

Compiled Solutions

1. 02-03 C1 What Does This Program Do - Branching

The results of the IF statements produce the following:

D = 2, A = 5, C = 2 and B remains unchanged.

$$5 + 2 + 2 + 2 = 11$$

11

2. 02-03 C2 What Does This Program Do - Looping

The table shows the values of I, J and X:

43

I	J	X
1	1	1
2	1	3
2	2	7
3	1	5
3	2	11
3	3	10
4	1	7
4	2	15
4	3	27
4	4	43

3. 02-03 C3 What Does This Program Do – 1D Arrays

J=1: 8 4 2 5 8 3 6 1

J=2: 8 5 2 5 8 3 6 1

J=3: 8 5 2 1 8 3 6 1

J=4: 8 5 2 1 0 3 6 1

J=5: 8 5 2 1 0 -1 6 1

J=6: 8 5 2 1 0 -1 -2 1

J=7: 8 5 2 1 0 -1 -2 -3

A(1) = 8

A(2) = 5

A(3) = 2

A(4) = 1

A(5) = 0

A(6) = 0

A(7) = -2

A(8) = -3

A(K+1) is twice A(K) when K = 6.

The array printed contains: 8 5 2 1 0 0 -2 -3

4. 02-03 C4 What Does This Program Do – Strings

16

I	B
0	C
2	CNNC
4	CNNCE
6	ECNNCEE
8	ECNNCEEELLECNNCEE

5. **03-04 C1 What Does this Program Do – Branching**

A = 36: B = 4: C = 5 1332
D = A / B + C = 36/4 + 5 = 14
E = (2 * D + 2) / C = (2*14 + 2)/5 = 6
E > B, B = 6
F = A / (B - 1) * C = 36/5 * 5 = 36
F/2 > INT(F/2), F = 37
G = F * B * E = 37*6*6 = 1332

6. **03-04 C2 What Does this Program Do – Loops**

When the program is executed the first nine values printed are: 86
10, 18, 24, 28, 30, 50, 66, 78 and 86.

7. **03-04 C3 What Does this Program Do – 2D Arrays**

The values are: -5

J	K	A(J,K)	X
1	1	8	8
1	2	1	7
1	3	6	13
2	1	3	10
2	2	5	5
2	3	7	-2
3	1	4	2
3	2	9	-7
3	3	2	-5

8. **03-04 C4 What Does this Program Do – Strings** Y = M

The original A has its letters compared from each end, letter 1 is compared with letter 21, 2 is compared with 20, etc. If the first numbered letter is the bigger, it is added to X. The second loop adds just those letters less than N to X= “UNTSTTOM” and Y = “M”.

9. **04-05 C1 What Does this Program Do - Branching**

The values of A, B and C as the program is executed are: A = 8, B = 24, C = 16

A	B	C
10	2	6
8	2	6
8	2	16
8	24	16
8	24	16

10. **04-05 C2 What Does this Program Do – Loops**

The value of X changes as follows:

37

I	J	X
1	3	1
2	5	2
3	7	4
4	9	7
5	11	10
6	13	14
7	15	19
8	17	24
9	19	30
10	21	37

11. **04-05 C3 What Does this Program Do – 2D Arrays**

The initialization of the array using $2*I + J$ produces even numbers in only columns 2 and 4. Therefore the first IF statement is TRUE 10 times. Of those 10, only 5 are still even when divided by 2 in the second IF statement.

15

12. **04-05 C4 What Does this Program Do – Strings**

I	A
2	E
3	AE
5	EEAE
6	EEAEE
11	EEEEAE
15	EEEEAE
16	AEEEEAE
18	AAEEEEAE
23	AAEEEEAE
24	AAEEEEAEEL

AAEEEEAEEL

13. **05-06 C1 What Does this Program Do – Branching**

The values of the variables change as follows:

21

A	B	C	D
100	10	5	2
100	12	5	2
100	12	10	2
100	12	10	20
100	12	100	20
50	12	100	20
50	12	100	21

14. 05-06 C2 What Does this Program Do – Loops

Only odd integers are considered from 11 to 99. 1
The program is looking for the numbers that are divisible by three,
that have the tens digit bigger than the units digit, that have a units digit of 3
or a tens digit of 5 and the product of the digits is greater than 21.
Only 1 number fits all the criteria: 93

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: $A(1,2) + A(1,5) + A(3,2) + A(3,5) = 1+2+1+2 = 6$

16. 05-06 C4 What Does this Program Do – Strings

After the first loop B = “ETUPMOC” TEOM
The second loop compares the characters in locations 0 and 1, 2 and 3,
and then 4 and 5. If the first character is less than the second the two
characters are concatenated to E. The final string is TEOM.

17. 06-07 C1 What Does this Program Do – Branching

A = 1 B = 2 C = 5 D = 12 A = 5, B = 6, C = 5, D = 12

If $1 + 5 > 12 / 2$ THEN A = 24
A = 24 B = 2 C = 5 D = 12
IF $12 - 2 \leq 4 * 24$ THEN B = 6
A = 24 B = 6 C = 5 D = 12
IF $24 * 5 > 6 * 12$ THEN A = 120
A = 120 B = 6 C = 5 D = 12
IF $120 > 6$ OR $5 < 12$ THEN A = 5
A = 5 B = 6 C = 5 D = 12

18. 06-07 C2 What Does this Program Do – Loops

The J loop produces values 15, 18, 21, 24, ...48. The I loop produces the odd numbers from 1 to 25. The IF statement asks which values of I divide the J values with no remainder.

40

I	# of *	I	# of *
1	12	15	3
3	12	17	0
5	3	19	0
7	2	21	2
9	4	23	0
11	1	25	0
13	1	TOTAL	40

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

20. 06-07 C4 What Does this Program Do – Strings

B compares 2 letters at a time in A and concatenates the first letter to the right if it is alphabetically before the second or to the left if it isn't.

MIGCUCOAUTAI

B = "ENISUIAEACMEEG"

C eliminates the E's and the A's and reverses the remaining string.

C = "GMCIUSIN"

D = "MIG" + "CUC" + "OAU" + "TAI"

D = "MIGCUCOAUTAI"

21. 07-08 C1 What Does this Program Do – Branching

The table contains the values of A, B, C, D and E after each line.

A=2, B=8,C=-2, D=16, E=8

line	A	B	C	D	E
1	4	3	12	2	6
2	4	3	6	2	6
3	4	3	6	4	6
4	9	3	6	4	6
5	2	4	-2	-4	8
6	2	8	-2	-4	8
7	2	8	-2	16	8

22. 07-08 C2 What Does this Program Do – Loops

This program counts the factors of I. If the number of factors is 3, then I is a perfect square with only a “prime root” other than 1 and itself as factors. There are only 4 prime number squares between 1 and 100 (4, 9, 25 and 49).

4

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

24. 07-08 C4 What Does this Program Do – Strings

The first loop compares some of the side-by-side letters and alphabetizes the letters in pairs. The result is:

2

IMSISPOSRU

The second loop eliminates any letter not between E and N. The final result is IMI.

Therefore there are only 2 unique letters in the output: I and M.

25. 08-09 C1 What Does this Program Do – Branching

The table contains the values of A, B, C and D after each line.

A=4, B=2, C=4, D=8

A	B	C	D
1	2	4	8
2	2	4	8
2	10	4	8
10	10	4	8
4	10	4	8
4	2	4	8

26. 08-09 C2 What Does this Program Do – Loops

This program counts the number of even deficient, perfect and abundant numbers from 4 to 30. Consider the sum of the proper factors of a number, i.e. factors less than the number. If the sum is less than the number, the number is deficient. If the sum equals the number, the number is perfect. If the sum is greater than the number, the number is abundant. There are 2 perfect numbers less than 30: 6 and 28.

2

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		$\therefore 3 + 8 + 15$	
	3	3	4		$= 26$	
		C				

28. 08-09 C4 What Does this Program Do – Strings

The first loop removes the E's and S's. The result is:

6

UNITDTATPACANDROCKTCNT

The second loop checks the letters in pairs and keeps the one alphabetically first. The result is: NIDAAADOCN

Therefore there are 6 unique letters in the output: N, I, D, A, O, C.

29. 09-10 C1 What Does this Program Do – Branching

The table contains the values of A, B, C and D after each line.

7

Line #	T/F	A	B	C	D
10		12	6	3	2
20	2>3 F	12	9	3	2
30	14>12 T	12	9	3	4
40	9>3^4<12 T^T	24	9	3	4
50	27<4 v13>3 FvT	24	9	3	4

$$(A + C)/B + D = (24 + 3)/9 + 4 = 27/9 + 4 = 3 + 4 = 7$$

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

This program takes each number and adds the proper factors (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.

3

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

The program produces the following arrays:

28

A				B			
3	4	5	6	4	7	6	12
5	6	7	8	6	12	8	15
7	8	9	10	8	15	13	18
9	10	11	12	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.

32. **09-10 C4 What Does this Program Do – Strings**

The loop removes some of the identical side-by-side letters and eliminates the A's and B's. The result is: SELFOTLSKETL
Therefore there are 7 unique letters in the output: E, F, O, K, L, S, T

7

33. **10-11 C1 What Does this Program Do – Branching**

The table contains the values of A, B, C, D and E after each line.

34

A	B	C	D
100	2	50	5
100	2	100	5
105	2	100	5
5	2	100	5
100	2	100	5
100	2	7	5

$$A / D + B * C = 100 / 5 + 2 * 7 = 20 + 14 = 34$$

34. **10-11 C2 What Does this Program Do – Loops**

This program counts the number of odd factors of the numbers 1 to 9.

15

I	1	2	3	3	4	5	5	6	6	7	7	8	9	9	9
J	1	1	1	3	1	1	5	1	3	1	7	1	1	3	9
S	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

35. **10-11 C3 What Does this Program Do – 2D Arrays**

The program produces the following array:

3

1	4	2	9
3	5	5	4
3	10	13	2
14	1	2	1

Each entry represents the result of the final division on the original number.
For example, $36/1=36$, $36/2=18$, $18/3=6$, $6/6=1$. So 3 of the entries are 1.

36. 10-11 C4 What Does this Program Do – Strings

The first loop compares the strings character by character and keeps the one alphabetically first. This results in the string CONNECAICNHIGA
The second loop gets rid of the A's, B's and C's resulting in the string ONNEINHIG. The question is how long is M? Answer: 9

37. 11-12 C1 What Does this Program Do – Branching

The table contains the values of A, B, C and D after each line. A=3, B=256, C=252, D=7

A	B	C	D
1	2	3	4
8	2	3	4
8	2	4	4
8	2	4	4
8	16	4	4
8	256	4	4
3	256	4	4
3	256	252	4
3	256	252	7
3	256	252	7

38. 11-12 C2 What Does this Program Do – Loops

This program finds and counts all the palindromes (numbers that read the same forwards and backwards) from 1000 to 2000 that contain 2 different pairs of numbers: 1st & 4th and 2nd & 3rd. There are 9 such numbers: 1001, 1221, 1331, 1441, 1551, 1661, 1771, 1881, 1991

39. 11-12 C3 What Does this Program Do – 2D Arrays

The first double loop sets up the initial array. 7

2	3	4	5
3	4	5	6
4	5	6	7
5	6	7	8

The next double loop searches for multiples of 4 and 3. Sets that entry to 0 and adds or subtracts 1 from the rest.

0	3	-1	0
3	-1	0	6
-1	0	6	7
0	6	7	0

The last double loop counts the entries that are unchanged from the initial values

40. **11-12 C4 What Does this Program Do – Strings**

The first loop removes the E's and N's. The result is: 2
DCLARATIOOFIDPDC
The second loop removes the letters alphabetically after N. The result is:
DCLAAIFIDDC
The third loop checks the letters in pairs and keeps the one alphabetically first. The result is: AF
Therefore there are 2 unique letters in the output: A and F.

41. **12-13 C1 What Does this Program Do – Branching**

The table contains the values of a, b, c, d and e after each line. 9

a	b	c	d	E
20	5	2	4	16
20	10	2	4	16
20	20	2	4	16
2	20	2	4	16

$$(c+d)/a-b/a+e=(2+4)/2-20/2+16=9$$

42. **12-13 C2 What Does this Program Do – Loops**

This program finds and counts the numbers from 100 to 1000 that 211
are divisible by 3 (150) and divisible by 5 (91) and then eliminates
those divisible by both (30).

43. **12-13 C3 What Does this Program Do – 1D Arrays**

This program counts the number of primes from 2 to 15
50. It uses the method called the Sieve of
Eratosthenes.

44. **12-13 C4 What Does this Program Do – Strings**

The first loop compares characters in pairs. If the first one is alphabetically 0
first, it goes into b, otherwise it goes into c. The result is:
b = "oieeedaf" and c = "swent"
The second loop compares the shorter string to the longer string character
by character. If the first one is alphabetically first, it goes into e, else
it goes into d. The result is: d = "" and e = "oieee".
Therefore the product of the lengths is $0*5 = 0$

45. **13-14 C1 What Does this Program Do – Branching**

The table contains the values of a, b, c, d and e after each line.

304

a	b	c	d	E
100	10	5	4	2
100	20	5	4	2
5	20	5	4	2
5	20	5	2	2
5	20	15	2	2

$$a + b * c - d / e = 5 + 20 * 15 - 2 / 2 = 5 + 300 - 1 = 304$$

46. **13-14 C2 What Does this Program Do – Loops**

This program counts the number of times 2, 3, 4 and 5 are factors of the first 25 counting numbers.

31

2 is a factor 12 times, 3 is a factor 8 times, 4 is a factor 6 times, and 5 is a factor 5 times. The sum is 31.

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

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

6

The final table is:

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

48. **13-14 C4 What Does this Program Do – Strings**

The first loop looks for k's and letters after m and puts them in b.

3

It also looks for i's and letters after w and puts them in c.

The result is: b = "ktstssksnwvsownyrontostvs" and c = "iiyii"

d is the concatenation of b and c. The next loop compares 2 letters at a time and only places a letter in e if the first is alphabetically after the second letter. It places the second letter in e resulting in "ssknvsonronosii". The last loop counts the number of n's in e. The answer is 3.

49. **14-15 C1 What Does this Program Do – Branching**

The table contains the values of a, b, c, d and e after each line.

-1

a	b	c	d	e
24	8	-1	2	0
16	8	-1	2	0
16	8	1	2	0
16	8	1	2	2
16	8	-1	2	2
16	8	-1	2	4
20	8	-1	2	4

$$a / e + b * c - d + b / 2 = 20 / 4 + 8 * (-1) - 2 + 8 / 2 = 5 - 8 - 2 + 4 = -1$$

50. **14-15 C2 What Does this Program Do – Loops**

This program sums the digits of N to see which sums are divisible by 9. There are only 2 in the range from 17150 to 17190, inclusive. They are 17154 and 17190.

17154 and 17190

51. **14-15 C3 What Does this Program Do – 2D Arrays**

The original array entries are changed. If the sum of the row and column numbers is even, then the entry is halved and then the greatest integer function is applied. Otherwise, the entry is doubled. The rest of the program finds the largest absolute value of the product of the row entries.

140

52. **14-15 C4 What Does this Program Do – 2D Arrays**

The first nested loop fills the 10x10 array with the numbers 1-100. 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

21

53. **15-16 C1 What Does this Program Do – Branching**

The table contains the values of a, b, c and d after each line.

34

a	b	C	d
4	10	1	2
14	10	1	2
14	10	-1	2
14	12	-1	2
14	12	2	2
14	2	2	2

$$a / b + c * (a * b - c / d) / b = 14 / 2 + 2 * (14 * 2 - 2 / 2) / 2 = 7 + 2 * 27 / 2 = 7 + 27 = 34$$

54. 15-16 C2 What Does this Program Do – Loops

This program counts the integers from 1 to 50 inclusive that are divisible by 2, 3 or 5. There are 36 such integers. S counts the ones that are not, which is 89 since 125 integers were checked in the three loops.

36, 89

55. 15-16 C3 What Does this Program Do – 2D Arrays

This program checks for entries in the table that are divisible by 4 and 2 and greater than or equal to 20. Entries in A are adjusted accordingly. The final table is below. There are 5 entries less than or equal to 10 so there are 11 others.

11

6	22	2	5
1	30	49	10
1	78	8	9
10	5	18	2

56. 15-16 C4 What Does this Program Do – Strings

The first loop looks for letters before N and puts them in B. Those N's or letters after N are placed in C. The result is B = "LDMAIHEMAI" and C = "ONNTOUNTN". The next loop compares 2 letters at a time in B and places a letter in D if the first is alphabetically before the second letter else it is put in E. The result is D = "DAEA" and E = "LMIHM". The next loop drops A's and E's in D. The result is F = "D". The last loop places the characters in G that are alphabetically after G. The length of G is 0. There are no letters left from A just like the actual Old Man In The Mountain in NH that crumbled away.

0