

**1. What Does This Program Do? (Pascal)**

How different numbers are in the array x after the following program is run?

```

program intermediates(input,output);
var
  r, c: integer;
  x: array [1..11][1..11] of integer;
begin
  for r:=1 to 11 do
    for c:= 1 to 11 do
      x[r][c] := 1;
    for r:= 2 to 5 do begin
      for c:= 3 to 6 do begin
        x[r,c] := x[r+1,c+5]+2;
        x[r+5,c+5] := x[r-1,c+1]+3;
      end;
    end;
  end.

```

**2. Prefix/Infix/Postfix Notation**

Convert the following expression into prefix:

$$\frac{a * b^2}{c + 1} - \frac{a^2 + b}{a * b}$$

**3. Prefix/Infix/Postfix Notation**

Suppose that @ is a binary operator whose value is the larger of its operands. Evaluate the following prefix expression.

+ @ 2 3 @ 4 @ 2 5

**4. Data Structures**

Build a binary search tree with the letters S A L T L A K E C I T Y, starting with the S and ending with the Y. What is the internal path length of the tree?

**5. Data Structures**

Consider the following sequence of operations on an empty stack:

```

push(a)
push(c)
push(s)
push(1)
pop
push(u)
pop
pop
push(s)
push(a)
pop
pop

```

If the next operation were a pop, what would be popped off the stack?

1. The final array is:

```

1 1 1 1 1 1 1 1 1 1
1 1 3 3 3 3 1 1 1 1
1 1 3 3 3 3 1 1 1 1
1 1 3 3 3 3 1 1 1 1
1 1 3 3 3 3 1 1 1 1
1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 4 4 4
1 1 1 1 1 1 1 6 6 6
1 1 1 1 1 1 1 6 6 6
1 1 1 1 1 1 1 6 6 6
1 1 1 1 1 1 1 1 1 1

```

4

2. The expression converts to  $-\boxed{x}\boxed{y}$ ,  
 where  $\boxed{x}$  and  $\boxed{y}$  are the operands of the minus.

$\boxed{x}$  converts to  $/\boxed{p}\boxed{q}$ ,

where  $\boxed{p}$  is  $* a \uparrow b 2$  and  $\boxed{q}$  is  $+ c 1$ .

$\boxed{y}$  converts to  $/\boxed{s}\boxed{t}$ ,

where  $\boxed{s}$  is  $+ \uparrow a 2 b$  and  $\boxed{t}$  is  $* a b$ .

$- / * a \uparrow b 2 + c 1 / + \uparrow a 2 b * a b$

3. The expression evaluates as follows (an expression is boxed when it is about to be evaluated):

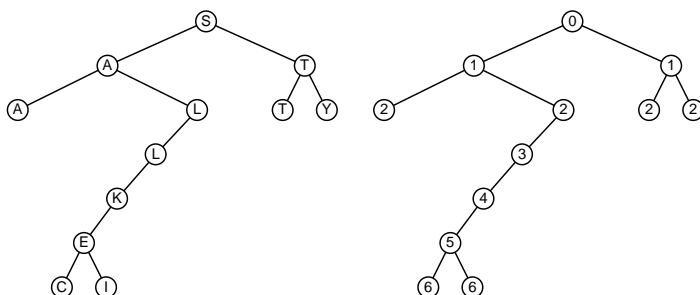
```

+ @ 2 3 @ 4 @ 2 5 ⇒ + 3 @ 4 5
      + 3 @ 4 5 ⇒ + 3 5
            + 3 5 ⇒ 8

```

8

4. The resulting tree in on the left; the tree on the right shows the amount that each node contributes to the internal path length.



34

5. Remember that a stack is “last-in, first-out.” That is, the last item pushed will be the first removed. The items popped are L, U, S, A, and S in this order. The next item to be popped will be C.

C