

**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 №3

Неделя третья

Выполнил:

Жумиков Егор Олегович

Преподаватели:

Романов Алексей Андреевич

Волчек Дмитрий Геннадьевич

Оглавление

Задача «Сортировка целых чисел »	3
Условие.....	3
Формат входного файла.....	3
Формат выходного файла	3
Решение	3
Результат	4
Задача «Цифровая сортировка».....	6
Условие.....	6
Формат входного файла.....	6
Формат выходного файла	6
Решение	6
Результат	7

Задача «Сортировка целых чисел»

Условие

В этой задаче Вам нужно будет отсортировать много неотрицательных целых чисел.

Вам даны два массива, A и B , содержащие соответственно n и m элементов. Числа, которые нужно будет отсортировать, имеют вид $A_i \cdot B_j$, где $1 \leq i \leq n$ и $1 \leq j \leq m$. Иными словами, каждый элемент первого массива нужно умножить на каждый элемент второго массива.

Пусть из этих чисел получится отсортированная последовательность C длиной $n \cdot m$. Выведите сумму каждого десятого элемента этой последовательности (то есть, $C_1 + C_{11} + C_{21} + \dots$).

Формат входного файла

В первой строке содержатся числа n и m ($1 \leq n, m \leq 6000$) — размеры массивов. Во второй строке содержится n чисел — элементы массива A . Аналогично, в третьей строке содержится m чисел — элементы массива B . Элементы массива неотрицательны и не превосходят 40000.

Формат выходного файла

Выведите одно число — сумму каждого десятого элемента последовательности, полученной сортировкой попарных произведений элементов массивов A и B .

Решение

```
#include <iostream>
#include "edx-io.hpp"
```

```
void radix_sort(long *array, int n, long max_element) {
    auto *output = new long[n];
    auto *count = new long[256];

    for (int pow = 0; (1LL << pow) <= max_element; pow += 8) {
        memset(count, 0, sizeof(long) * 256);

        for (int i = 0; i < n; i++) {
            count[(array[i] >> pow) & 255]++;
        }

        for (int i = 1; i < 256; i++) {
            count[i] += count[i - 1];
        }

        for (int i = n - 1; i >= 0; i--) {
            output[count[(array[i] >> pow) & 255] - 1] = array[i];
            count[(array[i] >> pow) & 255]--;
        }

        for (int i = 0; i < n; i++) {
            array[i] = output[i];
        }
    }

    delete[] output;
```

```

        delete[] count;
    }

int main() {
    int m, n;
    long max_element = -1;
    io >> n >> m;
    const int cnt = m * n;

    auto
        *array = new long[n * m],
        *a = new long[n],
        *b = new long[m];

    for (int i = 0; i < n; i++) {
        io >> a[i];
    }

    for (int i = 0; i < m; i++) {
        io >> b[i];
    }

    for (int i = 0; i < n; i++) {
        for (int j = 0; j < m; j++) {
            long c = a[i] * b[j];
            array[i * m + j] = c;
            max_element = max_element > c ? max_element : c;
        }
    }

    radix_sort(array, cnt, max_element);

    long long sum = 0;
    for (int i = 0; i < n * m; i += 10) {
        sum += array[i];
    }

    io << sum;

    delete[] array;
    delete[] a;
    delete[] b;

    return 0;
}

```

Результат

№ теста	Результат	Время, с	Память	Размер входного файла	Размер выходного файла
Max		1.796	289935360	68699	16

1	OK	0.000	2224128	24	2
2	OK	0.000	2236416	34	1
3	OK	0.000	2236416	38	2
4	OK	0.015	2232320	106	10
5	OK	0.000	2220032	234	11
6	OK	0.015	2240512	698	11
7	OK	0.000	2240512	705	12
8	OK	0.000	2244608	586	12
9	OK	0.000	2228224	34325	12
10	OK	0.000	2273280	5769	12
11	OK	0.015	2269184	3498	12
12	OK	0.000	2232320	924	12
13	OK	0.000	2273280	3494	12
14	OK	0.015	2285568	5772	12
15	OK	0.015	2220032	34449	12
16	OK	0.015	2351104	34368	13
17	OK	0.015	2293760	4006	13
18	OK	0.000	2314240	2886	13
19	OK	0.000	2289664	4009	13
20	OK	0.000	2347008	34361	13
21	OK	0.031	6672384	34966	14
22	OK	0.031	6623232	9167	14
23	OK	0.031	6623232	9162	14
24	OK	0.031	6672384	34917	14
25	OK	0.312	49881088	39991	15
26	OK	0.312	51863552	28668	15
27	OK	0.296	49881088	40034	15
28	OK	0.875	145903616	51489	15
29	OK	0.953	145907712	51525	15

30	OK	1.781	289935360	68655	16
31	OK	1.796	289931264	68625	16
32	OK	1.765	289931264	68699	16

Задача «Цифровая сортировка»

Условие

Дано n строк, выведите их порядок после k фаз цифровой сортировки.

Формат входного файла

В первой строке входного файла содержатся числа n — число строк, m — их длина и k — число фаз цифровой сортировки ($1 \leq n \leq 10^6$, $1 \leq k \leq m \leq 10^6$, $n \cdot m \leq 5 \cdot 10^7$). Далее находится описание строк, **но в нетривиальном формате**. Так, i -ая строка ($1 \leq i \leq n$) записана в i -ых символах второй, ..., $(m + 1)$ -ой строк входного файла. Иными словами, строки написаны по вертикали. **Это сделано специально, чтобы сортировка занимала меньше времени.**

Строки состоят из строчных латинских букв: от символа «a» до символа «z» включительно. В таблице символов ASCII все эти буквы располагаются подряд и в алфавитном порядке, код буквы «a» равен 97, код буквы «z» равен 122.

Формат выходного файла

В выходной файл надо вывести число инверсий в массиве.

Решение

```
#include <iostream>
#include <string>

#ifdef LOCAL

#define cin std::cin
#define cout std::cout

#else

#include "edx-io.hpp"
#define cin io
#define cout io

#endif

int main() {
    int n, m, k;
    cin >> n >> m >> k;

    auto *strings = new std::string[m];
    auto
        *tmp1 = new long[n],
```

```

        *tmp2 = new long[n];

    for (int i = 0; i < n; i++) {
        tmp1[i] = i;
    }

    for (int i = 0; i < m; i++) {
        cin >> strings[i];
    }

#define DV 200
    auto *counter = new long[DV];
    long *ptr[2] = { tmp1, tmp2 };
    int ptr_ind = 0;

    for (int pow = m - 1; pow ≥ m - k; --pow) {
#define CURR_CHAR (strings[pow][ptr[ptr_ind][i]])
        memset(counter, 0, sizeof(long) * DV);

        for (int i = 0; i < n; i++) {
            counter[CURR_CHAR]++;
        }

        for (int i = 1; i < DV; i++) {
            counter[i] += counter[i - 1];
        }

        for (int i = n - 1; i ≥ 0; i--) {
            ptr[1 - ptr_ind][--counter[CURR_CHAR]] =
ptr[ptr_ind][i];
        }

        ptr_ind = 1 & ++ptr_ind;
    }

    for (int i = 0; i < n; i++) {
        cout << (ptr[ptr_ind][i] + 1);
        cout << ' ';
    }

    return 0;
}

```

Результат

№ теста	Результат	Время, с	Память	Размер входного файла	Размер выходного файла
Max		1.828	166232064	52000020	6888896
1	OK	0.000	2232320	22	6
2	OK	0.000	2224128	22	6

3	OK	0.000	2220032	22	6
4	OK	0.015	2236416	10	2
5	OK	0.000	2224128	11	4
6	OK	0.000	2224128	130	21
7	OK	0.000	2224128	129	21
8	OK	0.015	2224128	129	21
9	OK	0.000	2236416	129	21
10	OK	0.015	2224128	129	21
11	OK	0.000	2220032	230	51
12	OK	0.000	2224128	229	51
13	OK	0.000	2232320	229	51
14	OK	0.031	2224128	229	51
15	OK	0.000	2236416	229	51
16	OK	0.000	2232320	450	51
17	OK	0.000	2232320	449	51
18	OK	0.000	2228224	450	51
19	OK	0.015	2228224	449	51
20	OK	0.015	2244608	449	51
21	OK	0.000	2244608	530	141
22	OK	0.000	2232320	529	141
23	OK	0.000	2232320	529	141
24	OK	0.000	2232320	529	141
25	OK	0.015	2244608	529	141
26	OK	0.015	2232320	1212	21
27	OK	0.000	2232320	1210	21
28	OK	0.031	2260992	1211	21

29	OK	0.000	2248704	1211	21
30	OK	0.015	2244608	1211	21
31	OK	0.000	2228224	2031	692
32	OK	0.015	2232320	2030	692
33	OK	0.015	2232320	2030	692
34	OK	0.000	2232320	2030	692
35	OK	0.000	2244608	2030	692
36	OK	0.015	2232320	2610	141
37	OK	0.000	2236416	2609	141
38	OK	0.000	2232320	2610	141
39	OK	0.000	2232320	2610	141
40	OK	0.015	2232320	2609	141
41	OK	0.015	2252800	4051	692
42	OK	0.015	2252800	4050	692
43	OK	0.000	2232320	4051	692
44	OK	0.000	2228224	4051	692
45	OK	0.015	2252800	4051	692
46	OK	0.000	2240512	6012	21
47	OK	0.000	2240512	6010	21
48	OK	0.000	2240512	6012	21
49	OK	0.000	2236416	6012	21
50	OK	0.000	2240512	6010	21
51	OK	0.015	2240512	10213	292
52	OK	0.015	2244608	10211	292
53	OK	0.000	2244608	10212	292
54	OK	0.000	2244608	10212	292

55	OK	0.000	2256896	10212	292
56	OK	0.015	2265088	20052	3893
57	OK	0.000	2285568	20051	3893
58	OK	0.000	2265088	20052	3893
59	OK	0.000	2285568	20052	3893
60	OK	0.031	2265088	20051	3893
61	OK	0.000	2322432	26012	141
62	OK	0.000	2310144	26010	141
63	OK	0.000	2310144	26012	141
64	OK	0.000	2322432	26011	141
65	OK	0.000	2322432	26012	141
66	OK	0.000	2293760	40413	692
67	OK	0.000	2293760	40411	692
68	OK	0.015	2281472	40413	692
69	OK	0.000	2281472	40412	692
70	OK	0.000	2289664	40413	692
71	OK	0.000	2347008	52014	141
72	OK	0.000	2351104	52011	141
73	OK	0.000	2351104	52013	141
74	OK	0.015	2363392	52013	141
75	OK	0.000	2367488	52013	141
76	OK	0.000	2404352	102015	292
77	OK	0.000	2408448	102012	292
78	OK	0.015	2408448	102014	292
79	OK	0.015	2404352	102014	292
80	OK	0.000	2420736	102014	292

81	OK	0.000	2596864	200033	108894
82	OK	0.000	2588672	200032	108894
83	OK	0.015	2600960	200032	108894
84	OK	0.015	2600960	200032	108894
85	OK	0.000	2600960	200032	108894
86	OK	0.015	2543616	250112	23893
87	OK	0.000	2547712	250111	23893
88	OK	0.000	2547712	250112	23893
89	OK	0.000	2547712	250111	23893
90	OK	0.015	2543616	250112	23893
91	OK	0.015	2785280	400053	108894
92	OK	0.015	2785280	400052	108894
93	OK	0.015	2805760	400053	108894
94	OK	0.015	2805760	400053	108894
95	OK	0.015	2801664	400053	108894
96	OK	0.000	2871296	501014	3893
97	OK	0.000	2867200	501012	3893
98	OK	0.015	2871296	501014	3893
99	OK	0.000	2871296	501014	3893
100	OK	0.000	2871296	501013	3893
101	OK	0.015	3936256	1000414	23893
102	OK	0.000	3936256	1000412	23893
103	OK	0.015	3936256	1000414	23893
104	OK	0.000	3932160	1000413	23893
105	OK	0.015	3940352	1000414	23893
106	OK	0.109	10612736	2400018	21

107	OK	0.015	10612736	2400013	21
108	OK	0.109	10608640	2400018	21
109	OK	0.093	10612736	2400018	21
110	OK	0.093	10608640	2400018	21
111	OK	0.031	7237632	2500113	288894
112	OK	0.015	7237632	2500112	288894
113	OK	0.046	7237632	2500113	288894
114	OK	0.015	7237632	2500112	288894
115	OK	0.031	7237632	2500113	288894
116	OK	0.062	10309632	4004016	8893
117	OK	0.015	10305536	4004013	8893
118	OK	0.046	10305536	4004016	8893
119	OK	0.031	10309632	4004015	8893
120	OK	0.046	10305536	4004016	8893
121	OK	0.062	12247040	5000215	288894
122	OK	0.031	12247040	5000213	288894
123	OK	0.062	12247040	5000214	288894
124	OK	0.031	12247040	5000214	288894
125	OK	0.062	12247040	5000214	288894
126	OK	0.156	22654976	10000216	588895
127	OK	0.031	22654976	10000214	588895
128	OK	0.171	22654976	10000215	588895
129	OK	0.046	22654976	10000215	588895
130	OK	0.062	22654976	10000215	588895
131	OK	0.390	43454464	20000216	1288895
132	OK	0.078	43450368	20000214	1288895

133	OK	0.390	43454464	20000215	1288895
134	OK	0.234	43454464	20000215	1288895
135	OK	0.281	43454464	20000215	1288895
136	OK	0.312	52289536	25001015	288894
137	OK	0.062	52289536	25001013	288894
138	OK	0.312	52293632	25001015	288894
139	OK	0.218	52289536	25001015	288894
140	OK	0.140	52293632	25001015	288894
141	OK	0.546	84045824	26000018	141
142	OK	0.140	84049920	26000013	141
143	OK	0.546	84045824	26000018	141
144	OK	0.546	84049920	26000018	141
145	OK	0.390	84049920	26000018	141
146	OK	0.312	55074816	25100017	1892
147	OK	0.062	55074816	25100013	1892
148	OK	0.312	55074816	25100017	1892
149	OK	0.187	55074816	25100017	1892
150	OK	0.109	55074816	25100016	1892
151	OK	0.296	53002240	25010016	23893
152	OK	0.046	53002240	25010013	23893
153	OK	0.281	53002240	25010016	23893
154	OK	0.093	53006336	25010015	23893
155	OK	0.250	53002240	25010016	23893
156	OK	0.656	55861248	25000114	3388895
157	OK	0.156	55861248	25000113	3388895
158	OK	0.640	55861248	25000114	3388895

159	OK	0.312	55861248	25000114	3388895
160	OK	0.156	55857152	25000113	3388895
161	OK	0.453	84230144	40040018	8893
162	OK	0.078	84226048	40040014	8893
163	OK	0.453	84230144	40040018	8893
164	OK	0.250	84230144	40040017	8893
165	OK	0.359	84230144	40040018	8893
166	OK	0.546	93745152	40400019	692
167	OK	0.093	93741056	40400014	692
168	OK	0.562	93745152	40400019	692
169	OK	0.296	93745152	40400018	692
170	OK	0.109	93745152	40400016	692
171	OK	0.468	82194432	40004017	108894
172	OK	0.078	82194432	40004014	108894
173	OK	0.453	82194432	40004017	108894
174	OK	0.234	82198528	40004016	108894
175	OK	0.312	82198528	40004017	108894
176	OK	0.796	83468288	40000416	1288895
177	OK	0.125	83468288	40000414	1288895
178	OK	0.734	83464192	40000416	1288895
179	OK	0.203	83468288	40000415	1288895
180	OK	0.140	83468288	40000414	1288895
181	OK	0.812	133132288	51000019	292
182	OK	0.156	133132288	51000014	292
183	OK	0.812	133136384	51000019	292
184	OK	0.203	133132288	51000018	292

185	OK	0.609	133132288	51000019	292
186	OK	0.578	105033728	50100018	3893
187	OK	0.109	105033728	50100014	3893
188	OK	0.578	105033728	50100018	3893
189	OK	0.234	105033728	50100018	3893
190	OK	0.359	105033728	50100018	3893
191	OK	1.828	109875200	50000115	6888896
192	OK	0.296	109879296	50000114	6888896
193	OK	1.828	109879296	50000115	6888896
194	OK	1.140	109879296	50000115	6888896
195	OK	1.484	109879296	50000115	6888896
196	OK	0.593	108425216	50200019	1892
197	OK	0.125	108425216	50200014	1892
198	OK	0.593	108425216	50200018	1892
199	OK	0.390	108425216	50200018	1892
200	OK	0.484	108425216	50200018	1892
201	OK	0.765	102715392	50001016	588895
202	OK	0.125	102715392	50001014	588895
203	OK	0.750	102715392	50001016	588895
204	OK	0.390	102707200	50001016	588895
205	OK	0.171	102715392	50001015	588895
206	OK	0.609	102346752	50002017	288894
207	OK	0.109	102346752	50002014	288894
208	OK	0.609	102346752	50002016	288894
209	OK	0.375	102346752	50002016	288894
210	OK	0.609	102346752	50002016	288894

211	OK	1.281	105877504	50000216	3388895
212	OK	0.203	105877504	50000214	3388895
213	OK	1.234	105877504	50000215	3388895
214	OK	1.171	105873408	50000215	3388895
215	OK	0.875	105877504	50000215	3388895
216	OK	1.156	166232064	52000020	141
217	OK	0.234	166232064	52000014	141
218	OK	1.093	166232064	52000019	141
219	OK	1.015	166232064	52000019	141
220	OK	0.265	166232064	52000018	141
221	OK	0.593	103751680	50010017	48894
222	OK	0.109	103763968	50010014	48894
223	OK	0.578	103755776	50010017	48894
224	OK	0.234	103755776	50010017	48894
225	OK	0.218	103759872	50010017	48894
226	OK	0.562	103890944	50020018	23893
227	OK	0.093	103890944	50020014	23893
228	OK	0.562	103895040	50020017	23893
229	OK	0.484	103890944	50020017	23893
230	OK	0.484	103890944	50020017	23893