МИНИСТЕРСТВО науки и высшего ОБРАЗОВАНИЯ РОссИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ

(национальный исследовательский университет)»

Институт №3 «Системы управления, информатика и электроэнергетика»

Кафедра № 304 «Вычислительные машины, системы и сети»

Программирование

Отчет по лабораторной работе №1

Формирование массивов экспериментальных данных

Выполнили студенты группы M3О-211Б-21

Плоцкий Б.А.

Раужев Ю.М.

Проверила Дмитриева Е.А.

Москва 2022 г.

Оглавление

[Задание 3](#_Toc115019822)

[Общая блок-схема программы 4](#_Toc115019823)

[Структурные схемы алгоритмов 5](#_Toc115019824)

[Функция first\_task() 5](#_Toc115019825)

[Функция second\_task() 8](#_Toc115019826)

[Код программы 11](#_Toc115019827)

[Тестирование программы 12](#_Toc115019828)

[Тестирование некорректных режимов 12](#_Toc115019829)

[Тест 1 12](#_Toc115019830)

[Тестирование корректных режимов 13](#_Toc115019831)

[Тест 1 13](#_Toc115019832)

[Вывод 15](#_Toc115019833)

# Задание

**1 этап работы**

Разработать функции, формирующие последовательности чисел:

* упорядоченные по возрастанию,
* упорядоченные по убыванию,
* случайную последовательность,
* частично упорядоченные: «пилообразные», «синусоидальные», «ступенчатые».

для целых чисел и чисел с плавающей запятой (должно быть два комплекта функций и соответственно результатов) .

При реализации функций считать, что выделение памяти под последовательности происходит вне этих функций, функции в качестве формальных параметров получают указатель на массив, его размер и, возможно, **диапазон изменения величин и длину интервалов** (для частично упорядоченных последовательностей).

Рекомендуется использовать указатели на функцию в качестве формального параметра или массив указателей на функции для автоматизации сбора статистической информации.

Для массивов размерностью **от 150 до 200 элементов** (размерность выбирается самостоятельно) предусмотреть вывод значений в файл.

Используя данные из файлов, **построить графики** полученных зависимостей (возможно с привлечением других программных средств) для доказательства правильности полученных функций.

**2 этап работы**

1. Оценить длительность формирования последовательностей всех типов для различных значений размеров последовательностей (5⋅105, 10⋅105, …, 50⋅105), и на основе полученных значений построить графики зависимостей длительностей формирования массивов от их размера (лучше в виде столбчатых диаграмм, можно в Microsoft Exel).
2. Составить отчет, в котором привести структурные схемы алгоритмов главной функции и какой-то одной из функций (например, реализующей частично упорядоченную последовательность), текст программы, результаты по временным интервалам для различных размерностей массивов, графики зависимостей (времени от размерностей) и выводы по полученным результатам.

# Общая блок-схема программы



# Структурные схемы алгоритмов

## Функция first\_task()

1. Назначение:

Разработать функции, формирующие последовательности чисел:

* + упорядоченные по возрастанию,
  + упорядоченные по убыванию,
  + случайную последовательность,
  + частично упорядоченные: «пилообразные», «синусоидальные», «ступенчатые».

1. Прототип функции:

template<typename ARRAY\_TYPE, typename TYPE\_OF\_TIME = chrono::nanoseconds>

void first\_stage(

ARRAY\_TYPE\* arr,

int size,

double x\_step,

int func\_num,

TYPE\_OF\_TIME elapsed\_time

);

1. Обращение:

first\_task();

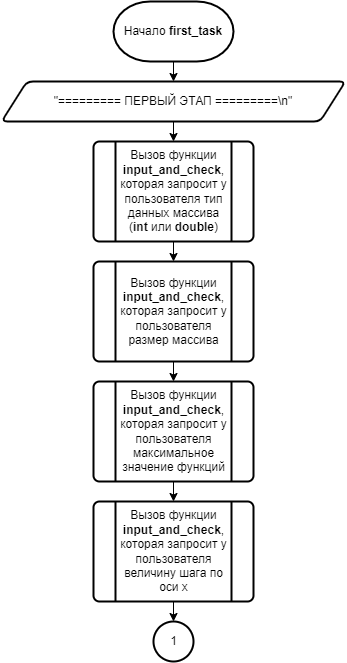
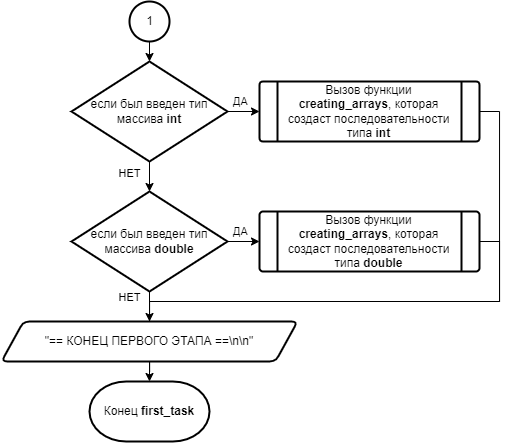
 

Рисунок 1. Структурная схема алгоритма функции first\_task().

## Функция second\_task()

1. Назначение:

Оценить длительность формирования последовательностей всех типов для различных значений размеров последовательностей (5⋅105, 10⋅105, …, 50⋅105)

1. Прототип функции:

template<typename ARRAY\_TYPE, typename TYPE\_OF\_TIME = chrono::microseconds>

void second\_stage(

ARRAY\_TYPE\* arr,

int size,

double x\_step,

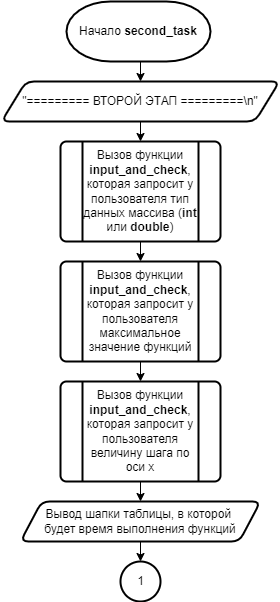
int func\_num,

TYPE\_OF\_TIME elapsed\_time

);

1. Обращение

second\_task();



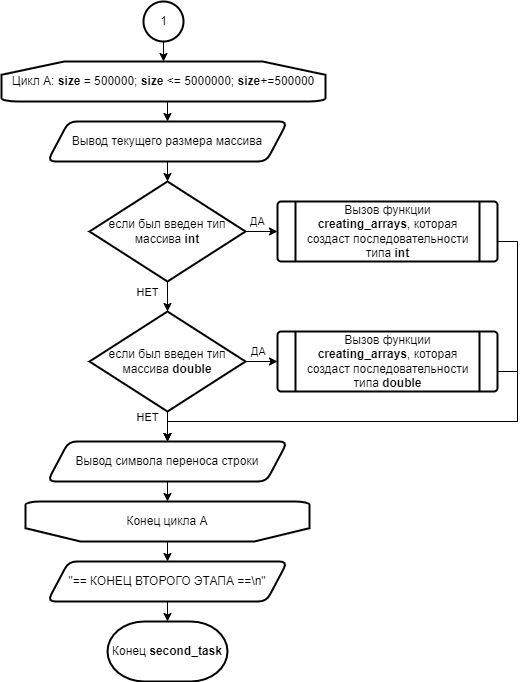


Рисунок 2. Структурная схема алгоритма функции second\_task().

# Код программы

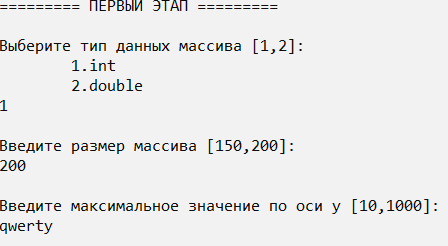
# Тестирование программы

## Тестирование некорректных режимов

### Тест 1

**Цель:** Проверить работу программы при введении некорректных данных

**Исходные данные:** 1 200 qwerty



**Ожидаемый Результат:** Вывод сообщения об ошибке:

Должно быть введено значение в интервале [10,1000].

**Полученный результат:**



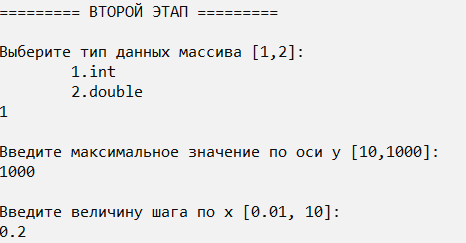
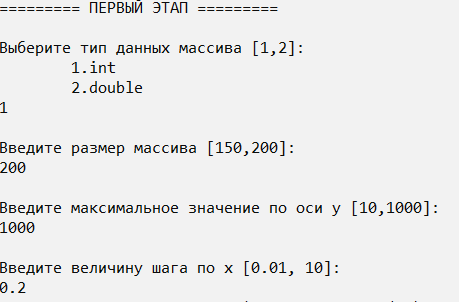
**Вывод по тесту:** Ожидаемый результат совпал с полученным результатом. Тест ошибок не обнаружил.

## Тестирование корректных режимов

### Тест 1

**Цель:** Проверить работу программы при выполнении всех условий задачи.

**Исходные данные:** 1 200 1000 0.2



**Ожидаемый Результат:**

Время выполнения 1 функции = 3000 (нс)

Время выполнения 2 функции = 600 (нс)

Время выполнения 3 функции = 8600 (нс)

Время выполнения 4 функции = 15600 (нс)

Время выполнения 5 функции = 14400 (нс)

Время выполнения 6 функции = 13400 (нс)

Длина последовательности |f1 мкС |f2 мкС |f3 мкС |f4 мкС |f5 мкС |f6 мкС |

500000| 1084 | 1139 | 20470 | 6346 | 13380 | 20500 |

1000000| 2164 | 2264 | 42607 | 12734 | 26839 | 41261 |

1500000| 3223 | 3563 | 62023 | 18923 | 41698 | 63468 |

2000000| 4603 | 4773 | 81696 | 25520 | 54821 | 82164 |

2500000| 5439 | 5696 |106791 | 31774 | 68604 |104894 |

3000000| 6526 | 7123 |128728 | 38289 | 87759 |128972 |

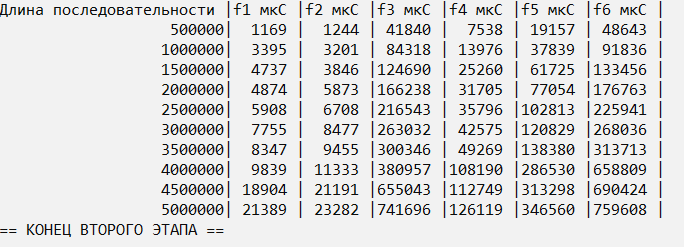
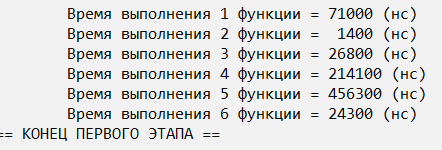
3500000| 7951 | 7924 |148033 | 44747 | 96866 |146140 |

4000000| 8813 | 9271 |163997 | 49858 |108219 |167511 |

4500000| 9957 | 10903 |186861 | 57289 |121730 |185694 |

5000000| 10926 | 11407 |205685 | 63566 |134516 |208095 |

**Полученный результат:**



**Вывод по тесту:** Ожидаемый результат совпал с полученным результатом. Тест ошибок не обнаружил.

# Вывод

Работа программы завершена на основании:

1) Полученные результаты совпали с ожидаемыми;

2) Считаем набор тестов полным.