**Міністерство освіти і науки України**

**Національний технічний університет України «Київський політехнічний**

**інститут імені Ігоря Сікорського"**

**Факультет інформатики та обчислювальної техніки**

**Кафедра інформатики та програмної інженерії**

**Звіт**

з лабораторної роботи № 9 з дисципліни

«Алгоритми та структури даних 2. Структури даних»

**«Динамічне програмування»**

**Виконав** ІП-11Трикош Іван Володимирович

**Перевірила** Халус Олена Андріївна

Київ 2022

**Код програми**

def readfile():

'''Функція для зчитування даних з файлу'''

filename = "input.txt"

l = []

with open(filename, "r") as file:

l = file.readlines()

return l

def convert(l):

'''Функція для перетворення даних в цілий тип'''

l[0] = l[0].split()

Weight = int(l[0][0])

l = l[1:]

for i in range(len(l)):

l[i] = l[i].split()

l[i][0], l[i][1] = int(l[i][0]), int(l[i][1])

return l, Weight

def algorithm(weight, array):

'''Функція для "наповнення рюкзака"'''

V = [[0 for i in range(weight + 1)] for i in range(len(array))] # Заповнюємо матрицю нулями

for i in range(1, len(array)): # Проходимо по рядках матриці

for x in range(weight + 1): # Проходимо по стовпцях матриці

if x < array[i][1]: # Якщо вага предмету більша за максимальну поточну, то присвоюємо:

V[i][x] = V[i - 1][x]

else:

# Знаходимо, яке із значень V[i - 1][x] та V[i - 1][x - array[i][1]] + array[i][0] більше та присвоюємо його елементу матриці

if V[i - 1][x] >= V[i - 1][x - array[i][1]] + array[i][0]:

V[i][x] = V[i - 1][x]

else:

V[i][x] = V[i - 1][x - array[i][1]] + array[i][0]

return V[-1][-1] # Повертаємо останній елемент матриці

def writefile(S):

'''Функція для запису результати у файл'''

filename = "output.txt"

with open(filename, "w") as file:

file.write(str(S))

return

def main():

array = readfile()

array, weight = convert(array)

S = algorithm(weight, array)

writefile(S)

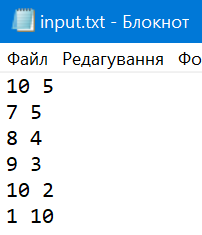
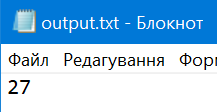
return

if \_\_name\_\_ == "\_\_main\_\_":

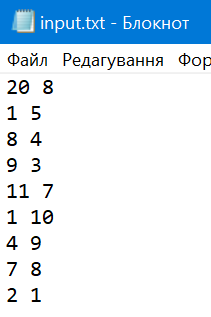
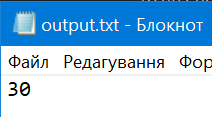
main()

**Приклади роботи програми**

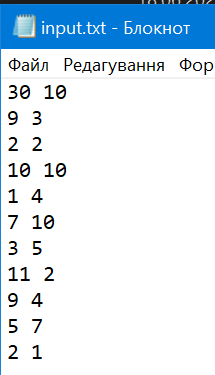
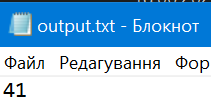
*Приклад 1.*

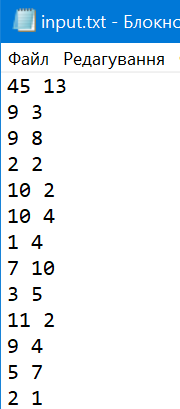
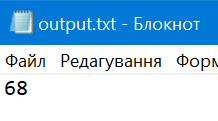
*Приклад 2.*

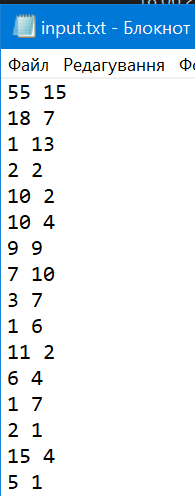
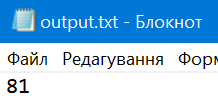
*Приклад 3.*

*Приклад 4.*

* *

*Приклад 5.*

* *