What Does This Program Do?

Frequently one must use or modify sections of another programmer's code. Since the original author is often unavailable to explain his/her code, it is essential to be able to read and understand an arbitrary program. This category has been rewritten based on the pseudocode used for the AP CS Principles course. Pseudo-code is English-like, language independent algorithmic code and should be able to be traced by students regardless of whether they are familiar with BASIC, Pascal, C++, Java, Python, or any other high level language. We will use the following constructs in writing this code for this topic in ACSL.

Construct	**Code segment**
Operators	! (not) , ABS (absolute value), SQR(square root), INT(integer division), ^ and ↑ (exponent), *, / (real division), % (modulus), +, -, >, <, >=, <=, !=, ==, && (and), || (or) in that order of precedence
Variables	Start with a letter, only letters and digits
Sequential statements	INPUT variable Variable = expression (assignment) OUTPUT variable

Decision statements	IF boolean expression THEN
	Statement(s)
	ELSE (optional)
	Statement(s)
	End if

Indefinite Loop statements	WHILE Boolean expression Statement(s) END WHILE
Definite Loop statements	FOR variable = start TO end STEP increment Statement(s) NEXT

Arrays: 1D arrays use a single subscript such as A(5). 2D arrays use row major order starting with (0,0) in the upper left corner.	Use ( ) for identifying the subscript(s) so that A(5) is the 6th item in a list and A(2,3) is row 2 (third row) and column 3 (4th column). The size of the array will be specified in the problem statement.
Strings: They can contain 0 or more characters and the indexed position starts with 0 as the first character. An empty string has a length of 0. Errors occur if accessing a character that is in a negative position or greater than the length of the string. The len[A] function will find the length of the string which is the total number of characters.	Strings are identified with surrounding double quotes. Use [] for identifying the characters in a substring of a given string as follows: If S = "ACSL WDTPD", then S[:3] = "ACS" S[5:] = "WDTPD" S[2:6] = "SL WD" S[0] = "A"

The questions in this topic will cover any of the above constructs in the Intermediate and Senior Division. In the Junior Division, loops will not be included in Contest 1, arrays will be included in only Contests 3 and 4.

Sample Problems

After the following program is executed, what is the final value of *B* if the input values are 50 and 10?

```
INPUT H, R
B = 0
IF H>48
    B = B + (H-48)*2*R
    H=48
ELSE
   IF H>40
        B = B +(H-40)*(3/2)*R
        H=40
   END IF
END IF
B = B + H*R
```

This program computes an employee's weekly salary, given the hourly rate (*R*) and the number of hours worked in the week (*H*). The employee is paid his/her hourly rate for the number of hours worked, up to 40; time and a half for the overtime hours, up to 48 hours; double for all hours after 48. The following table monitors variables *B* and *H* through the program execution:

B	*H*
0	50
40	50
40	48
160	48
160	40
560	40

Therefore, the final value of *B* is 560.

After the following program is executed, what is the final value of *X*?

```
A = "BANANAS"
X = 0 : T = ""
FOR j = len[A] TO 1 STEP –1
    T = T + A[j]
NEXT
FOR j = 1 TO len[A]
   if A[j] = T[j] then X = X+1
NEXT
```

The program first stores the reverse of *A$* into *T$*, and then counts the number of letters that are in the same position in both strings.

A	B	A	N	A	N	A	S
T	S	A	N	A	N	A	B
		*	*	*	*	*	

Those positions marked with an asterisk contribute one to the value of *X*. There are 5 such positions.

After the following program is executed, what is the final value of C(4)?

```
A(1)=12: A(2)=41: A(3)=52
A(4)=57: A(5)=77: A(6)=-100
B(1)=17: B(2)=34: B(3)=81
j=1: k=1: n=1
WHILE A(j) > 0
    WHILE B(k) <= A(j)
        C(n) = B(k)]
        n = n+1
        k = k+1
    END WHILE
  C(n) = A(j): n = n+1: j = j+1
END WHILE
```

The following table traces the variables through the execution of the program.

j	k	n	A(j)	B(k)	C(n)
1	1	1	12	17	12
2	1	2	41	17	17
2	2	3	41	34	34
2	3	4	41	81	41

Thus, the value of C(4) is 41. Note that this program merges two arrays in increasing order into one.