

Laboratory work #1

Performance, Data Structure & Algorithms

“Insertion sort”

Group:ITDS-1901

Ibrayeva Madina

Implement INSERTION-SORT using pseudocode that was given in the lecture.

for i=1 to arr.n

d= arr[ind]

k=ind

while k > 0 and arr[k - 1] > d:

arr[k] = arr[k - 1]

k = k - 1

arr[k] = d

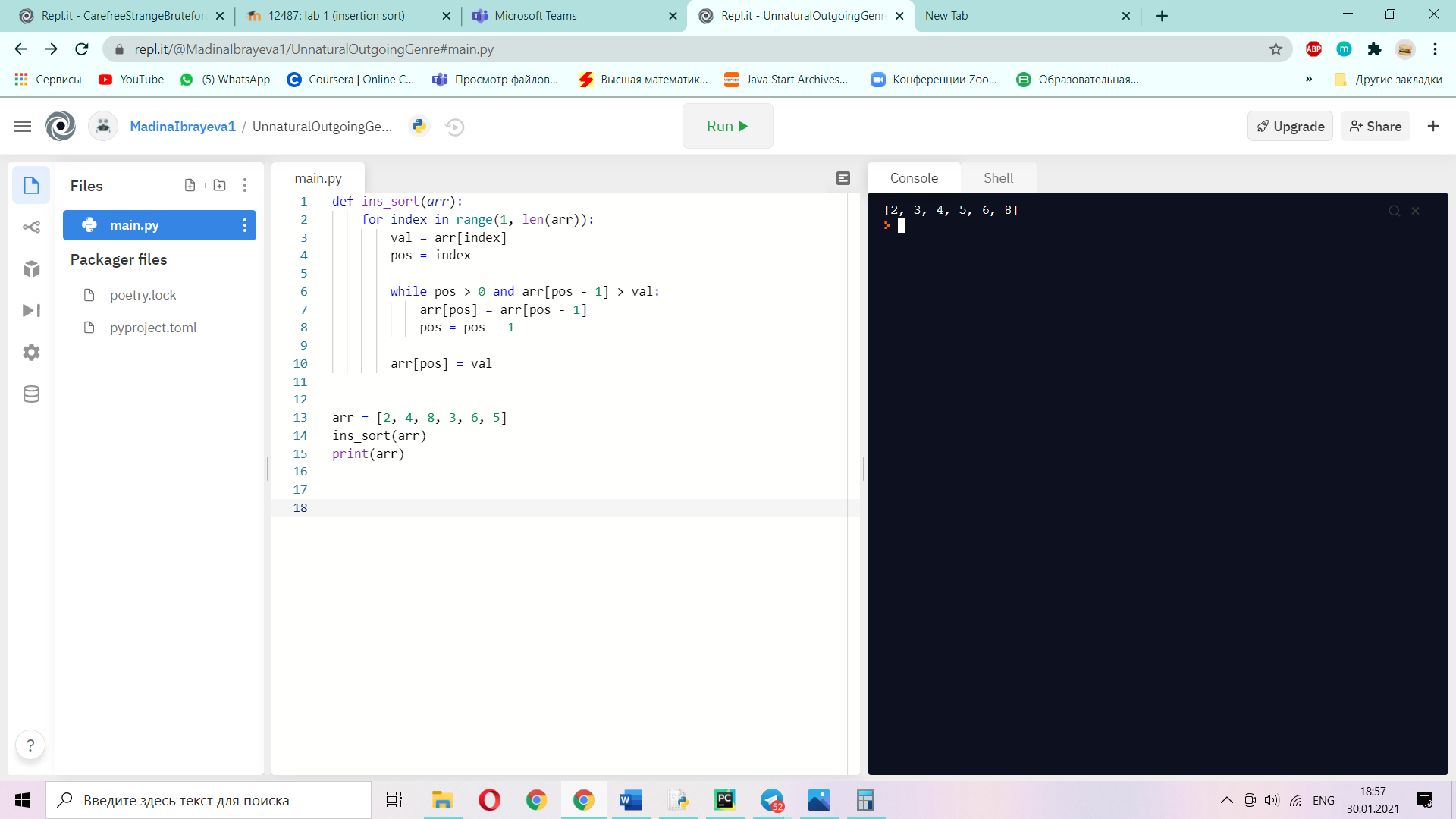
1. Illustrate the operation of INSERTION-SORT on the array A = (31; 41; 59; 26; 41; 58)

Изображение выглядит как стол

Автоматически созданное описание

1. Rewrite the INSERTION-SORT procedure to sort into nonincreasing instead of nondecreasing order.

*'Insertion sort'*def ins\_sort(arr):  
 for index in range(1, len(arr)):  
 val = arr[index]  
 pos = index  
  
 while pos > 0 and arr[pos - 1] > val:  
 arr[pos] = arr[pos - 1]  
 pos = pos - 1  
  
 arr[pos] = val  
  
  
arr = [2, 4, 8, 3, 6, 5]  
ins\_sort(arr)  
print(arr)

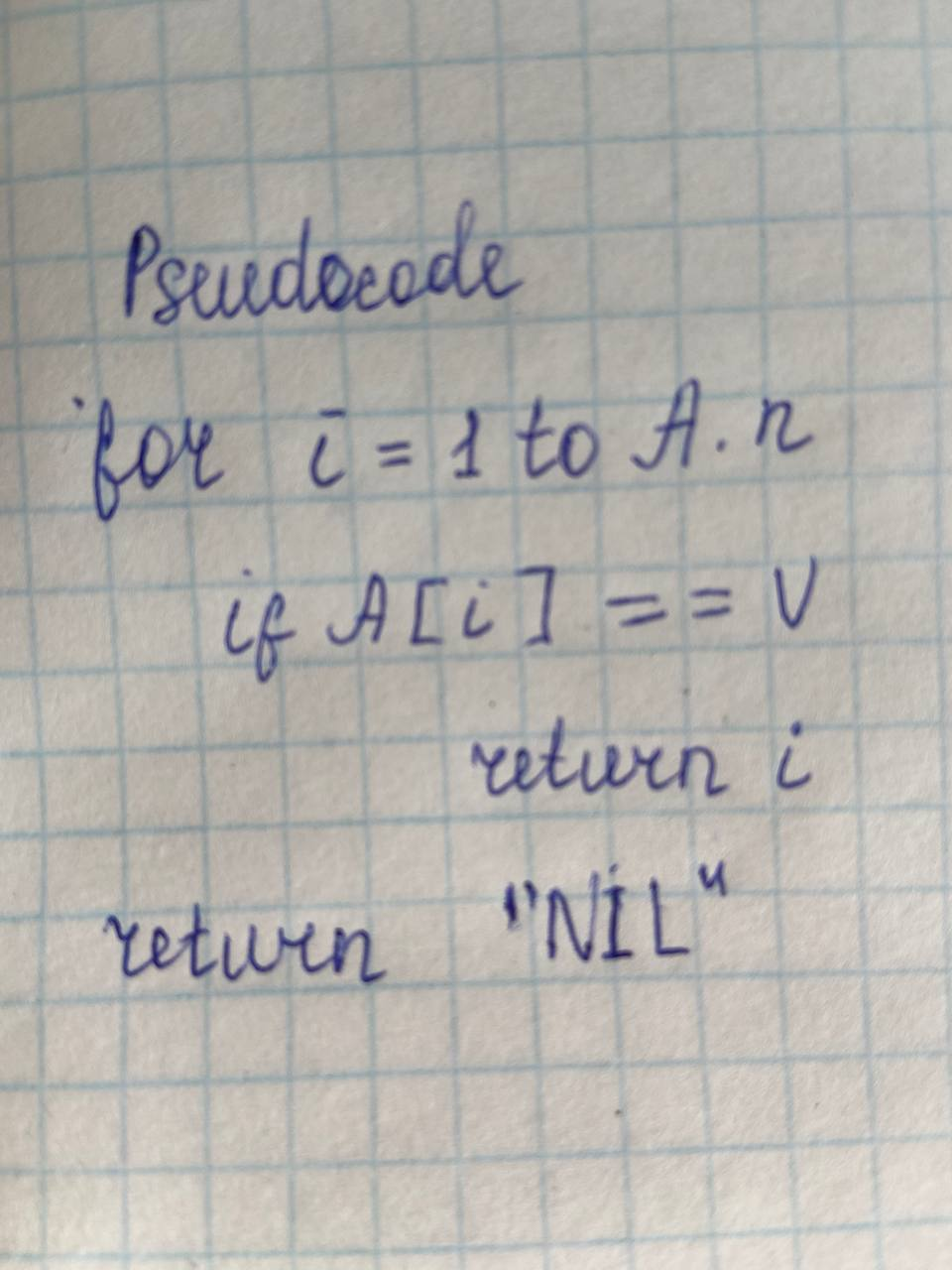


1. Consider the searching problem:

**Input**: A sequence of n numbers A = (a1, a2, … ,an) and a value V.

**Output**: An index i such that V = A[i] or the special value NIL if V does not appear in A.

Write pseudocode for linear search, which scans through the sequence, looking for V. Using a loop invariant, prove that your algorithm is correct. Make sure that your loop invariant fulfills the three necessary properties.



In 4th task we should write a code that will find number V in the sequence A, in output must be index of number that equal to V, not just number. And if there is no number that equal to V, output would be “NIL”

We used loop invariant for that is searching number in sequence.

1. Consider the problem of adding two n-bit binary integers, stored in two n-element arrays A and B. The sum of the two integers should be stored in binary form in an (n+1)-element array C. State the problem formally and write pseudocode for adding the two integers.

