

## **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 builds these vital skills. In addition, the code fragments used in this category are examples of "good" BASIC programming, and can be studied by students for that purpose.

The dialect of BASIC that we will use is ACSL BASIC. The questions will emphasize elementary structured flow of control rather than the particular syntax of the dialect. Students familiar with BASIC or any other high-level language (e.g., Pascal, PL/C, C, FORTRAN), should be able to master this category with just a little additional preparation.

Topics covered in the Senior and Intermediate Divisions include (but are not limited to): the evaluation of arithmetic expressions, loops (single and nested), branching, subscripted variables, multi-dimensional arrays, and string manipulations. In the Junior Division, the specific topic to be tested is given in the title of the category.

The following system functions may be used: INT, ABS, LEFT\$, RIGHT\$, MID\$, LEN, VAL, STR\$, SQR and SGN. The functions NOT, AND and OR will also be used; they have the order of precedence listed (that is, NOT is highest and OR is the lowest).

The following statements may be used: INPUT, DATA, READ, RESTORE, PRINT, TAB, FOR/NEXT, GOTO, GOSUB, RETURN, IF-THEN, IF-THEN-ELSE, LET, REM, DIM, DEF (for user defined functions), END and STOP.

Files and matrix commands will NOT be used. Assume that our programs are immune to syntax errors, variables have enough storage (for example, one doesn't need an explicit "DIM X(100)" in a program that uses an array with 100 elements).

While most commands are functions that are the same on each system, the following frequently vary from system to system. Below we have listed these functions and the interpretation that will be used for the contests.

INT(A) returns the largest integer not greater than A.

LEFT\$(A\$,N) returns a string consisting of the first N characters of the string A\$.

RIGHT\$(A\$,N) returns a string consisting of the last N characters of the string A\$.

MID\$(A\$,A,B) returns a string consisting of the B characters beginning with the A<sup>th</sup> character of A\$. For example, if A\$="AMERICAN COMPUTER" then MID\$(A\$,7,5) would equal "AN CO".

LEN(A\$) returns an integer representing the number of characters in the string A\$. If A\$ is null (A\$="") then LEN(A\$)=0.

STR\$(A) converts the integer A into a string. Note that an extra space will exist on the left of the string. For example, if A=123 then STR\$(A)= “ 123”.

VAL(A\$) is the inverse to STR\$. This converts the string to its numeric value. The string’s contents must form a valid number.

## References

Golden, Neal. *Computer Programming in the BASIC Language*, Harcourt Brace Jovanovich (1981).

Kemeny, John G. and Thomas E. Kurtz. *True BASIC*, Addison Wesley (1985).

Kemeny, John G. and Thomas E. Kurtz. *BASIC Programming*, John Wiley & Sons (1980).

## Sample Problems

After the following program is executed, what is the final value of  $B$ ?

```

10 read H, R
15 data 50, 10
20 B = 0
25 if H<=48 then 40
30   B = B + (H-48)*2*R
35   H=48
40 rem
45 if H<=40 then 60
50   B = B +(H-40)*(3/2)*R
55   H=40
60 rem
65 B = B + H*R
70 end

```

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:

| LINE | $B$ | $H$ |
|------|-----|-----|
| 20   | 0   | 50  |
| 30   | 40  | 50  |
| 35   | 40  | 48  |
| 50   | 160 | 48  |
| 55   | 160 | 40  |
| 65   | 560 | 40  |

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

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

```

10 X = 0
20 for K = 7 to 52 step 6
30   X = X + K
40 next K

```

$K$  takes on values of 7, 13, 19, ..., 49. The loop adds these numbers together in line 30.

| 50 end  |  |     |      |      |      |      |      |   |   |     |    |    |    |   |   |   |    |    |    |   |   |   |    |    |    |   |   |   |    |    |    |
|---|--|-----|------|------|------|------|------|---|---|-----|----|----|----|---|---|---|----|----|----|---|---|---|----|----|----|---|---|---|----|----|----|
| <p>After the following program is executed, what is the final value of <math>X</math>? (Assume that the initial value of <math>T\\$</math> is the null string.)</p> <pre>10 read A\$ 20 X = 0 25 for J = len(A\$) to 1 step -1 30   T\$ = T\$ + mid\$(A\$,J,1) 35 next J 40 for J = 1 to len(A\$) 45   if mid\$(A\$,J,1) = mid\$(T\$,J,1) then X = X+1 50 next J 55 data BANANAS 60 end</pre> | <p>The program first stores the reverse of <math>A\\$</math> into <math>T\\$</math>, and then counts the number of letters that are in the same position in both strings.</p> <table><tr><td>A\$</td><td>B</td><td>A</td><td>N</td><td>A</td><td>N</td><td>A</td><td>S</td></tr><tr><td>A\$</td><td>S</td><td>A</td><td>N</td><td>A</td><td>N</td><td>A</td><td>B</td></tr><tr><td></td><td></td><td>*</td><td>*</td><td>*</td><td>*</td><td>*</td><td></td></tr></table> <p>Those positions marked with an asterisk contribute one to the value of <math>X</math>. There are 5 such positions.</p>          | A\$ | B    | A    | N    | A    | N    | A | S | A\$ | S  | A  | N  | A | N | A | B  |    |    | * | * | * | *  | *  |    |   |   |   |    |    |    |
| A\$   | B  | A   | N    | A    | N    | A    | S    |   |   |     |    |    |    |   |   |   |    |    |    |   |   |   |    |    |    |   |   |   |    |    |    |
| A\$   | S  | A   | N    | A    | N    | A    | B    |   |   |     |    |    |    |   |   |   |    |    |    |   |   |   |    |    |    |   |   |   |    |    |    |
|   |  | *   | *    | *    | *    | *    |      |   |   |     |    |    |    |   |   |   |    |    |    |   |   |   |    |    |    |   |   |   |    |    |    |
| <p>After the following program is executed, what is the final value of <math>C(4)</math>?</p> <pre>10 A(1)=12: A(2)=41: A(3)=52 15 A(4)=57: A(5)=77: A(6)=-100 20 B(1)=17: B(2)=34: B(3)=81 25 J=1: K=1: L=1 30 rem 35 if B(K) &gt; A(J) then 60 40   C(L) = B(K) 45   L = L+1 50   K = K+1 55   goto 35 60 C(L) = A(J) 65 L = L+1 70 J = J+1 75 if A(J)&gt;0 then 30 80 end</pre>            | <p>The following table traces the variables through the execution of the program.</p> <table><tr><th>J</th><th>K</th><th>L</th><th>A(J)</th><th>B(K)</th><th>C(L)</th></tr><tr><td>1</td><td>1</td><td>1</td><td>12</td><td>17</td><td>12</td></tr><tr><td>2</td><td>1</td><td>2</td><td>41</td><td>17</td><td>17</td></tr><tr><td>2</td><td>2</td><td>3</td><td>41</td><td>34</td><td>34</td></tr><tr><td>2</td><td>3</td><td>4</td><td>41</td><td>81</td><td>41</td></tr></table> <p>Thus the value of <math>C(4)</math> is 41. Note that this program merges two arrays in increasing order into one.</p> | J   | K    | L    | A(J) | B(K) | C(L) | 1 | 1 | 1   | 12 | 17 | 12 | 2 | 1 | 2 | 41 | 17 | 17 | 2 | 2 | 3 | 41 | 34 | 34 | 2 | 3 | 4 | 41 | 81 | 41 |
| J   | K  | L   | A(J) | B(K) | C(L) |      |      |   |   |     |    |    |    |   |   |   |    |    |    |   |   |   |    |    |    |   |   |   |    |    |    |
| 1   | 1  | 1   | 12   | 17   | 12   |      |      |   |   |     |    |    |    |   |   |   |    |    |    |   |   |   |    |    |    |   |   |   |    |    |    |
| 2   | 1  | 2   | 41   | 17   | 17   |      |      |   |   |     |    |    |    |   |   |   |    |    |    |   |   |   |    |    |    |   |   |   |    |    |    |
| 2   | 2  | 3   | 41   | 34   | 34   |      |      |   |   |     |    |    |    |   |   |   |    |    |    |   |   |   |    |    |    |   |   |   |    |    |    |
| 2   | 3  | 4   | 41   | 81   | 41   |      |      |   |   |     |    |    |    |   |   |   |    |    |    |   |   |   |    |    |    |   |   |   |    |    |    |