Name: Model A1

- Write a C# program to print the following pattern:
- use a nested ***for*** loop

```
9 8 7 6
5 4 3
2 1
0
```

Name: Model B1

- Write a C# program to define an array that accept the following values from user

1 7 5 4 8 2 6 9 10

- print array with decrease elements that are greater than or equal 6 and increase elements that are less than 6 by 3
- use a ***for*** or a ***while*** loop

Name: Model A2

- Define a 2D array that holds the following elements

22 16 90
37 18 50

- Use two nested for loops to print the elements that are greater than 25 and divisible by 2 or elements that are less than 25 and divisible by 3

Name: Model B2

- Write a C# program that define 2 arrays with 3 elements accepted from user
- Print the following in third array: sum for each element in first array with corresponding element in second array multipliy 3
- Use a ***for*** or a ***while*** loop

Name: Model A3

- Define an array that holds the following values accepted from user

18 70 33 12 56 30

- Increase the elements that are between 15 and 25 by 2, then print the sum of these elemenst after increment.
- Print the result array in one line
- Use ***foreach*** to print result

Name: Model B3

- Define an array that holds the following values accepted from user

33 67 95 90 56 36

- Print square root for elements that are divisible by 3 but not 2
- Use a ***for*** or a ***while*** loop

Name: Model A4

- Create an integer array with the following elements

72 101 108 108 111

- Print array elements with original value in one line
- Copy elements to another array increased by 6 using ***for*** (in case of second element copy the original value without increment and print array elements in one line.

Name: Model B4

- Write a C# program to define an array that accept the following values from user

1 2 3 4 5 6 7 9 10

• print array with increase elements that have odd index by 4 , and increase elements that have even index by 3
• use a ***for*** or a ***while*** loop

Name: Model A5

- Define a 2D array that holds the following elements

22 16 90
37 18 50

- Use two nested for loops to print the largest and smallest element

Name: Model B5

- Write a C# program that define 2 arrays with 3 elements accepted from user
- Print the following in third array: sum for each element in first array with corresponding element in second array
- Use a ***for*** or a ***while*** loop

Name: Model A6

- Define an array that holds the following values accepted from user

18 70 33 12 56 30

- Multiply each element by its index
- Print the result array in one line
- Use ***foreach*** to print result

Name: Model B6

- Define an array that holds the following values accepted from user

33 67 95 90 56 36

- Print square root for elements that are divisible by 3
- Use a ***for*** or a ***while*** loop

Name: Model A7

- Read 6 integers into a 1D array from user.
- Swap the first and last elements using loops
- print the resulting array in one line (space separated).

Name: Model B7

- Write a C# program to read an array of 8 integers (accept values from user).
- Then read one integer x from the array
- Count and print how many times x appears in the array
- Use any type of loop

Name: Model A8

- Write a C# program to read an array of the following values (accept values from user)

5 -3 0 -1 9 -2

- Replace any negative element with its absolute value and print the modified array.
- Use a ***for*** loop

Name: Model B8

- Write a C# program to read an array of 7 integers (accept values from user)
- Find and print the second largest element.
- If duplicates occur (e.g., max repeated), second largest means the next distinct lower value; if no second distinct, print the max again.
- Use any type of loop

Name: Model A9

- Write a C# program to read an array of 10 integers (accept values from user)
- Print elements that are located at even indices (0,2,4,...)
- Each element printed on same line separated by spaces.
- Use ***for*** to print result

Name: Model B9

- Define a 2D array that holds the following elements

2 5 6
7 8 9

- Print the sum of each column in first row with corresponding column in second row (one per line).
- Use a nested ***for*** loop

Name: Model C1

- Write a C# program to read an array of the following integers (accept values from user).

5 8 3 5 9 8 2 3 5 1

- Print the array, then print the elements that appear more than once in the array.
- Use nested ***for*** loop

Name: Model D1

- Write a C# program to create an array of the following elements:

50 45 40 35 30 25 20 15 10

- Print the elements in reverse order, but only if they are greater than 20.
- Use any type of loop.

Name: Model C2

- Write a C# program to read an array of the following integers (accept values from user).

44 4 18 2 8 64 32 3 47 50 22 12

- Print only numbers that are between 25 and 75 inclusive in one line separated by space
- Use a ***for*** or a ***foreach*** loop.

Name: Model D2

- Write a C# program to read n (≤12) and n integers into array
- Print the index of the first maximum element (0-based).
- If multiple maxima, print the first index.
- Use single ***for*** scanning loop.

Name: Model C3

- Write a C# program to read n (n ≤ 10), then read n integers into an array.
- Compute the average (double) and print only the elements strictly greater than the average, each on one line.
- Use any type of loop

Name: Model D3

- Write a C# program to print the following pattern:
- Use a nested ***for*** loop

1

2 3

4 5 6

7 8 9 10

Name: Model C4

- Write a C# program to read an array of 8 integers (accept values from user).
- Print two numbers: count of even elements and count of odd elements
- (format: Even: X then Odd: Y on next line).
- Use any type of loop

Name: Model D4

- Write a C# program to read an array of 6 integers (accept values from user).
- Reverse the array in-place (swap elements) using loops (no temp array allowed)
- Print the reversed array.
- Use a ***for*** loop and in-place swaps.

Name: Model C5

- Write a C# program to read an array of 10 integers (accept values from user).
- Copy elements at odd indices into a new array and print it in one line.
- Use a ***for*** loop to copy element and print result.

Name: Model D5

- Write a C# program to read an array of 8 integers (accept values from user).
- Print sum of elements at even indices and sum of elements at odd indices.
- Use ***for*** or a ***while*** loop only once.

Name: Model C6

- Write a C# program to read an array of 8 integers (accept values from user).
- Print how many times the array value increases compared to the previous value. (i.e., number of i where arr[i] > arr[i-1]).
- Use any type of loop

Name: Model D6

- Write a C# program to read an array of 8 integers (accept values from user).
- Print only elements divisible by both 2 and 3 (i.e., divisible by 6).
- Use a ***for*** or a ***while*** loop

Name: Model C7

- Write a C# program to read an array of the following integers (accept values from user).

12 4 18 2 8 1 32 3 47 50

- Print only numbers that are between 10 and 50 inclusive in one line separated by space
- Use a ***for*** or a ***foreach*** loop

Name: Model D7

- Write a C# program to read an array of the following integers (accept values from user).

3 10 18 8 4 43 25

- Print the array in the original order, then print only the elements that are greater than 15.

Use a ***for*** or a ***foreach*** loop

Name: Model C8

- Write a C# program to read an array of the following integers (accept values from user).

3 8 12 5 6 11 4 9

- Print the array, then print the sum of all even numbers in the array.
- Use a ***for*** loop

Name: Model D8

- Write a C# program to read an array of 10 integers (accept values from user)
- Print the array, then print the average of all elements rounded to the nearest integer.
- Use any type of loop

Name: Model C9

- Write a C# program to read an array of 8 integers (accept values from user)
- Print the array, then print the difference between the sum of even and odd elements.
- Use any type of loop

Name: Model D9

- Write a C# program to read an array of 9 integers (accept values from user)
- Print the array, then print the maximum and minimum values in the array.
- Use any type of loop