

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

|  |  |
| --- | --- |
| Факультет | Информатика и вычислительная техника |
|  | (наименование факультета) |
| Кафедра | Кибербезопасность информационных систем |
|  | (наименование кафедры) |

**ОТЧЕТ по практической работе**

**по дисциплине “Методы программирования”**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Автор |  | | |  | Карпов Д.И. | | | |
|  | (подпись, дата) | | |  |  | | | |
| Обозначение | 10.05.01.550000.000 О | | Группа | | | | ВКБ31 |
| Направление подготовки | | 10.05.01 Компьютерная безопасность | | | | | |
| Профиль | Компьютерная безопасность | | | | | | |
| Преподаватель |  | | |  | | Савельев В.А. | |
|  | (подпись, дата) | | |  | |  | |

г. Ростов-на-Дону

2021 год

**Лабораторная работа №2**

**Алгоритмы на массивах, сортировка и поиск: скользящее окно, два указателя, префиксные суммы, стандартная сортировка и классификация алгоритмов сортировки, линейный, бинарный, тернарный поиск**

Дано NN натуральных чисел. Требуется для каждого числа найти количество вариантов разбиения его на сумму двух других чисел из данного набора.

Входные данные

В первой строке дано число NN ( 1 ≤ NN ≤ 10000). Далее заданы NN натуральных чисел, не превосходящих 109109. Для каждого числа количество разбиений меньше 231.

Выходные данные

Вывести NN чисел – количество разбиений, в порядке, соответствующем исходному.

**Примеры**

|  |  |
| --- | --- |
| входные данные  5  3  3  2  2  1 | выходные данные  2  2  0  0  0 |

def find(c,v) :

y = -1

x = len(v)

while x > y + 1:

m = (y + x) // 2

if v[m] >= c:

x = m

else:

y = m

if x == len(v) :

bx = 0

else :

if v[x] == c :

bx = x+1

else :

bx = 0

y = -1

x = len(v)

while x > y + 1:

m = (y + x) // 2

if v[m] > c:

x = m

else:

y = m

if y == -1 :

by = 0

else :

if v[y] == c :

by = x+1

else :

by = 0

return abs(bx-by)

def gen(n,v) :

if n < 3 :

return 0

q = 0

for i in range(1,(n+1)//2) :

q = q + find(i,v) \* find(n-i,v)

return q

def pod(v,s) :

for i in v :

print(gen(i,s))

v = []

for \_ in range(int(input())) :

v.append(int(input()))

c = v[:]

c.sort()

pod(v,c)

