

1. 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=

2. LISP

Evaluate the following expression:

```
(MULT (DIV 10 (ADD 2 3)) (SQUARE 3))
```

3. 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)

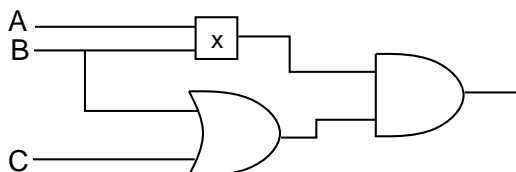
4. Digital Electronics

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

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

5. 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.



Intermediate Division — SOLUTIONS

1. 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= b= c=

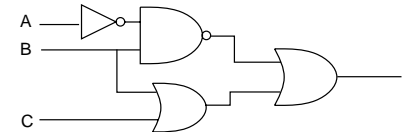
2. Evaluate the expression from the inside to the outside. First, evaluate the ADD, next the DIV, then the SQUARE, and finally the MULT. The value of the DIV is 2; the value of the SQUARE is 9.

18

3. 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)

4. 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.



5. 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