**Министерство образования и науки Российской Федерации**

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ,   
МЕХАНИКИ И ОПТИКИ**

Факультет программной инженерии и компьютерной техники

Кафедра информатики и прикладной математики   
Направление подготовки 09.03.04 Программная инженерия

Дисциплина «Алгоритмы и структуры данных»

**ОТЧЁТ**

по лабораторной работе №3

Студент Кочарян С.А. группы P3218

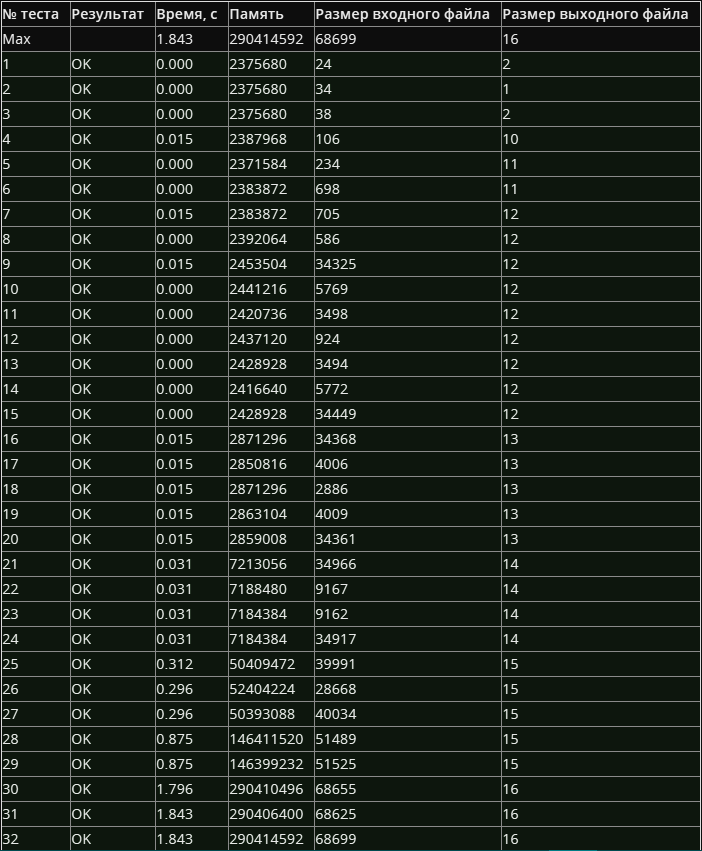
Преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2019

## Сортировка целых чисел

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

#include <fstream>  
using namespace std;  
#define NULL nullptr  
  
  
void radix\_sort(long \*array, int size, long maxNum) {  
 int digit = 1;  
 while (maxNum >> digit > 0) {  
 digit++;  
 }  
 digit--;  
 long \*count = new long[256];  
 long \*output = new long[size];  
 for (int pow = 0; pow <= digit; pow += 8) {  
 for (int i = 0; i < 256; i++) {  
 count[i] = 0;  
 }  
 for (int i = 0; i < size; i++) {  
 count[(array[i] >> pow) & 255]++;  
 }  
 for (int i = 1; i < 256; i++) {  
 count[i] += count[i - 1];  
 }  
 for (int i = size - 1; i >= 0; i--) {  
 output[count[(array[i] >> pow) & 255] - 1] = array[i];  
 count[(array[i] >> pow) & 255]--;  
 }  
 for (int i = 0; i < size; i++) {  
 array[i] = output[i];  
 }  
 }  
}  
  
int main() {  
 int firstLen, secondLen;  
  
 ifstream input("input.txt");  
 input >> firstLen >> secondLen;  
  
 long \*firstArray = new long[firstLen];   
 long \*finalArray = new long[firstLen \* secondLen];  
  
 long maxA = 0;  
 for (int i = 0; i < firstLen; i++) {  
 input >> firstArray[i];  
 if (firstArray[i] > maxA) {  
 maxA = firstArray[i];  
 }  
 }  
 long maxB = 0;   
 long a;  
  
 for (int i = 0; i < secondLen; i++) {  
 input >> a;   
 for (int j = 0; j < firstLen; j++) {   
 finalArray[(i \* firstLen) + j] = a \* firstArray[j];  
 }  
 if (a > maxB) {  
 maxB = a;  
 }  
 }  
 firstArray = NULL;   
 input.close();  
  
 long maxNum = maxA \* maxB;  
   
 radix\_sort(finalArray, firstLen\*secondLen, maxNum);  
   
 long long sum = 0;  
  
 for (int i = 0; i < firstLen \* secondLen ; i += 10) {  
 sum += finalArray[i];  
 }  
 ofstream output("output.txt");  
 output << sum;  
 output.close();  
  
 return 0;  
}



## Цифровая сортировка



#include <fstream>  
#include <string>   
#include "edx-io.hpp"   
using namespace std;  
  
  
int main() {   
 long n, m, k;  
 io >> n >> m >> k;  
  
 string \*A = new std::string[m];  
 for (long i = 0; i < m; i++) {  
 io >> A[i];  
 }  
  
 long \*Pos = new long[n];  
 long \*Pos2 = new long[n];   
 long \*Controller[2] = { Pos, Pos2 };   
 int pIn = 0;  
 for (long i = 0; i < n; i++) {  
 Pos[i] = i;  
 }  
  
 int \*count = new int[123];   
 for (int j = m - 1; m - j <= k; j--) {   
 for (int i = 97; i < 123; i++) {  
 count[i] = 0;  
 }  
  
 for (int i = 0; i < n; i++) {   
 count[A[j][Controller[pIn][i]]]++;  
 }  
  
 for (int i = 98; i < 124; i++) {  
 count[i] += count[i - 1];  
 }  
  
 for (int i = n - 1; i >= 0; i--) {   
 Controller[1 - pIn][--count[A[j][Controller[pIn][i]]]] = Controller[pIn][i];  
 }   
 pIn = ~pIn & 1;  
  
 }   
 ofstream output("output.txt");  
 for (long i = 0; i < n; i++) {  
 output << Controller[pIn][i] + 1 << ' ';  
 }  
 output.close();  
  
 return 0;  
}

