

Лабораторная работа №7

Эффективность алгоритмов

Задание 1 Тестирование программы

Задание 2 Вычислительные задачи

Задание 3 Числовые последовательности

Задача на «4»

Задача на «5»

Задание 1

Задача. Время выполнения программы

Программа ищет минимальное значение в массиве размером в 200000 элементов. Числа генерируются случайным образом.

Время выполнения программы зависит от того, насколько загружен компьютер и от того, какая у компьютера вычислительная мощность.

Скопировать предложенную ниже программу.

1. Провести многократное (10 - 15) тестирование программы на время выполнения в различных условиях.
2. Изменить загрузку компьютера.
3. Заменить генерацию вводом натуральных чисел, одинаковых значений.
4. Заполнить таблицу с условиями и результатами тестирования.

Задание 1

Задача. Время выполнения программы

// runtime.cpp: определяет точку входа для консольного приложения.

```
#include "stdafx.h"
```

```
#include <iostream>
```

```
#include <ctime>
```

```
using namespace std;
```

```
int main(int argc, char* argv[ ])
```

```
{
```

```
    srand(time(0));
```

```
    const int array_size = 200000; // размер массива
```

```
    int a[array_size]; // объявление массива
```

```
    for (int counter = 0; counter < array_size; counter++)
```



Задание 1

Задача. Время выполнения программы

```
{
a[counter] = rand() % 50 - rand() % 50; // заполнение массива в диапазоне
[-49;49]
    cout << a[counter] << " "; // печать элементов массива
}
int min = a[0]; // переменная для хранения минимального значения
for (int counter = 1; counter < array_size; counter++)
{
    if ( min > a[counter] ) // поиск минимального значения в массиве
        min = a[counter];
}
cout << "\nmin = " << min << endl;
cout << "runtime = " << clock()/1000.0 << endl; // время работы программы
system("pause");    return 0;
}
```

Пояснения к последующим заданиям

1. Реализовать контроль времени выполнения составленных программ (см. следующий слайд). В каждом задании указано предельное время работы программы.
2. Дать оценку времени работы алгоритма решаемых задач (трудоемкость линейная, квадратичная и т.д.).
3. Оценить сложность составленных программ по метрикам Холстеда, Джилба, стилистики и понятности программ.
4. Применить элементы защитного программирования. В отчете отметить, какие именно.

Время работы программы

// Заголовочный файл с прототипом функции clock()

```
#include <ctime>
```

```
// ...
```

```
    unsigned int start_time = clock(); // начальное время
```

<Код программы, время выполнения которой нужно измерить >

```
    unsigned int end_time = clock(); // конечное время
```

```
    unsigned int search_time = end_time - start_time; // искомое время
```

```
    return 0;
```

Прототип функции clock() находится в заголовочном файле <ctime>. Функция clock() возвращает значение времени в миллисекундах (1с = 1000млс).

Отсчёт времени начинается с момента запуска программы.

Задание 2

Вариант 1. Возведение в степень (0,6 с)

В данной задаче необходимо реализовать алгоритм возведения в степень.

Формат входного файла

Первая строка файла содержит целое число $N \leq 100000$ – количество чисел. Далее N строк содержат целое число и степень, в которую его надо возвести, разделенные пробелом. Гарантируется, что результат возведения в степень не превосходит 10^{18} .

Формат выходного файла

В выходной файл необходимо вывести N результатов возведения в степень, по одному на строке.

Примеры

<i>Ввод с консоли</i>	<i>На экране</i>
2	32
2 5	1728
12 3	

Задание 2

Вариант 2. Наибольший общий делитель (2 с)

Дан массив натуральных чисел. Найти НОД (наибольший общий делитель) этих чисел.

Формат входного файла

Первая строка файла содержит целое число $N \leq 1000$ – количество чисел. Во второй строке файла содержится N целых положительных чисел, все не более 1000000.

Формат выходного файла

В выходной файл необходимо вывести одно число – ответ на условие задачи.

Примеры

<i>Ввод с консоли</i>	<i>На экране</i>
5 42 91 21 70 119	7

Задание 3

Вариант 1. Числа Фибоначчи (2 с)

Числа Фибоначчи – это последовательность, задаваемая следующим образом:

$$A_0 = 1$$

$$A_1 = 1$$

$$A_{i+2} = A_i + A_{i+1}.$$

Начало последовательности: 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89...

Вам задан список чисел. Вывести только те из них, которые являются числами Фибоначчи. Выводить в том же порядке, в каком числа встречаются в исходном списке.

Формат входного файла

Первая строка входного файла содержит количество чисел в списке (не более 10000)

Вторая строка содержит список чисел через пробел, каждое число не больше 2 000 000 000.

Формат выходного файла

В выходной файл необходимо вывести ответ задачи.

Ввод с консоли	На экране
5 10 1 16 13 3	1 13 3

Задание 3

Вариант 2. Последовательность чисел (1 с)

Дано последовательность натуральных чисел. Необходимо определить минимальное число, отсутствующее в последовательности.

Формат входного файла

Первая строка файла содержит целое число $N \leq 100000$ – количество чисел в последовательности. Во второй строке файла содержится N целых положительных чисел, все не более 1000000.

Формат выходного файла

В выходной файл необходимо вывести одно число – ответ на условие задачи.

Примеры

<i>Ввод с консоли</i>	<i>На экране</i>
8 4 6 18 7 1 30 2 9	3

Задача на «4»

Частота встречаемости букв (2 с)

Дана строка, состоящая из маленьких латинских букв и пробелов. Определить, какие буквы встречается в тексте наиболее часто.

Формат входного файла

Входной файл содержит одну строку, состоящую только из маленьких латинских букв и пробелов. Длина строки не превосходит 10^6 .

Формат выходного файла

В выходной файл необходимо записать самые часто встречающиеся буквы без пробелов в алфавитном порядке.

Примеры

<i>Ввод с консоли</i>	<i>На экране</i>
mata mila ramu	am

Задача на «5»

Скобочная последовательность (2 с)

Задана последовательность круглых скобок. Определить, является ли эта последовательность правильной. Правильной скобочной последовательностью называется строка, состоящая только из символов "скобки". где для каждой закрывающей скобки найдётся соответствующая открывающая, и наоборот, для каждой открывающейся скобки - закрывающая. Ели говорить более формально, то:

- пустая строка — правильная скобочная последовательность;
- правильная скобочная последовательность, взятая в скобки одного типа — правильная скобочная последовательность;
- правильная скобочная последовательность, к которой приписана слева или справа правильная скобочная последовательность — тоже правильная скобочная последовательность.

Формат входного файла

Входные данные содержат одну строку - скобочную последовательность.

Формат выходного файла

Выведите YES, если последовательность правильная, и NO, если неправильная.

<i>Ввод с консоли</i>	<i>На экране</i>
<code>((()((()))))</code>	YES
<code>((()))(())</code>	NO