Arrays:

1.

bool isFirstOrLast = true;

if (numbers[0] == 6 || numbers[numbers.Length-1]==6)

{

isFirstOrLast = true;

}

else

{

isFirstOrLast = false;

}

return isFirstOrLast;

2.

bool sameEnds = false;

if (numbers.Length > 0 && numbers[0] == numbers[numbers.Length - 1])

{

sameEnds = true;

}

return sameEnds;

3.

int[] numberOfPi = { 3, 1, 4, 1, 5, 9, 2, 6, 5, 3, 5, 8, 9 };

int[] returnArray = new int[n];

for (int i=0; i<n; i++)

{

returnArray[i] = numberOfPi[i];

}

return returnArray;

4.

bool sameFirstLastElement = false;

if (a[0]==b[0] || a[a.Length-1]==b[b.Length-1])

{

sameFirstLastElement = true;

}

else

{

sameFirstLastElement = false;

}

return sameFirstLastElement;

5.

var sumNumbers = 0;

for (int i = 0; i < numbers.Length; i++)

{

sumNumbers += numbers[i];

}

return sumNumbers;

6.

int[] isRotated = new int[numbers.Length];

for (int i=1; i<numbers.Length; i++)

{

isRotated[i - 1] = numbers[i];

}

isRotated[numbers.Length - 1] = numbers[0];

return isRotated;

7.

int[] isReversed = new int[numbers.Length];

for (int i = 0; i < numbers.Length; i++)

{

isReversed[i] = numbers[numbers.Length - (i + 1)];

}

return isReversed;

8.

int[] newArray = new int[numbers.Length];

int highNum = numbers[0];

if (numbers[0] > numbers[numbers.Length-1])

{

highNum = numbers[0];

}

else

{

highNum = numbers[numbers.Length - 1];

}

for (int i=0; i<numbers.Length; i++)

{

newArray[i] = highNum;

}

return newArray;

9.

int[] newArray = new int[2];

newArray[0] = a[1];

newArray[1] = b[1];

return newArray;

10.

foreach (var i in numbers)

{

if (i%2==0)

{

return true;

}

}

return false;

11.

int[] twiceAsLong = new int[numbers.Length \* 2];

twiceAsLong[twiceAsLong.Length - 1] = numbers[numbers.Length - 1];

return twiceAsLong;

12.

bool hasItTwice = false;

int twoCount = 0;

int threeCount = 0;

for (int i = 0; i < numbers.Length; i++)

{

if (numbers[i] == 2)

{

twoCount += 1;

}

if (numbers[i] == 3)

{

threeCount += 1;

}

}

if (twoCount == 2 || threeCount == 2)

{

hasItTwice = true;

}

else

{

hasItTwice = false;

}

return hasItTwice;

}

13.

for (int i = 0; i < numbers.Length-1; i++)

{

if (numbers[i] == 2 && numbers[i + 1] == 3)

{

numbers[i + 1] = 0;

}

}

return numbers;

14.

bool isUnlucky = false;

if (numbers[0] == 1 && numbers[1] == 3 || numbers[1] == 1 && numbers[2] == 3 || numbers[numbers.Length-2] == 1 && numbers[numbers.Length-1] == 3)

{

isUnlucky = true;

}

else

{

isUnlucky = false;

}

return isUnlucky;

15.

int[] newArray = new int[2];

if (a.Length == 0)

{

newArray[0] = b[0];

newArray[1] = b[1];

}

else if (a.Length == 1)

{

newArray[0] = a[0];

newArray[1] = b[0];

}

else if (a.Length > 1)

{

newArray[0] = a[0];

newArray[1] = a[1];

}

return newArray;