Санкт-Петербургский национальный исследовательский университет
информационных технологий, механики и оптики

Кафедра информатики и прикладной математики

Лабораторная работа №3
Дисциплина «Алгоритмы и структуры данных»

Выполнил:
Молодецкий Арсений Алексеевич
Группа P3217

Санкт-Петербург
2018

Задание №1:

Код:

```
#include <iostream>
#include <fstream>
#include <algorithm>

using namespace std;


int main() {

	ifstream in;
	in.open("input.txt");
	int n;
	in >> n;
	int m;
	in >> m;
	auto firstArr = new int[n];
	for (int i = 0; i < n; ++i) {
		in >> firstArr[i];
	}

	auto secondArr = new int[m];
	for (int i = 0; i < m; ++i) {
		in >> secondArr[i];
	}

	sort(firstArr, firstArr + n);
	sort(secondArr, secondArr + m);
	in.close();
	auto *arr = new int[m*n];
	int k = 0;
	for (int i = 0; i < n; ++i) {
		for (int j = 0; j < m; ++j) {
			arr[k] = firstArr[i] * secondArr[j];
			k++;
		}
	}
	int count = m*n;
	cout << endl;
	int d = 8, w = 32;
	int * c = new int[256];
	int * b = new int[count];
	for (int p = 0; p < 4; p++) {
		for(int i = 0; i <256; ++i) c[i] = 0;
		for (int i = 0; i < count; ++i) {b[i] = 0;}

		for (int i = 0; i < count; i++)
			c[(arr[i] >> d*p)&((1<<d)-1)]++;
		for (int i = 1; i < 1<<d; i++)
			c[i] += c[i-1];
		for (int i = count-1; i >= 0; i--)
			b[--c[(arr[i] >> d*p)&((1<<d)-1)]] = arr[i];

		for (int i = 0; i < count; ++i) {
			arr[i] = b[i];
```

```
                }
        }

        ofstream out;
        out.open ("output.txt");
        long long sum = 0;
        for (int i = 0; i < count; i+= 10) {
                sum += arr[i];
        }
        out << sum;
        out.close();

        return 0;
}
```

№ теста	Результат	Время, с	Память	Размер входного файла	Размер выходного файла
Max		1.640	291692544	68699	16
1	OK	0.031	3616768	24	2
2	OK	0.000	3612672	34	1
3	OK	0.015	3624960	38	2
4	OK	0.000	3596288	106	10
5	OK	0.000	3629056	234	11
6	OK	0.015	3641344	698	11
7	OK	0.000	3641344	705	12
8	OK	0.000	3641344	586	12
9	OK	0.015	3682304	34325	12
10	OK	0.000	3670016	5769	12
11	OK	0.000	3657728	3498	12
12	OK	0.000	3665920	924	12
13	OK	0.000	3661824	3494	12
14	OK	0.046	3649536	5772	12
15	OK	0.015	3698688	34449	12
16	OK	0.015	4132864	34368	13
17	OK	0.046	4112384	4006	13
18	OK	0.015	4104192	2886	13
19	OK	0.015	4091904	4009	13
20	OK	0.015	4145152	34361	13
21	OK	0.015	8450048	34966	14
22	OK	0.031	8417280	9167	14
23	OK	0.015	8413184	9162	14
24	OK	0.031	8445952	34917	14
25	OK	0.281	51646464	39991	15
26	OK	0.281	53653504	28668	15
27	OK	0.265	51654656	40034	15
28	OK	0.812	147673088	51489	15
29	OK	0.812	147677184	51525	15
30	OK	1.640	291667968	68655	16
31	OK	1.625	291692544	68625	16
32	OK	1.625	291676160	68699	16

Задание №2:

Код:

```
#include <iostream>
#include <fstream>
#include <algorithm>

using namespace std;
```

```
int main() {

    ifstream in;
    in.open("input.txt");
    int n;
    in >> n;
    int m;
    in >> m;
    int k;
    in >> k;

    auto ** matrix = new char*[m];
    auto ** tmpMatrix = new char*[m];
    for (int i = 0; i < m; ++i) {
        matrix[i] = new char[n];
        tmpMatrix[i] = new char[n];
    }

    for (int i = 0; i < m; ++i) {
        for (int j = 0; j < n; ++j) {
            in >> matrix[i][j];
            tmpMatrix[i][j] = matrix[i][j];
        }
    }
    in.close();


    int* c = new int[123];

    int* indexes = new int[n];
    int* tmpIndexes = new int[n];
    for (int i = 0; i < n; ++i) {
        indexes[i] = i;
        tmpIndexes[i] = i;
    }

    for (int i = 0; i < k; ++i) {

        for (char j = 97; j < 123; ++j) {
            c[j] = 0;
        }

        for (int j = 0; j < n; ++j) {
            c[matrix[m-i-1][j]]++;
        }

        for (int j = 98; j < 123; ++j) {
            c[j] += c[j-1];
        }

        for (int j = n-1; j >= 0; j--){

            int indexReplace = c[matrix[m-i-1][j]] - 1;

            c[matrix[m-i-1][j]]--;

            for (int i2 = 0; i2 < m; ++i2) {
                tmpMatrix[i2][indexReplace] = matrix[i2][j];
```

```
            }

            tmpIndexes[indexReplace] = indexes[j];

        }

        for (int i2 = 0; i2 < m; ++i2) {
            for (int j2 = 0; j2 < n; ++j2) {
                matrix[i2][j2] = tmpMatrix[i2][j2];
            }
        }

        for (int i2 = 0; i2 < n; ++i2) {
            indexes[i2] = tmpIndexes[i2];
        }

    }

    ofstream out;
    out.open ("output.txt");
    for (int i = 0; i < n; ++i) {
        out << indexes[i] + 1 << ' ';
    }
    out.close();

    return 0;
}
```

№ теста	Результат	Время, с	Память	Размер входного файла	Размер выходного файла
Max		1.828	166232064	52000020	6888896
1	OK	0.015	2220032	22	6
2	OK	0.015	2236416	22	6
3	OK	0.000	2220032	22	6
4	OK	0.000	2224128	10	2
5	OK	0.000	2224128	11	4
6	OK	0.000	2236416	130	21
7	OK	0.015	2224128	129	21
8	OK	0.000	2224128	129	21
9	OK	0.015	2224128	129	21
10	OK	0.000	2224128	129	21
11	OK	0.000	2224128	230	51
12	OK	0.015	2232320	229	51
13	OK	0.015	2224128	229	51
14	OK	0.000	2224128	229	51
15	OK	0.000	2224128	229	51
16	OK	0.000	2236416	450	51
17	OK	0.000	2236416	449	51
18	OK	0.000	2232320	450	51
19	OK	0.000	2248704	449	51
20	OK	0.015	2240512	449	51
21	OK	0.015	2232320	530	141
22	OK	0.015	2232320	529	141
23	OK	0.000	2232320	529	141
24	OK	0.015	2244608	529	141
25	OK	0.000	2228224	529	141
26	OK	0.000	2232320	1212	21
27	OK	0.000	2240512	1210	21
28	OK	0.000	2232320	1211	21
29	OK	0.000	2232320	1211	21
30	OK	0.000	2244608	1211	21
31	OK	0.000	2232320	2031	692
32	OK	0.000	2232320	2030	692
33	OK	0.015	2244608	2030	692
34	OK	0.000	2232320	2030	692
35	OK	0.000	2228224	2030	692
36	OK	0.015	2236416	2610	141
37	OK	0.000	2236416	2609	141
38	OK	0.000	2228224	2610	141
39	OK	0.000	2228224	2610	141
40	OK	0.015	2244608	2609	141
41	OK	0.000	2240512	4051	692
42	OK	0.000	2240512	4050	692
43	OK	0.000	2248704	4051	692
44	OK	0.000	2240512	4051	692
45	OK	0.000	2252800	4051	692
46	OK	0.000	2252800	6012	21
47	OK	0.000	2240512	6010	21
48	OK	0.000	2240512	6012	21
49	OK	0.000	2252800	6012	21
50	OK	0.000	2236416	6010	21
51	OK	0.015	2240512	10213	292
52	OK	0.000	2256896	10211	292
53	OK	0.031	2240512	10212	292
54	OK	0.000	2256896	10212	292
55	OK	0.000	2240512	10212	292
56	OK	0.015	2285568	20052	3893
57	OK	0.000	2260992	20051	3893
58	OK	0.000	2265088	20052	3893
59	OK	0.000	2260992	20052	3893
60	OK	0.000	2285568	20051	3893
61	OK	0.000	2306048	26012	141
62	OK	0.000	2322432	26010	141
63	OK	0.000	2306048	26012	141
64	OK	0.000	2322432	26011	141
65	OK	0.015	2310144	26012	141
66	OK	0.000	2281472	40413	692
67	OK	0.015	2281472	40411	692
68	OK	0.015	2277376	40413	692
69	OK	0.015	2293760	40412	692
70	OK	0.046	2293760	40413	692
71	OK	0.000	2351104	52014	141
72	OK	0.015	2351104	52011	141
73	OK	0.000	2351104	52013	141
74	OK	0.000	2351104	52013	141
75	OK	0.000	2351104	52013	141
76	OK	0.000	2404352	102015	292
77	OK	0.031	2420736	102012	292
78	OK	0.015	2404352	102014	292
79	OK	0.015	2408448	102014	292
80	OK	0.015	2420736	102014	292
81	OK	0.015	2584576	200033	108894
82	OK	0.015	2588672	200032	108894
83	OK	0.015	2588672	200032	108894
84	OK	0.000	2600960	200032	108894
85	OK	0.000	2600960	200032	108894
86	OK	0.000	2547712	250112	23893
87	OK	0.000	2543616	250111	23893
88	OK	0.000	2560000	250112	23893
89	OK	0.015	2560000	250111	23893
90	OK	0.015	2547712	250112	23893
91	OK	0.015	2789376	400053	108894
92	OK	0.015	2789376	400052	108894
93	OK	0.015	2805760	400053	108894
94	OK	0.015	2801664	400053	108894
95	OK	0.000	2789376	400053	108894
96	OK	0.015	2871296	501014	3893
97	OK	0.000	2871296	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	0.031	3936256	1000414	23893
104	OK	0.000	3932160	1000413	23893
105	OK	0.015	3936256	1000414	23893
106	OK	0.109	10612736	2400018	21
107	OK	0.015	10612736	2400013	21

108	OK	0.109	10612736	2400018	21
109	OK	0.109	10612736	2400018	21
110	OK	0.109	10608640	2400018	21
111	OK	0.046	7237632	2500113	288894
112	OK	0.031	7237632	2500112	288894
113	OK	0.046	7237632	2500113	288894
114	OK	0.015	7237632	2500112	288894
115	OK	0.062	7237632	2500113	288894
116	OK	0.046	10313728	4004016	8893
117	OK	0.015	10305536	4004013	8893
118	OK	0.046	10313728	4004016	8893
119	OK	0.015	10309632	4004015	8893
120	OK	0.046	10309632	4004016	8893
121	OK	0.078	12242944	5000215	288894
122	OK	0.015	12247040	5000213	288894
123	OK	0.078	12247040	5000214	288894
124	OK	0.031	12247040	5000214	288894
125	OK	0.062	12247040	5000214	288894
126	OK	0.171	22654976	10000216	588895
127	OK	0.046	22654976	10000214	588895
128	OK	0.156	22650880	10000215	588895
129	OK	0.062	22654976	10000215	588895
130	OK	0.062	22654976	10000215	588895
131	OK	0.406	43454464	20000216	1288895
132	OK	0.078	43454464	20000214	1288895
133	OK	0.421	43454464	20000215	1288895
134	OK	0.218	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	52289536	25001015	288894
139	OK	0.203	52285440	25001015	288894
140	OK	0.156	52293632	25001015	288894
141	OK	0.546	84049920	26000018	141
142	OK	0.109	84045824	26000013	141
143	OK	0.546	84049920	26000018	141
144	OK	0.515	84049920	26000018	141
145	OK	0.406	84049920	26000018	141
146	OK	0.312	55070720	25100017	1892
147	OK	0.046	55074816	25100013	1892
148	OK	0.312	55074816	25100017	1892
149	OK	0.171	55074816	25100017	1892
150	OK	0.109	55074816	25100016	1892
151	OK	0.281	53002240	25010016	23893
152	OK	0.062	52998144	25010013	23893
153	OK	0.281	53006336	25010016	23893
154	OK	0.078	53006336	25010015	23893
155	OK	0.250	53006336	25010016	23893
156	OK	0.781	55861248	25000114	3388895
157	OK	0.140	55857152	25000113	3388895
158	OK	0.625	55861248	25000114	3388895
159	OK	0.312	55861248	25000114	3388895
160	OK	0.156	55861248	25000113	3388895
161	OK	0.468	84230144	40040018	8893
162	OK	0.078	84226048	40040014	8893
163	OK	0.453	84230144	40040018	8893
164	OK	0.218	84230144	40040017	8893
165	OK	0.375	84226048	40040018	8893
166	OK	0.562	93745152	40400019	692
167	OK	0.125	93741056	40400014	692
168	OK	0.546	93745152	40400019	692
169	OK	0.265	93745152	40400018	692
170	OK	0.109	93741056	40400016	692
171	OK	0.468	82194432	40004017	108894
172	OK	0.093	82194432	40004014	108894
173	OK	0.515	82194432	40004017	108894
174	OK	0.234	82194432	40004016	108894
175	OK	0.296	82194432	40004017	108894
176	OK	0.828	83468288	40000416	1288895
177	OK	0.109	83468288	40000414	1288895
178	OK	0.734	83468288	40000416	1288895
179	OK	0.218	83468288	40000415	1288895
180	OK	0.125	83468288	40000414	1288895
181	OK	0.828	133128192	51000019	292
182	OK	0.187	133132288	51000014	292
183	OK	0.828	133132288	51000019	292
184	OK	0.234	133132288	51000018	292
185	OK	0.656	133132288	51000019	292
186	OK	0.625	105033728	50100018	3893
187	OK	0.109	105029632	50100014	3893
188	OK	0.593	105033728	50100018	3893
189	OK	0.218	105033728	50100018	3893
190	OK	0.359	105033728	50100018	3893
191	OK	1.828	109879296	50000115	6888896
192	OK	0.312	109879296	50000114	6888896
193	OK	1.781	109879296	50000115	6888896
194	OK	1.140	109879296	50000115	6888896
195	OK	1.484	109879296	50000115	6888896
196	OK	0.609	108425216	50200019	1892
197	OK	0.125	108425216	50200014	1892
198	OK	0.609	108421120	50200018	1892
199	OK	0.437	108425216	50200018	1892
200	OK	0.468	108425216	50200018	1892
201	OK	0.765	102715392	50001016	588895
202	OK	0.125	102711296	50001014	588895
203	OK	0.734	102715392	50001016	588895
204	OK	0.406	102715392	50001016	588895
205	OK	0.187	102715392	50001015	588895
206	OK	0.625	102346752	50002017	288894
207	OK	0.109	102346752	50002014	288894
208	OK	0.609	102346752	50002016	288894
209	OK	0.359	102346752	50002016	288894
210	OK	0.609	102346752	50002016	288894
211	OK	1.328	105877504	50000216	3388895
212	OK	0.187	105873408	50000214	3388895
213	OK	1.296	105877504	50000215	3388895
214	OK	1.000	105877504	50000215	3388895
215	OK	0.781	105877504	50000215	3388895
216	OK	1.140	166232064	52000020	141
217	OK	0.218	166232064	52000014	141
218	OK	1.093	166227968	52000019	141
219	OK	1.015	166232064	52000019	141
220	OK	0.296	166232064	52000018	141
221	OK	0.578	103759872	50010017	48894
222	OK	0.093	103759872	50010014	48894
223	OK	0.562	103763968	50010017	48894
224	OK	0.250	103759872	50010017	48894
225	OK	0.203	103759872	50010017	48894
226	OK	0.562	103890944	50020018	23893
227	OK	0.093	103895040	50020014	23893
228	OK	0.546	103895040	50020017	23893
229	OK	0.531	103890944	50020017	23893
230	OK	0.484	103890944	50020017	23893