5.1 task

Дано:

Входные параметры:

Два одномерных массива из одинакового количества элементов

Найти:



Выходные параметры:

Неотсортированный массив, отсортированный массив

Дополнительные условия:

Ввод числа n, выделение памяти и заполнение массива случайными

числами.

Вводится m - индекс элемента до которого выполняется пузырек с

проверкой на введенное значение (2, n-3).

Сортировка массив до m по неубыванию с помощью «пузырька».

Сортировка оставшийся части массива по невозрастанию с помощью

алгоритма сортировки массива выбором.

Текст программы:

from random import randint

def main():

num = int(input('Количество элементов массива: '))

floor = int(input('Нижний порог: '))

ceiling = int(input('Верхний порог: '))

mainList = [randint(floor, ceiling) for num in range(num)]

def listinput():

for curIndex in range(0, len(mainList)):

if curIndex % 5 == 0:

print()

print(mainList[curIndex], end=' ')

print('\nНеотсортированный список: ', end='')

listinput()

m = int(input('\n\nДо какого пузырьком?: '))

for border in range(m):

for step in range(m - 1, -1, -1):

if mainList[step + 1] < mainList[step]:

mainList[step], mainList[step + 1] = mainList[step + 1], mainList[step]

for border in range(m, num - 1):

numMin = border

for step in range(border + 1, num):

if mainList[step] < mainList[numMin]:

numMin = step

mainList[border], mainList[numMin] = mainList[numMin], mainList[border]

print('\nОтсортированный список: ', end='')

listinput()

main()

Результат тестирования:





5.2 taskhome

Дано:

Входные параметры:

Два одномерных массива из одинакового количества элементов

Задание:



Выходные параметры:

Отсортированный массив одним из способов сортировки, время и количество перестановок

Дополнительные условия:



Текст программы:

from random import randint  
from datetime import datetime  
  
def main():  
  
 length = int(input('Количество элементов списка: '))  
 floor = int(input('Минимальный диапазон значений элементов: '))  
 ceiling = int(input('Максимальный диапазон значений элементов: '))  
  
 list = [randint(floor, ceiling) for length in range(length)]  
  
 def sortBubble():  
 start\_time = datetime.now()  
 k = 0  
 for i in range(length - 1):  
 for j in range(length - 2, i - 1, -1):  
 if list[j + 1] < list[j]:  
 list[j], list[j + 1] = list[j + 1], list[j]  
 k += 1  
 end\_time = datetime.now()  
 print('Количество перестановок:', k)  
 print('Время выполнения: {}'.format(end\_time - start\_time))  
  
  
 def sortChoice():  
 start\_time = datetime.now()  
 k = 0  
 for i in range(length - 1):  
 nMin = i  
 for j in range(i + 1, length):  
 if list[j] < list[nMin]:  
 nMin = j  
 if i != nMin:  
 list[i], list[nMin] = list[nMin], list[i]  
 k += 1  
  
 end\_time = datetime.now()  
 print('Количество перестановок:', k)  
 print('Время выполнения: {}'.format(end\_time - start\_time))  
  
  
 def sortInsert():  
 k = 0  
 start\_time = datetime.now()  
 for i in range(1, length):  
 indx = list[i]  
 j = i - 1  
 while j >= 0 and list[j] > indx:  
 list[j + 1] = list[j]  
 j -= 1  
 k += 1  
 list[j + 1] = indx  
 end\_time = datetime.now()  
 print('Время выполнения: {}'.format(end\_time - start\_time))  
 print('Количество перестановок:', k)  
  
  
 def menu():  
 n = 1  
 while n != 0:  
 print('\n1: Сделать сортировку пузырьком, \n2: Сделать сортировку выбором, \n3: Сделать сортировку вставкой\n4: Вывести список\n0: Выйти из программы')  
 n = int(input())  
 if n == 1:  
 sortBubble()  
 if n == 2:  
 sortChoice()  
 if n == 3:  
 sortInsert()  
 if n == 4:  
 for i in range(0, len(list)):  
 if i % 5 == 0:  
 print()  
 print(list[i], end=' ')  
 print('')  
  
 menu()  
  
main()

Результат тестирования:







