15. **05-06 C3 What Does this Program Do – 2D Arrays**

The first loop produces the following table: 6

4	1	2	6	2
0	2	4	1	3
2	1	2	4	2

The second loop adds: A(1,2) + A(1,5) + A(3,2) + A(3,5) = 1+2+1+2 = 6

19. **06-07 C3 What Does this Program Do – 2D Arrays**

The program produces the following array: 5

0	**-1**	**-2**	**0**
1	**-2**	**0**	**-4**
-2	**2**	**-2**	**-1**
-2	**-4**	**1**	**-3**

23. **07-08 C3 What Does this Program Do – 2D Arrays**

The program produces the following array: 164

1	4	9
1	16	36
1	3	81
2	4	6

S contains the sum of the elements = 164

27. **08-09 C3 What Does this Program Do – 2D Arrays**

The program produces the following arrays: 26

```
    4    5    6            3    6    9
    3    4    5            2    4    6
    2    3    4            1    2    3
         A                      B

    7    11   15
    5    8    5            ∴  3 + 8 + 15
    3    3    4                 = 26
         C
```

30. **09-10 C2 What Does this Program Do – 1D Arrays**

   This program takes each number and adds the proper factors   3
   (factors less than the number) of that number. If the sum and the
   number differ by 1, then it is printed. This occurs only for
   numbers that are powers of 2. That is 2, 16, and 32.

31. **09-10 C3 What Does this Program Do – 2D Arrays**

   The program produces the following arrays: 28

   A

3	4	5	6
5	6	7	8
7	8	9	10
9	10	11	12

   B

4	7	6	12
6	12	8	15
8	15	13	18
13	18	12	28

   B(I,J) contains the sum of the factors of A(I,J).
   12 has the most factors and they add to 28.

47. **13-14 C3 What Does this Program Do – 2D Arrays**

   This program changes entries in the table that are divisible by   6
   4, 3, 10 and 2. Then it counts the entries with a value of 1.
   The final table is:

7	1	1	3
2	11	1	9
7	5	3	1
2	1	27	1

52. **14-15 C4 What Does this Program Do – 2D Arrays**
   The first nested loop fills the 10x10 array with the numbers 1-100.   21
   The WHILE loop checks each number from 2 to 10 and changes
   every multiple of these numbers to 0 except the number itself.
   The final nested loop counts the number of non-zero numbers left.
   This algorithm is Eratosthenes' Sieve for finding the prime
   numbers. The answer: 21