Problem Solving Using C++
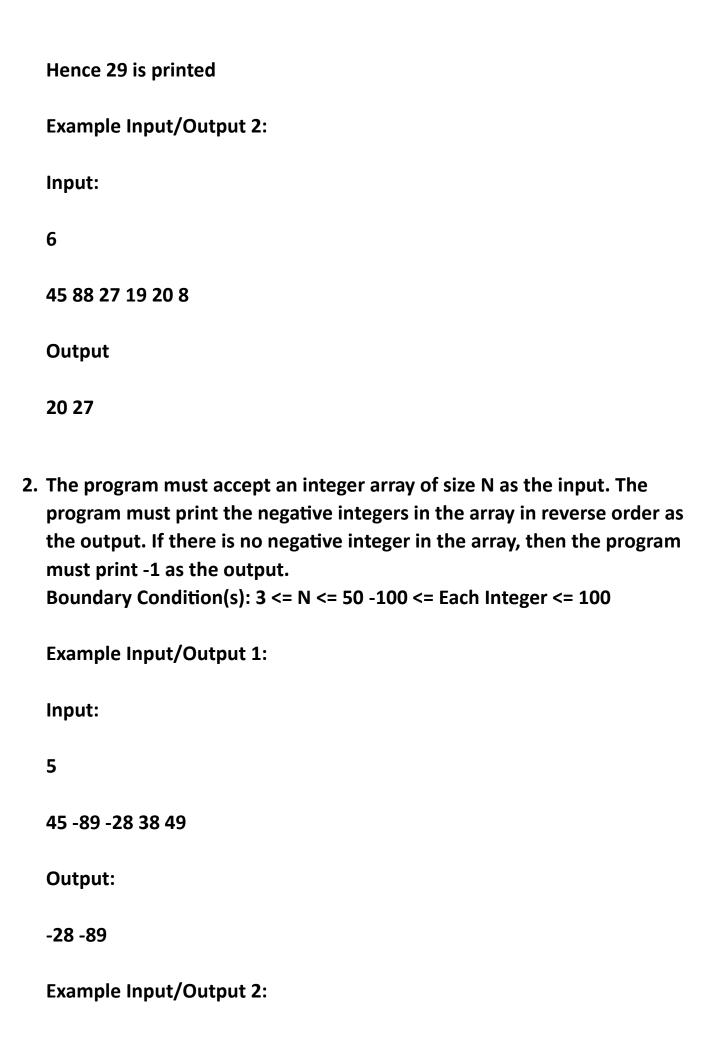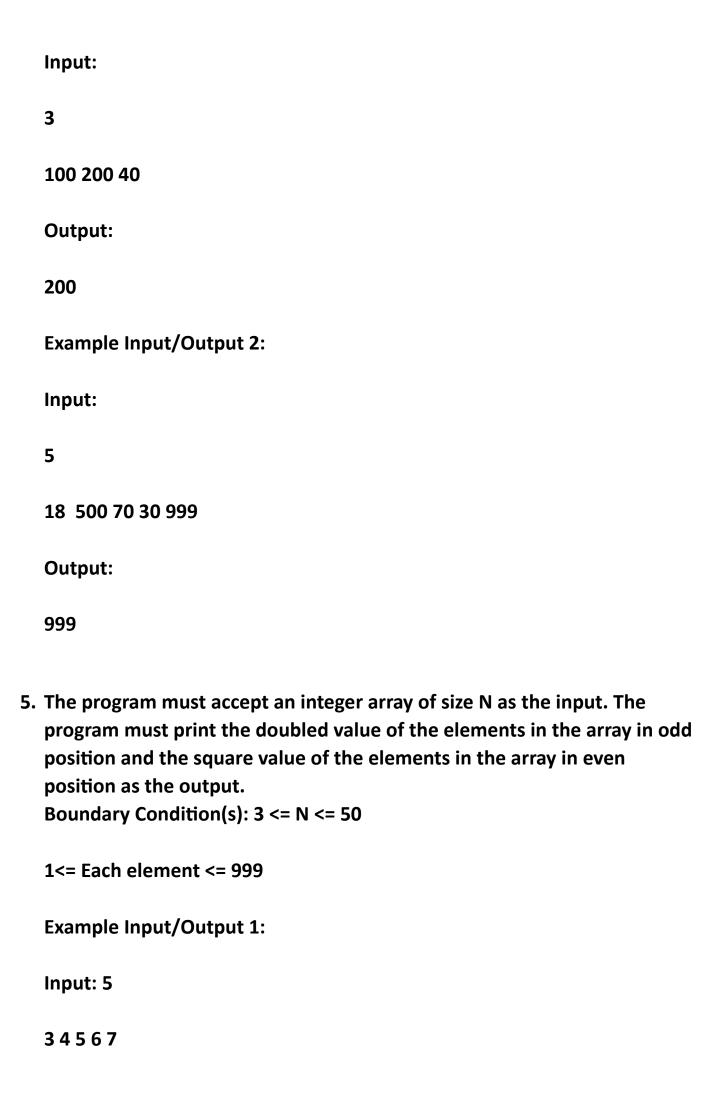
Arrays

1. The program must accept a positive integer array of size of N as the input. The program must print the middle element(s) in the sorted array as the output.
   Boundary Condition(s): 3 <= N <= 50

   1 <= Each array element value <= 999

   Input Format:

   The first line contains the value of N. The second line contains N integers separated by space(s).

   Output Format:

   The first line contains the middle element(s) in the sorted array.

   Example Input/Output 1:

   Input:

   5

   98 26 47 29 10

   Output:

   29

   Explanation:

   The elements in the sorted array are 10 2647 98.

   There is only one middle element 29.
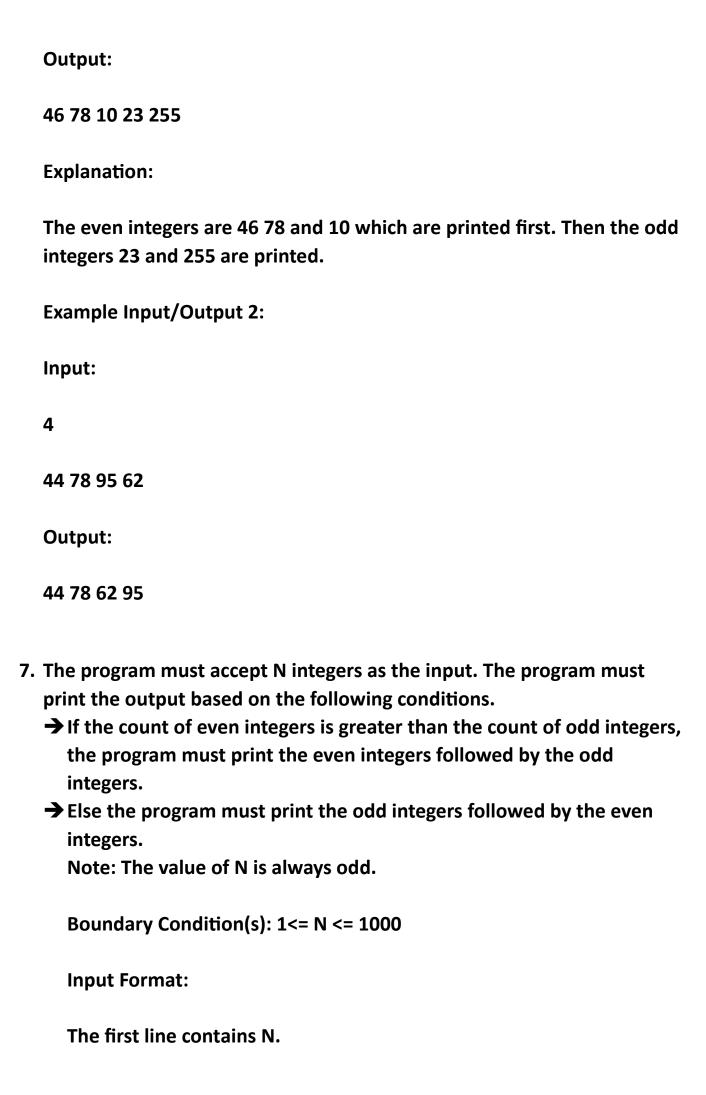
Hence 29 is printed

Example Input/Output 2:

Input:

6

45 88 27 19 20 8

Output

20 27

2. **The program must accept an integer array of size N as the input. The program must print the negative integers in the array in reverse order as the output. If there is no negative integer in the array, then the program must print -1 as the output.**
Boundary Condition(s): 3 <= N <= 50 -100 <= Each Integer <= 100

Example Input/Output 1:

Input:

5

45 -89 -28 38 49

Output:

-28 -89

Example Input/Output 2:

Input

3

12 34 56

Output

-1

3. The program must accept the elements of two arrays of size N as the input. The program must print the elements in the same position as the output.
   Boundary Condition(s): 3 <= N <= 50 1<= Each Element < 99999

   Example Input/Output 1:

   Input:

   5

   6 7 8 5 3

   8 7 5 3 1

   Output:

   6 8

   7 7

   8 5

   5 3

3 1

Example Input/Output 2:

Input:

3

234 364 756

37 927 2234

Output:

234 37

364 927

756 2234

4. **Number N is passed as the input. The program must accept N integer values and print the largest number L among these N numbers.**
Input Format:

The first line denotes the value of N. The next N lines denote the value of N numbers.

Output Format:

The first line denotes the value of L.

Boundary Conditions: 1 <= N <= 50

Example Input/Output 1:

Input:

3

100 200 40

Output:

200

Example Input/Output 2:

Input:

5

18 500 70 30 999

Output:

999

5. The program must accept an integer array of size N as the input. The program must print the doubled value of the elements in the array in odd position and the square value of the elements in the array in even position as the output.
Boundary Condition(s): 3 <= N <= 50

1<= Each element <= 999

Example Input/Output 1:

Input: 5

3 4 5 6 7

Output:

6 16 10 36 14

Example Input/Output 1:

Input: 3
10 24 36

Output:
20 576 72

6. **The program must accept N integers as the input. The program must print even integers followed by odd integers as the output.**
   Boundary Condition(s): 1<= N <= 1000

   Input Format:

   The first line contains N.

   The second line contains N integers separated by space.

   Output Format:

   The first line integers separated by a space.

   Example Input/Output 1:

   Input:

   5

   23 46 78 10 255

Output:

46 78 10 23 255

Explanation:

The even integers are 46 78 and 10 which are printed first. Then the odd integers 23 and 255 are printed.

Example Input/Output 2:

Input:

4

44 78 95 62

Output:

44 78 62 95

7. The program must accept N integers as the input. The program must print the output based on the following conditions.
   ➔ If the count of even integers is greater than the count of odd integers, the program must print the even integers followed by the odd integers.
   ➔ Else the program must print the odd integers followed by the even integers.
   Note: The value of N is always odd.

   Boundary Condition(s): 1<= N <= 1000

   Input Format:

   The first line contains N.

The second line contains N integers separated by space.

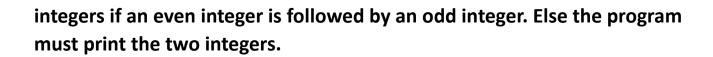**Output Format:**

The first line contains the integer values separated by a space.

Example Input/Output 1:

Input:

5

41 57 65 84 23

Output:

41 57 65 23 84

Explanation:

There are 4 odd integers and 1 even integer, so odd integers are printed first.

Example Input/Output 2:

Input:

7

59 64 72 76 147 45 78

Output:

64 72 76 78 59 147 45

8. The program must accept an integer array of size N as the input. The program must print the maximum and the minimum integers in the array as the output.
   Boundary Condition(s): 5 <= N <= 50 -9999 <= Each Integer <= 999

   Example Input/Output 1:

   Input:

   5

   37 48 29 18 89

   Output: 89 18

   Example Input/Output 2:

   Input:

   6
   -37 -93 -98 -123 -3874 -4874

   Output: -37 -4874

9. The program must accept N integers as the input. The program must print each integer multiplied by the unit digit of the next integer in the reverse order as the output. The last integer is multiplied by the unit digit of the first integer.
   Boundary Condition(s): 1 <= N <= 10^4

   Input Format:

   The first line contains the integer N. The second line contains N integers separated by space(s).

Output Format: The first line contains N integers separated by space(s).

Example Input/Output 1:

Input:

5

40 123 12 25 19

Output:

0 225 60 246

120

Explanation:

For integer 40, the next integer is 123. The unit digit of 123 is 3. So 40*3 = 120
For integer 123, the next integer is 12. The unit digit of 12 is 2. So 123*2 = 246
For integer 12, the next integer is 25. The unit digit of 25 is 5. So 12*5 = 60
For integer 25, the next integer is 19. The unit digit of 19 is 9. So 25*9 = 225
For integer 19, the first integer is 40. The unit digit of 40 is 0. So 19*0 = 0
The integers in the reverse order are 0, 225, 60, 246 and 120 Hence the output is 0 225 60 246 120

10.　　The program must accept N integers as the input. The program must print all the integers that are divisible by their unit digit in reverse order. If the unit digit is zero, then the number must be divisible by 10. Boundary Condition(s): 1 <= N <= 10^4

Input Format:

The first line contains the integer N.

The second line contains N integers separated by space(s).

Output Format: The first line contains the integers which are divisible by their unit digit.

Example Input/Output 1:

Input:

5

102 554 990 87 544

Output:

544 990 102

Explanation:

The integer 102 is divisible by 2 (the unit digit of 102 is 2).

The integer 554 is not divisible by 4 (the unit digit of 554 is 4).
The integer 990 is divisible by 10.

The integer 87 is not divisible by 7 (the unit digit of 87 is 27).

The integer 544 is divisible by 4 (the unit digit of 554 is 4).
The integers reverse order are 544, 990 and 102 Hence the output is 554 990 102

11. The program must accept an array of positive integers of size N as the input. The program must print the even numbers in the array in forward order followed by the odd numbers in the array in reverse order as the output.

Boundary Condition(s):

3<= N <= 100

Example Input/Output 1:

Input:

5

24 56 73 88 97

Output:

24 56 88 97 73

Example Input/Output 2:

Input

5

37 98 57 44 11

Output:

98 44 11 57 37

12. The program must accept an array of size N as the input. The program must print the product of two integers if an odd integer is followed by an even integer. The program must print the sum of two

integers if an even integer is followed by an odd integer. Else the program must print the two integers.

Boundary Condition(s): 3<= N <= 50 1 <= Each Integer <= 99

Example Input/Output 1:

Input:

5

1 2 3 4 5

Output:

2 5 1 2 9

13.   The program must accept N integers and an integer X as the input. The program must find the sum of the first X integers as S1 and then find the sum of the last X integers as S2. Finally, the program must print the sum of S1 and S2 as the output.
Boundary Condition(s):

1<= N <= 1000

1 <= x <= N

Input Format:

The first line conatins the integer N. The second line contains N integers separated by space(s). The third line contains the integer X.

Output Format:

The first line contains the sum of S1 and S2.

Example Input/Output 1:

Input:

8

1 2 3 4 5 6 7 8

3

Output: 27

Explanation:

The first 3 integers are 1, 2 and 3 and their sum (S1) is 6 (1 + 2 + 3). The last 3 integers are 6, 7 and 8 and their sum (S2) is 21 (6+7+8).

The sum of S1 and S2 is 27 (6+21).

Hence the output is 27

14.   The program must accept N integers and an integer X as the input. The program must print the integers after the Xth odd integer in reverse order. If there is no integer after the Xth odd integer, the program must print -1 as the output.
Note: At least X odd integers are always present in the N integers.

Boundary Condition(s): $1 <= N <= 10^4$

$1 < X <= 1000$

Input Format:

The first line contains the two integers N and X separated by a space. The second line contains N integers separated by space(s).

Output Format: The first line contains the integer value(s) or -1.

Example Input/Output 1:

Input:

9 3

10 5 17 22 90 89 20 85 54

Output:

54 85 20

Explanation: Here X = 3.

The 3rd odd integer is 89. So all the integers after 89 are printed in the reverse order.

Hence the output is 54 85 20

15.   The program must accept N distinct integers and an integer X as the input. The program must print the odd integers which are having the unit digit is X or the sum of the last two digits as X in the reverse order as the output. If there is no such integer then the program must print -1 as the output.
Boundary condition(s):

1<= N <= 10^5

1 <= x <= 18

Input Format: The first line contains the two integers N and X separated by a space. The second line contains N integers separated by space(s).

Output Format: The first line contains either the odd integers which are having the unit digit is X or the sum of the last two digits as X in the reverse order or -1.

Example Input/Output 1:

Input:

7 5

10 95 132 141 51 18 55

Output:

55 141 95

Explanation:

X = 5,

All the odd integers in the reverse order are 55, 51, 141 and 95. For the integer 55, the unit digit is 5 which is equal to 5. So 55 is printed.

For the integer 51, the unit digit is 1 which is not equal to 5 and also the sum of last two digits is 6 which is also not equal to 5. So 51 is not printed.

For the integer 141, the sum of last two digits is 5 which is equal to 5. So is printed. For the integer 95, the unit digit is 5 which is equal to 5. So 95 is printed. Hence the output is 55 141 95

"CONSISTENCY IS THE KEY TO SUCCESS"