Write a function in python according to the given specification:

Input parameters

A: 2D array of integers

Return value:

Return value is 1 if A contains two neighbour identical columns. Otherwise, the function returns 0.
Two columns are identical if they contain the same values in the same order.

Example:

```
A = [
  [1,1,4,4,5]
  [1,3,4,4,2]
  [3,2,0,0,7]
  [2,3,1,1,2]
  [1,2,3,3,2]
  ]
```

Return value:

1 (the 3rd and the 4th columns are identical).

Write a function in python according to the given specification:

Input parameters

A: array of integers

K, M: integer

Return value:

Return value is 1 if A contains K consecutive values which product is exactly M.

Otherwise, the function returns 0.

Examples:

A = [2,9,1,7,1,2,1,6,2,0,1]

K = 5, 24

Return value is 1 (1*2*1*6*2 = 24).

A = [10, 10, 10, 10, 10]

K = 3, 100

Return value is -1.

Input parameters
A: 2D array of integers

Return value:
The number of columns in A which do not contain value K.

Example:
A = [
[1,1,4,6,5]
[1,3,4,2,5]
[1,2,0,2,5]
[2,3,4,6,5]
[1,2,3,2,5]
]
K = 2

Return value:
2

Write a function in python according to the given specification:

Input parameters

A: 2D array of integers

K, M: integer

Return value:

The smallest index of the column which contains value K exactly M times. If such column does not exit the function returns -1.

Example:

```
A = [
  [1,2,4,6,5]
  [1,3,4,2,5]
  [1,2,0,2,5]
  [3,3,4,6,5]
  [1,2,4,2,5]
  ]
K = 2; M = 3
```

Return value:

1

Write a function in python according to the given specification:

Input parameters

A: 2D array of integers, with same number of rows and column

K: integer

Return value:

Array A.

All values in the main diagonal of A are increased by value of K.

Example:

```
A = [
  [8,7,6,5,4]
  [7,6,5,4,3]
  [6,5,4,3,2]
  [5,4,3,2,1]
  [4,3,2,1,0]
  ]
K = 1

Return value:
A = [
  [9,7,6,5,4]
  [7,7,5,4,3]
  [6,5,5,3,2]
  [5,4,3,3,1]
  [4,3,2,1,1]
  ]
```

Write a function in python according to the given specification:

Input parameters

A: array of integers

Return value:

Return value is 1 if the product of any two elements in A is either bigger than 100 or less than -100.

Otherwise the return value is 0.

Examples:

A = [12, 13, 14, -11, -20, 65]

Return value is 1.

A = [70, 1000, -420, 5, 200, 200, 7]

Return value is 0.

Write a function in python according to the given specification:

Input parameters

A: array of integers

Return value:

One integer representing the number of such elements of A which value differs from the average value of A by more than 2. (The average value of an array is the average value of all its elements.)

Examples:

A = [2,4,1,7,3,0,2,0,1]
Return value is 3.

A = [1000, 1000, 420, 10, 10]
Return value is 5.

Determine the return value and the total number of executions of the innermost loop statement when the given function is called with parameter
a) n=1, b) n=2 c) n=5 d) n = k.
Suppose that len(a) == k.

```
def f( a, n ):    # a is an array
  if n == 0: return 0
  s = f(a, n - 1)
    for i in range( len(a) )
      s += a[i]
  return s
}
```

What will be the effect of statement f(a, 0, 1)?
Describe in words the changes which will occur in array a.

```
def f( a, i, j ):
  while j < len(a) and a[j] != 0:
     j += 1
  if j >= len(a): return
  a[j],a[i] = a[i],a[j]
  f(a, i+1, j+1)
```

Determine the output produced by the statement **`print(recur(90)`**. The function **`recur`** is defined below.

```
def recur( x ):
  if x < 20: return 20
  return recur(x-20) + recur(x-40)
```

Which of the two following program fragments will take less time to complete? (In an identical HW/SW environment)?

```
n = 100
sum = 0
for i in range(n):
    for j in range(i):
        sum += i + j
```

```
n = 75
sum = 0
for i in range(n):
    for j in range(n):
        sum += i + j
```

Fill in the missing constant in the given code. The procedure `uvw()` should be called exactly 49 times.

```
for i in range(7):
    j = i
    while j < ___:
        uvw()
        j += 1
```