

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ
ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ**

Факультет программной инженерии и компьютерной техники

Лабораторная работа №2

Неделя вторая

Выполнил:

Жумиков Егор Олегович

Преподаватели:

Романов Алексей Андреевич

Волчек Дмитрий Геннадьевич

Оглавление

Задача «a + b»	3
Условие.....	3
Формат входного файла.....	3
Формат выходного файла	3
Решение	3
Задача «a+b ² »	5
Условие.....	5
Формат входного файла.....	5
Формат выходного файла	5
Решение	5
Задача «Сортировка вставками».....	5
Формат входного файла.....	8
Формат выходного файла	8
Решение	8
Задача «Знакомство с жителями Сортлэнда».....	10
Условие.....	10
Формат входного файла.....	10
Формат выходного файла	Ошибка! Закладка не определена.
Решение	10
Задача «Секретарь Своп»	14
Условие.....	14
Формат входного файла.....	Ошибка! Закладка не определена.
Формат выходного файла	Ошибка! Закладка не определена.
Решение	14

Задача «Сортировка слиянием»

Условие

Дан массив целых чисел. Ваша задача — отсортировать его в порядке неубывания с помощью сортировки слиянием.

Чтобы убедиться, что Вы действительно используете сортировку слиянием, мы просим Вас, после каждого осуществленного слияния (то есть, когда соответствующий подмассив уже отсортирован!), выводить индексы граничных элементов и их значения.

Формат входного файла

В первой строке входного файла содержится число n ($1 \leq n \leq 10^5$) — число элементов в массиве. Во второй строке находятся n целых чисел, по модулю не превосходящих 10^9 .

Формат выходного файла

В **последней строке** выходного файла требуется вывести отсортированный в порядке неубывания массив, данный на входе. Между любыми двумя числами должен стоять ровно один пробел.

Все предшествующие строки описывают осуществленные слияния, по одному на каждой строке. Каждая такая строка должна содержать по четыре числа: l_f l_l v_f v_l , где l_f — индекс начала области слияния, l_l — индекс конца области слияния, v_f — значение первого элемента области слияния, v_l — значение последнего элемента области слияния.

Все индексы начинаются с единицы (то есть, $1 \leq l_f \leq l_l \leq n$). **Индексы области слияния должны описывать положение области слияния в исходном массиве!** Допускается не выводить информацию о слиянии для подмассива длиной 1, так как он отсортирован по определению.

Решение

```
from bisect import bisect_left

try:
    inp = open('input.txt', 'r')
    outp = open('output.txt', 'w')

    input = inp.readline
    print = lambda *args: outp.write(' '.join(map(str, args)) + '\n')
except:
    pass

n = int(input())
a = list(map(int, input().split()))

def merge(l1, l2):
    l, r = 0, 0
    m = []

    for i in range(len(l1) + len(l2)):
        if l < len(l1) and r < len(l2):
            if l1[l] > l2[r]:
                m.append(l2[r])
                r += 1
            else:
                m.append(l1[l])
```

```

        l += 1
    elif r < len(l2):
        m.append(l2[r])
        r += 1
    else:
        m.append(l1[l])
        l += 1

return m

def msort(a, l=0, r=len(a)-1):
    if len(a) == 1:
        return a

    mp = len(a) // 2

    m = merge(msort(a[:mp], l, l + mp-1), msort(a[mp:], l + mp, r))
    print(l + 1, r + 1, m[0], m[-1])
    return m

print(*msort(a))

```

Результат

№ теста	Результат	Время, с	Память	Размер входного файла	Размер выходного файла
Max		1.718	25788416	1039245	4403714
1	OK	0.031	9080832	25	106
2	OK	0.031	9015296	6	3
3	OK	0.031	9121792	8	14
4	OK	0.031	9003008	8	14
5	OK	0.031	9125888	42	155
6	OK	0.046	9097216	43	155
7	OK	0.031	9011200	51	179
8	OK	0.031	9146368	45	162
9	OK	0.031	9068544	105	331
10	OK	0.031	9113600	110	344
11	OK	0.046	9023488	107	337
12	OK	0.031	9060352	461	2045
13	OK	0.031	9011200	560	2332
14	OK	0.031	9154560	388	1823
15	OK	0.031	9113600	408	1884

16	OK	0.031	9150464	1042	3777
17	OK	0.046	9048064	1043	3785
18	OK	0.031	9060352	1044	3776
19	OK	0.046	9289728	5587	25514
20	OK	0.046	9252864	6733	28938
21	OK	0.046	9121792	4737	22961
22	OK	0.031	9334784	5685	25800
23	OK	0.046	9302016	10383	39969
24	OK	0.046	9224192	10421	40061
25	OK	0.046	9318400	10420	40058
26	OK	0.171	10682368	65880	305389
27	OK	0.171	10883072	77550	340377
28	OK	0.187	11046912	57488	280214
29	OK	0.171	10584064	68090	311998
30	OK	0.187	10735616	103872	420190
31	OK	0.218	10948608	103940	420367
32	OK	0.187	10862592	103842	420123
33	OK	1.562	24825856	758839	3554252
34	OK	1.656	24604672	875802	3905103
35	OK	1.609	24768512	675241	3303455
36	OK	1.640	24825856	782803	3626114
37	OK	1.718	25788416	1038992	4403249
38	OK	1.656	25292800	1038702	4402564
39	OK	1.671	25145344	1039245	4403714

Задача «Число инверсий»

Условие

Инверсией в последовательности чисел A называется такая ситуация, когда $i < j$, а $A_i > A_j$.

Дан массив целых чисел. Ваша задача — подсчитать число инверсий в нем.

Формат входного файла

В первой строке входного файла содержится число n ($1 \leq n \leq 10^5$) — число элементов в массиве. Во второй строке находятся n целых чисел, по модулю не превосходящих 10^9 .

Формат выходного файла

В выходной файл надо вывести число инверсий в массиве.

Решение

```
from bisect import bisect_left
```

```

try:
    inp = open('input.txt', 'r')
    outp = open('output.txt', 'w')

    input = inp.readline
    print = lambda *args: outp.write(' '.join(map(str, args)) + '\n')
except:
    pass

n = int(input())
a = list(map(int, input().split()))
cnt = 0

def merge(l1, l2):
    global cnt
    l, r = 0, 0
    m = []

    for i in range(len(l1) + len(l2)):
        if l < len(l1) and r < len(l2):
            if l1[l] > l2[r]:
                cnt += len(l1) - l
                m.append(l2[r])
                r += 1
            else:
                m.append(l1[l])
                l += 1
        elif r < len(l2):
            m.append(l2[r])
            r += 1
        else:
            m.append(l1[l])
            l += 1

    return m

def msort(a):
    if len(a) == 1:
        return a

    mp = len(a) // 2

    return merge(msort(a[:mp]), msort(a[mp:]))

msort(a)
print(cnt)

```

Результат

№ теста	Результат	Время, с	Память	Размер входного файла	Размер выходного файла
Max		1.343	22851584	1039245	12
1	OK	0.031	9134080	25	4
2	OK	0.031	8998912	6	3
3	OK	0.046	9056256	8	3
4	OK	0.031	9109504	8	3
5	OK	0.031	9052160	42	3
6	OK	0.031	9019392	43	4
7	OK	0.031	9175040	51	3
8	OK	0.031	9039872	45	4
9	OK	0.031	9093120	105	4
10	OK	0.031	9121792	110	4
11	OK	0.046	9097216	107	4
12	OK	0.062	9007104	461	3
13	OK	0.031	9121792	560	6
14	OK	0.031	9060352	388	3
15	OK	0.031	9019392	408	6
16	OK	0.031	9138176	1042	6
17	OK	0.031	9048064	1043	6
18	OK	0.031	9068544	1044	6
19	OK	0.046	9179136	5587	3
20	OK	0.031	9109504	6733	8
21	OK	0.046	9076736	4737	3
22	OK	0.031	9187328	5685	8
23	OK	0.046	9162752	10383	8
24	OK	0.046	9080832	10421	8
25	OK	0.046	9216000	10420	8
26	OK	0.125	10203136	65880	3
27	OK	0.140	10133504	77550	10
28	OK	0.125	10145792	57488	3
29	OK	0.140	10268672	68090	10
30	OK	0.140	10088448	103872	10
31	OK	0.156	10448896	103940	10

32	OK	0.156	10326016	103842	10
33	OK	1.046	21626880	758839	3
34	OK	1.312	22851584	875802	12
35	OK	1.015	21831680	675241	3
36	OK	1.281	21790720	782803	12
37	OK	1.296	21942272	1038992	12
38	OK	1.296	22323200	1038702	12
39	OK	1.343	22294528	1039245	12

Задача «Анти-quick sort»

Дан массив целых чисел. Ваша задача — отсортировать его в порядке неубывания с помощью сортировки вставками.

Сортировка вставками проходится по всем элементам массива от меньших индексов к большим («слева направо») для каждого элемента определяет его место в предшествующей ему отсортированной части массива и переносит его на это место (возможно, сдвигая некоторые элементы на один индекс вправо). Чтобы проконтролировать, что Вы используете именно сортировку вставками, мы попросим Вас для каждого элемента массива, после того, как он будет обработан, выводить его новый индекс.

Формат входного файла

В первой строке входного файла содержится число n ($1 \leq n \leq 1000$) — число элементов в массиве. Во второй строке находятся n различных целых чисел, по модулю не превосходящих 10^9 .

Формат выходного файла

В первой строке выходного файла выведите n чисел. При этом i -ое число равно индексу, на который, в момент обработки его сортировкой вставками, был перемещен i -ый элемент исходного массива. Индексы нумеруются, начиная с единицы. Между любыми двумя числами должен стоять ровно один пробел.

Во второй строке выходного файла выведите отсортированный массив. Между любыми двумя числами должен стоять ровно один пробел.

Решение

```
from bisect import bisect_left
from math import *

try:
    inp = open('input.txt', 'r')
    outp = open('output.txt', 'w')

    input = inp.readline
    print = lambda *args: outp.write(' '.join(map(str, args)) + '\n')
except:
    pass
```



```

n = int(input())
d = [i for i in range(n)]
res = [0] * n

for i in range(n - 1, -1, -1):
    res[d[i // 2]] = i + 1
    d[i // 2], d[i] = d[i], d[i // 2]

if n > 1:
    i1 = res.index(1)
    i2 = res.index(2)

    res[i1], res[i2] = res[i2], res[i1]

print(*res)

```

Результат

№ теста	Результат	Время, с	Память	Размер входного файла	Размер выходного файла
Max		0.078	9207808	10415	14298
1	OK	0.031	9027584	25	42
2	OK	0.046	8925184	7	7
3	OK	0.031	9048064	12	14
4	OK	0.078	8929280	8	10
5	OK	0.062	9048064	10	14
6	OK	0.031	9015296	29	33
7	OK	0.031	9039872	10	14
8	OK	0.046	8933376	10	14
9	OK	0.031	8953856	10	14
10	OK	0.031	9109504	10	14
11	OK	0.031	8962048	10	14
12	OK	0.046	8916992	57	65
13	OK	0.031	9064448	56	64
14	OK	0.031	9011200	57	65
15	OK	0.046	8921088	77	89
16	OK	0.031	9043968	76	88
17	OK	0.031	8962048	77	89
18	OK	0.046	8921088	112	129
19	OK	0.031	8990720	111	129
20	OK	0.031	9084928	110	127
21	OK	0.031	8957952	949	1192

22	OK	0.031	9003008	960	1221
23	OK	0.031	9072640	957	1136
24	OK	0.031	8925184	1490	1890
25	OK	0.046	9019392	1486	1946
26	OK	0.046	9076736	1481	1763
27	OK	0.046	8957952	3723	4890
28	OK	0.031	9064448	3729	5049
29	OK	0.078	9109504	3727	4439
30	OK	0.031	9084928	8456	11340
31	OK	0.031	9093120	8471	11611
32	OK	0.031	9162752	8415	10037
33	OK	0.046	9175040	10415	14037
34	OK	0.046	9207808	10410	14298
35	OK	0.046	9199616	10393	12388

Задача «К-ая порядковая статистика»

Условие

Дан массив из n элементов. Какие числа являются k_1 -ым, (k_1+1) -ым, ..., k_2 -ым в порядке неубывания в этом массиве?

Формат входного файла

В первой строке входного файла содержатся три числа: n — размер массива, а также границы интервала k_1 и k_2 , при этом $2 \leq n \leq 4 \cdot 10^7$, $1 \leq k_1 \leq k_2 \leq n$, $k_2 - k_1 < 200$.

Во второй строке находятся числа A , B , C , a_1 , a_2 , по модулю не превосходящие 10^9 . Вы должны получить элементы массива, начиная с третьего, по формуле: $a_i = A \cdot a_{i-2} + B \cdot a_{i-1} + C$. Все вычисления должны производиться в 32-битном знаковом типе, переполнения должны игнорироваться.

Формат выходного файла

В первой и единственной строке выходного файла выведите k_1 -ое, (k_1+1) -ое, ..., k_2 -ое в порядке неубывания числа в массиве a . Числа разделяйте одним пробелом.

Решение

```
using System;
using System.IO;
using System.Linq;

namespace Temp
{
    public static class Program
    {
        private static StreamReader sr;
        private static StreamWriter sw;
```

```

private static int[] a;
private static int n, k1, k2, A, B, C;

public static void Quicksort(int[] elements, int left, int
right)
{
    while (true)
    {
        if (left > k2 || right < k1) return;

        int i = left, j = right;
        int pivot = elements[(left + right) / 2];

        while (i ≤ j)
        {
            while (elements[i].CompareTo(pivot) < 0)
            {
                i++;
            }

            while (elements[j].CompareTo(pivot) > 0)
            {
                j--;
            }

            if (i > j) continue;

            int tmp = elements[i];
            elements[i] = elements[j];
            elements[j] = tmp;

            i++;
            j--;
        }

        if (left < j)
        {
            Quicksort(elements, left, j);
        }

        if (i < right)
        {
            left = i;
            continue;
        }

        break;
    }
}

public static void Main(string[] args)

```

```

    {
        sr = new StreamReader("input.txt");
        sw = new StreamWriter("output.txt", false);

        var inp =
sr.ReadLine().Split().Select(int.Parse).ToArray();
        n = inp[0];
        k1 = inp[1] - 1;
        k2 = inp[2] - 1;

        inp = sr.ReadLine().Split().Select(int.Parse).ToArray();

        A = inp[0];
        B = inp[1];
        C = inp[2];
        a = new int[n];
        a[0] = inp[3];
        a[1] = inp[4];
        sr.Close();

        for (var i = 2; i < n; i++)
        {
            a[i] = A * a[i - 2] + B * a[i - 1] + C;
        }

        Quicksort(a, 0, a.Length - 1);
        for (int i = k1; i ≤ k2; i++)
        {
            sw.Write($"{a[i]} ");
        }

        sw.Close();
    }
}

```

Результат

№ теста	Результат	Время, с	Память	Размер входного файла	Размер выходного файла
Max		0.796	171458560	54	2400
1	OK	0.031	11108352	18	6
2	OK	0.031	11067392	28	9
3	OK	0.046	11165696	32	4
4	OK	0.015	11108352	33	5
5	OK	0.031	11108352	32	10
6	OK	0.031	11071488	33	5
7	OK	0.031	11071488	32	19

8	OK	0.031	11096064	32	21
9	OK	0.046	11132928	25	300
10	OK	0.015	11087872	22	382
11	OK	0.031	11137024	23	477
12	OK	0.031	11132928	35	12
13	OK	0.031	11177984	38	11
14	OK	0.015	11145216	36	1074
15	OK	0.046	11161600	36	561
16	OK	0.031	11161600	37	220
17	OK	0.031	11173888	24	400
18	OK	0.031	11153408	28	1200
19	OK	0.031	11157504	29	1400
20	OK	0.046	11132928	37	12
21	OK	0.031	11177984	45	11
22	OK	0.031	11165696	38	2400
23	OK	0.031	11251712	39	2400
24	OK	0.031	11169792	44	2200
25	OK	0.031	11231232	43	2200
26	OK	0.031	11165696	41	676
27	OK	0.031	11563008	28	600
28	OK	0.031	11542528	31	1400
29	OK	0.031	11526144	32	1600
30	OK	0.031	11481088	37	12
31	OK	0.031	11497472	48	11
32	OK	0.031	11563008	40	2400
33	OK	0.031	11579392	40	2400
34	OK	0.031	11522048	47	2200
35	OK	0.031	11509760	46	2200
36	OK	0.031	11522048	45	200
37	OK	0.062	15130624	32	800
38	OK	0.046	15118336	34	1600
39	OK	0.046	15171584	35	1800
40	OK	0.031	15085568	38	12
41	OK	0.031	15089664	49	11
42	OK	0.046	15183872	40	2400

43	OK	0.031	15171584	40	2003
44	OK	0.031	15138816	49	2200
45	OK	0.046	15138816	47	2200
46	OK	0.031	15093760	48	560
47	OK	0.375	171425792	33	800
48	OK	0.390	171356160	39	2000
49	OK	0.421	171421696	40	2200
50	OK	0.562	171388928	40	12
51	OK	0.406	171417600	52	11
52	OK	0.484	171413504	42	2400
53	OK	0.531	171446272	42	2400
54	OK	0.515	171429888	54	2200
55	OK	0.625	171372544	54	2200
56	OK	0.796	171372544	52	1076
57	OK	0.593	171450368	53	2200
58	OK	0.640	171376640	52	2076
59	OK	0.500	171425792	54	2035
60	OK	0.437	171409408	53	1859
61	OK	0.718	171360256	51	2208
62	OK	0.390	171458560	49	2189
63	OK	0.484	171458560	53	2057
64	OK	0.656	171433984	54	1991
65	OK	0.625	171372544	50	2004
66	OK	0.656	171356160	52	1793
67	OK	0.375	171376640	54	1930

Задача «Сортировка пугалом»

Условие

«Сортировка пугалом» — это давно забытая народная потешка, которую восстановили по летописям специалисты платформы «Открытое образование» специально для этого курса.

Участнику под верхнюю одежду продевают деревянную палку, так что у него оказываются растопырены руки, как у огородного пугала. Перед ним ставятся n матрёшек в ряд. Из-за палки единственное, что он может сделать — это взять в руки две матрёшки на расстоянии k друг от друга (то есть i -ую и $(i+k)$ -ую), развернуться и поставить их обратно в ряд, таким образом поменяв их местами.

Задача участника — расположить матрёшки по убыванию размера. Может ли он это сделать?

Формат входного файла

В первой строчке содержатся числа n и k ($1 \leq n, k \leq 10^5$) — число матрёшек и размах рук.

Во второй строчке содержится n целых чисел, которые по модулю не превосходят 10^9 — размеры матришек.

Формат выходного файла

Выведите «YES», если возможно отсортировать матришки по неубыванию размера, и «NO» в противном случае.

Решение

```
try:
    inp = open('input.txt', 'r')
    outp = open('output.txt', 'w')

    input = inp.readline
    print = lambda *args: outp.write(' '.join(map(str, args)) + '\n')
except:
    pass

def arr(t=int):
    return list(map(t, input().split()))

n, k = arr()
array = arr()

arrays = [
    sorted(array[i] for i in range(j, n, k))
    for j in range(k)
]

def get_p(i):
    return i % k, i // k

res = True
for i in range(1, n):
    if arrays[i % k][i // k] < arrays[(i - 1) % k][(i - 1) // k]:
        res = False
        break

print('YES' if res else 'NO')
```

Результат

№ теста	Результат	Время, с	Память	Размер входного файла	Размер выходного файла
Max		0.359	25636864	1039313	5
1	OK	0.031	9007104	12	4
2	OK	0.046	9142272	16	5
3	OK	0.031	8978432	112	5
4	OK	0.046	9011200	111	4
5	OK	0.031	8957952	112	5
6	OK	0.031	9125888	112	4
7	OK	0.031	9007104	109	5
8	OK	0.031	8957952	112	4

9	OK	0.031	9129984	110	5
10	OK	0.031	9035776	111	4
11	OK	0.031	8957952	108	5
12	OK	0.031	9252864	11674	5
13	OK	0.031	9183232	11707	4
14	OK	0.031	9154560	11712	5
15	OK	0.046	9269248	11754	4
16	OK	0.031	9211904	11708	5
17	OK	0.031	9134080	11740	4
18	OK	0.031	9256960	11726	5
19	OK	0.031	9183232	11680	4
20	OK	0.031	9121792	11741	5
21	OK	0.046	10440704	128736	5
22	OK	0.046	10383360	128832	4
23	OK	0.046	10342400	128751	5
24	OK	0.078	10981376	128866	4
25	OK	0.062	10813440	128700	5
26	OK	0.046	10461184	128707	4
27	OK	0.031	10403840	128729	5
28	OK	0.046	10342400	128807	4
29	OK	0.062	10465280	128784	5
30	OK	0.218	22048768	1039313	5
31	OK	0.140	22032384	1038610	4
32	OK	0.203	22319104	1038875	5
33	OK	0.281	25513984	1038723	4
34	OK	0.359	25636864	1038749	5
35	OK	0.140	22323200	1038747	4
36	OK	0.203	22302720	1039043	5
37	OK	0.218	22335488	1039210	4
38	OK	0.281	22020096	1038967	5