3 2)) 1))`
`(ADD 1 (MULT 2 (SUB 3 1.5) 1))`
`(ADD 1 (MULT 2 1.5 1))`
`(ADD 1 3)`
`4`
	4

6. Work from the inside to the outside:
`(CDR '(2 ((3 (4 5) 6) 7)))  =  ((3 (4 5) 6) 7)`
`(CAR '((3 (4 5) 6) 7))  =  (3 (4 5) 6)`
`(CDR '(3 (4 5) 6))  =  ((4 5) 6)`
`(CDR '((4 5) 6))  =  (6)`
The parentheses are needed; the result of `CDR` is a list. In this case, a list with one element.
	`(6)`

7. The circuit translates to $(\overline{A}\text{ AND }B)\text{ NOR }(B\text{ AND }C)$ which can be written as $\overline{\overline{A}B + BC}$ (choice (**a**)). Choice (**d**) is the same, just with some operators commuted. Choice (**b**) is derived from choice (**a**) by applying DeMorgan's Law; however, this is not a translation of the circuit. And choices (**c**) and (**e**) are like (**a**) and (**d**), but with the AND or OR operators swapped.	(**a**) *and* (**d**)

8. What's important about the drawing of the circuit is that there is one NOT gate, one NAND gate, and two OR gates. The input to the NOT gate is A; the inputs to the NAND gate are B and the output of the NOT gate; the inputs to one of the OR gates are B and C; and the inputs to the other OR gate are the outputs of the NAND gate and other OR gate.



9. The circuit can be represented as $(A?B)(B+C)$. For this to be true, both terms of the AND must be true. When $B+C$ is true, we have 6 possible solutions: (*,0,1), (*,1,0), and (*,1,1). If $A?B$ is true only for inputs (1,1,0) and (1,1,1), the missing gate must be an AND. The following truth table shows the output of the missing gate when it is replaced by an AND, OR, NOR, NAND, XOR, and XNOR:

A	B	C	AND	OR	NOR	NAND	XOR	XNOR
0	0	1	0	0	1	1	0	1
1	0	1	0	1	0	1	1	0
0	1	0	0	1	0	1	1	0
1	1	0	1	1	0	0	0	1
0	1	1	0	1	0	1	1	0
1	1	1	1	1	0	0	0	1

AND

10. The following truth table shows the triples that make the expression true.

A	B	C	$A \oplus B$	$B+C$	$(A \oplus B) \oplus (B+C)$
0	0	0	0	0	0
0	0	1	0	1	1
0	1	0	1	1	0
0	1	1	1	1	0
1	0	0	1	0	1
1	0	1	1	1	0
1	1	0	0	1	1
1	1	1	0	1	1

(0,0,1), (1,0,0), (1,1,0), and (1,1,1)