namespace recursya\_answers

{

class Program

{

// 15

// כתוב פעולה רקולסיבית המקבלת מערך ומציין (אינדקס) של איבר. הפעולה מחזירה את מספר

// האיברים החיוביים מתחילים המערך ועד למציין

public static int counter(int[] arr, int i)

// פעולה מקבלת מערך ואינדקס

// הפעולה מחזירה את מספר האיברים החיוביים מתחילת המערך ועד האינדקס

{

if (i == -1) //שלא יחרוג

return 0;

if (arr[i] < 0) //בודק אם חיוביים

return counter(arr, i - 1);

return counter(arr, i - 1) + 1;

}

static void Main(string[] args)

{

// 15

// uses "Counter()"

//Console.WriteLine(Zogi(16));//1\*3\*5\*7\*9\*11\*13\*15=2,027,025

int[] arr = { 1, 2, 3, -4, -5, 6 };

int i = int.Parse(Console.ReadLine());

Console.WriteLine(counter(arr, i));//counter(arr,i) =>

// 17

// כתוב פעולה בוליאנית רקולסיבית המקבלת מערך ובודקת אם אברי המערך ממוינים בסדר

// עולה, ומחזירה (נכון) אם כן, או (לא נכון) אחרת. הוסף פרמטרים לפי שיקול דעתך

public static bool IsSorted(int[] arr, int begin, int end, int before)

// פונקציה רקורסיבית שבודקת האם המערך מסודר מהמספר הקטן לגדול

{

if (begin > end)

// true תנאי עצירה חיובי = אם האינדקס הראשון עבר את האינדקס האחרון אז התנאי מחזיר

return true;

if (arr[begin] < before)

// flase תנאי עצירה שלילי = אם האיבר במיקום ההתחלתי קטן מהמשתנה הקודם אז התנאי מחזיר

return false;

before = arr[begin];

//(begin) המשתנה הקודם לאיבר במיקום ההתחלתי) להיות האיבר במיקום התחלתי) before הופכים את

return IsSorted(arr, begin + 1, end, before); // קריאה לפונקציה הרקורסיבית

}

static void Main(string[] args)

{

// 17

// uses "IsSorted()"

int[] arr1 = { 1, 3, 5, 10, 9 }; // יצירת מערך

Console.WriteLine(IsSorted(arr1, 0, arr1.Length - 1, 0)); // IsSorted(arr1, 0, 4, 0) -> IsSorted(arr1, 1, 3, 1) -> IsSorted(arr1, 2, 2, 3) -> IsSorted(arr1, 3, 1, 5) -> IsSorted(arr1, 4, 0, 10) = false

int[] arr2 = { 1, 3, 5, 7, 9, 10 }; // יצירת מערך

Console.WriteLine(IsSorted(arr2, 0, arr2.Length - 1, 0)); // IsSorted(arr2, 0, 5, 0) -> IsSorted(arr2, 1, 4, 1) -> IsSorted(arr2, 2, 3, 2) -> IsSorted(arr2, 3, 2, 3) -> IsSorted(arr2, 4, 1, 7) -> IsSorted(arr2, 5, 0, 9) -> IsSorted(arr2, 6, -1, 10)= true

// 18

// כתוב פעולה בוליאנית רקורסיבית המחזירה (נכון) אם אין מערך מספרים ראשוניים, אחרת

// מחזירה הפעולה (לא נכון). היעזר הפעולה שכתבת בשאלה 8. הוסף פרמטרים לפי שיקול דעתך

public static bool prime(int num, int half)

{

//מקבל מספר ומחצית ממנו ומחזיר אמת אם הוא ראשוני

if (half == 1)

return true;

else if (num % half == 0)

return false;

return prime(num, half - 1);

}

public static bool targil\_18(int[] arr, int i, int k)

{

//טענת כניסה: הפעולה מקבלת מערך

// טענת יציאה: הפעולה מחזירה אמת אם יש במערך מספרים ראשוניים ושקר אם לא

if (k == arr.Length) // למנוע חריגה מגבולות המערך

return true;

if (prime(arr[k], i) == true) // simple case

return false;

return targil\_18(arr, i, k + 1); // complicated case

}

static void Main(string[] args)

{

// 18

// uses "prime()" and "targil\_18()"

int[] a = { 2, 4, 7 };

int[] a1 = { 4, 6, 8 };

Console.WriteLine(targil\_18(a, 2, 0)); // targil\_18(a,2,0)=>targil\_18(a,2,1)=>targil\_18(a,2,2)=>false

Console.WriteLine(targil\_18(a1, 2, 0)); // targil\_18(a1,2,0)=>targil\_18(a1,2,1)=>targil\_18(a1,2,2)=>targil\_18(a1,2,3)=>true

// 20

// כתוב פעולה בוליאנית המקבלת מערך ומגרילה 2 ערכים המהווים מציינים המערך

// הפעולה בודקת אם אברי המערך שבין שני מציינים אלה מהווים פלינדרום או לא, ומחזירה (נכון

// או (לא נכון) בהתאם. היעזר בפעולה רקולסיבית

public static bool Poli\_arr(int[] arr)

{

Random rnd = new Random();

int num = rnd.Next(0, arr.Length - 1);

int num2 = rnd.Next(0, arr.Length - 1);

int max = Math.Max(num, num2);

int min = Math.Min(num, num2);

return Poli\_arr(arr, min, max);

}

public static bool Poli\_arr(int[] arr, int min, int max)

{

if (min == max || min == max - 1 && arr[min] == arr[max])

return true;

if (arr[min] != arr[max])

return false;

return Poli\_arr(arr, min + 1, max - 1);

}

static void Main(string[] args)

{

// 20

// uses "Poli\_arr()"

int[] arr\_20 = { 1, 2, 3, 3, 2, 1 };

Console.WriteLine(Poli\_arr(arr\_20));

//Poli\_arr( arr) => Poli\_arr( arr, 1, 3)=> false

//Poli\_arr( arr) => Poli\_arr( arr, 0, 5)=>Poli\_arr( arr, 1, 4)=>Poli\_arr( arr, 2, 3)=>true

// 31

// כתוב פעולה רקולסיבית המקבלת מערך ומדפיסה את איברי המערך שהמציינים שלם זוגיים.

// הוסף פרמטרים לפי שיקול דעתך

public static void PrintIndexZogi1(int[] arr, int index)

{//מקבלת מערך ומדפיסה את אברי המערך במיקומים הזוגיים

if (index < arr.Length)

{

Console.WriteLine(arr[index]);

PrintIndexZogi1(arr, index + 2);

}

}

public static void PrintIndexZogi(int[] arr)

{//קוראת לפעולת עזר

PrintIndexZogi1(arr, 0);

}

//32

// כתוב פעולה רקולסיבית המקבלת מערך ומדפיסה את אברי המערך הקטנים מהאיבר העוקב

// להם. הוסף פרמטרים לפי שיקול דעתך.

public static void STB(int[] nums, int i) //32

{

if (i < nums.Length) //complecated

{

if (nums[i] > nums[i - 1])

Console.WriteLine(nums[i]);

STB(nums, i + 1);

}

}

public static void CSTB(int[] nums) //32

{

STB(nums, 1);

}

static void Main(string[] args)

{

// 32

// uses "CSTB()" and "STB"

int[] num3 = { 1 };

int[] num = { 1, 2, 3, 4 };

int[] num2 = { 7, 9, 3, 4 };

CSTB(num); //CSTB(num) => STB(num, 1) => STB(num, 2) => STB(num, 3) => STB(num, 4) => stop

//output: 2 3 4

CSTB(num2);//CSTB(num) => STB(num2, 1) => STB(num2, 2) => STB(num2, 3) => STB(num2, 4) => stop

//output: 9 4