import math  
  
  
def mean(arr**,** n):  
 *"""  
 Determines the mean of all numbers in an array* ***:param*** *arr: list of numbers (list)* ***:param*** *n: length of the array (int)* ***:return****: mean of the elements in the array (float)  
 """* if n == **1**:  
 return arr[**0**]  
 else:  
 return ((n - **1**) / n) \* mean(arr[:n]**,** n - **1**) + (**1** / n) \* arr[n - **1**]  
  
  
def descending\_binary\_search(arr**,** start**,** end**,** k):  
 *"""  
 Locates a search key in an array using binary search* ***:param*** *arr: list of elements to search through (list)* ***:param*** *start: index at which to begin searching (int)* ***:param*** *end: index at which to stop searching (int)* ***:param*** *k: search key (any)* ***:return****: index of search key (int) or None  
 """* m = math.floor((end + start) / **2**)  
 if start > end:  
 print(k**,** " not found in list.")  
 return None  
 else:  
 print("Index checked: "**,** m)  
 if arr[m] == k:  
 return m  
 elif arr[m] > k:  
 return descending\_binary\_search(arr**,** m + **1,** end**,** k)  
 else:  
 return descending\_binary\_search(arr**,** start**,** m - **1,** k)  
  
  
def euclidean\_gcd(a**,** b):  
 *"""  
 Determine the greatest common divisor of two integers using Euclidean algorithm* ***:param*** *a: first integer (int)* ***:param*** *b: second integer (int)* ***:return****: the greatest common divisor (int)  
 """* print("Input Parameters: "**,** a**,** b)  
 if b == **0**:  
 return a  
 else:  
 return euclidean\_gcd(b**,** a % b)  
  
  
def main():  
 # Answers to Question 1b:  
 a1 = [**12, 23, 37, 45, 63, 82, 47, 75, 91, 88, 102**]  
 a2 = [-**1.7, 6.5, 8.2, 0.0, 4.7, 6.3, 9.5, 12.2, 37.9, 53.2**]  
  
 x = mean(a1**, 11**)  
 y = mean(a2**, 10**)  
  
 print("Answers for Question 1:")  
 print("Mean a1:"**,** x**,** " Mean a2:"**,** y)  
 print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")  
  
 # Answers to Question 2b  
 a3 = [**100, 87, 85, 80, 72, 67, 55, 50, 48, 42, 40, 31, 25, 22, 18**]  
  
 print("Answers for Question 2:")  
 print("Search for 87:")  
 descending\_binary\_search(a3**, 0,** len(a3) - **1, 87**)  
 print("--------------")  
 print("Search for 48:")  
 descending\_binary\_search(a3**, 0,** len(a3) - **1, 48**)  
 print("--------------")  
 print("Search for 33:")  
 descending\_binary\_search(a3**, 0,** len(a3) - **1, 33**)  
 print("--------------")  
 print("Search for 10:")  
 descending\_binary\_search(a3**, 0,** len(a3) - **1, 10**)  
 print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*")  
  
 # Answers to Question 3b  
 print("Answers for Question 3:")  
  
 print("Calls for 3bi:")  
 z = euclidean\_gcd(**2468, 1357**)  
 print("Result 3bi: "**,** z)  
  
 print("Calls for 3bii:")  
 w = euclidean\_gcd(**111, 378**)  
 print("Result 3bii: "**,** w)  
  
 print("Calls for 3biii:")  
 s = euclidean\_gcd(**123456789, 987654321**)  
 print("Result 3biii: "**,** s)  
  
  
main()

Answers for Question 1:

Mean a1: 60.45454545454545 Mean a2: 13.68

-----------

Answers for Question 2:

Search for 87:

Index checked: 7

Index checked: 3

Index checked: 1

--------------

Search for 48:

Index checked: 7

Index checked: 11

Index checked: 9

Index checked: 8

--------------

Search for 33:

Index checked: 7

Index checked: 11

Index checked: 9

Index checked: 10

33 not found in list.

--------------

Search for 10:

Index checked: 7

Index checked: 11

Index checked: 13

Index checked: 14

10 not found in list.

--------------

Answers for Question 3:

Calls for 3bi:

Input Parameters: 2468 1357

Input Parameters: 1357 1111

Input Parameters: 1111 246

Input Parameters: 246 127

Input Parameters: 127 119

Input Parameters: 119 8

Input Parameters: 8 7

Input Parameters: 7 1

Input parameters: 1 0

Result 3bi: 1

Calls for 3bii:

Input Parameters: 111 378

Input Parameters: 378 111

Input Parameters: 111 45

Input Parameters: 45 21

Input Parameters: 21 3

Input parameters: 3 0

Result 3bii: 3

Calls for 3biii:

Input Parameters: 123456789 987654321

Input Parameters: 987654321 123456789

Input Parameters: 123456789 9

Input parameters: 9 0

Result 3biii: 9