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

Символьные данные

Выполнили студенты группы M3О-211Б-21

Плоцкий Б.А.

Раужев Ю.М.

Проверила Дмитриева Е.А.

Москва 2022 г.

Оглавление

[Задание 3](#_Toc99210030)

[Общая блок-схема программы 4](#_Toc99210031)

[Структурные схемы алгоритмов 5](#_Toc99210032)

[Функция read\_str() 5](#_Toc99210033)

[Функция first\_task() 7](#_Toc99210034)

[Функция second\_task() 9](#_Toc99210035)

[Код программы 12](#_Toc99210036)

[Тестирование программы 25](#_Toc99210037)

[Тестирование некорректных режимов 25](#_Toc99210038)

[Тест 1 25](#_Toc99210039)

[Тест 2 25](#_Toc99210040)

[Тест 3 25](#_Toc99210041)

[Тестирование корректных режимов 27](#_Toc99210042)

[Тест 1 27](#_Toc99210043)

[Тест 2 28](#_Toc99210044)

[Тест 3 29](#_Toc99210045)

[Тест 4 30](#_Toc99210046)

[Тест 5 31](#_Toc99210047)

[Тест 6 32](#_Toc99210048)

[Тест 7 33](#_Toc99210049)

[Тест 8 34](#_Toc99210050)

[Тест 9 35](#_Toc99210051)

[Тест 10 36](#_Toc99210052)

[Тест 11 37](#_Toc99210053)

[Тест 12 38](#_Toc99210054)

[Вывод 39](#_Toc99210055)

# 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();

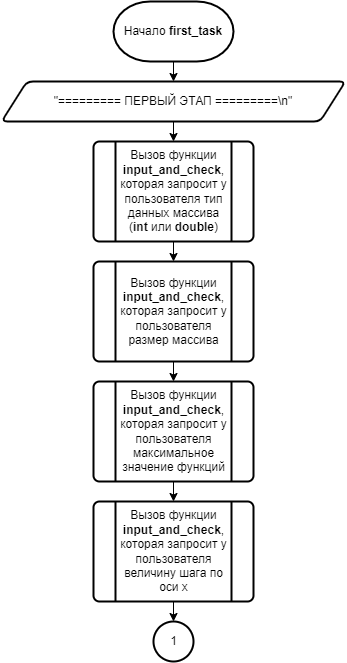
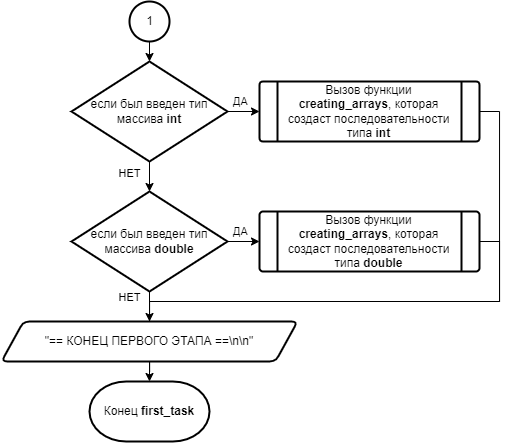
 

Рисунок 1. Структурная схема алгоритма функции first\_task().

## 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,

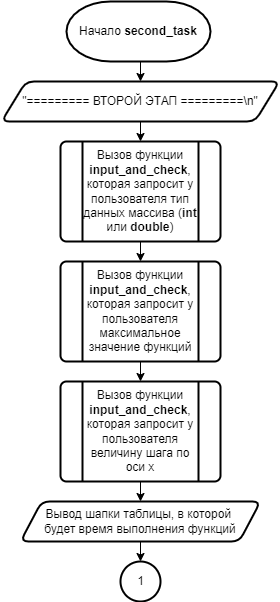
int func\_num,

TYPE\_OF\_TIME elapsed\_time

);

1. Обращение

second\_task();



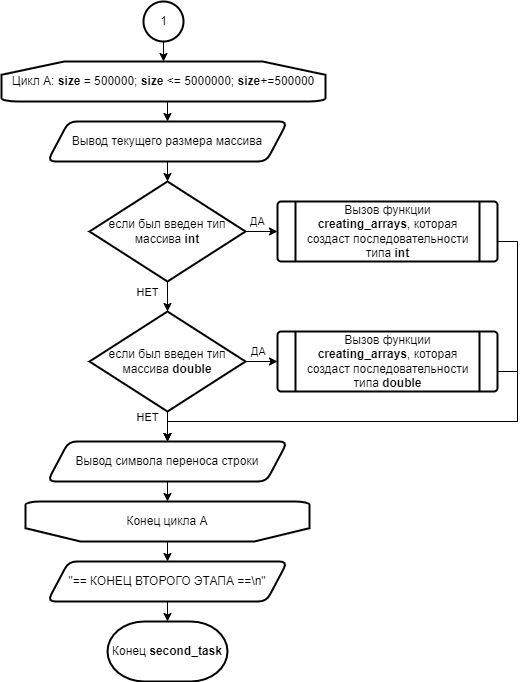


Рисунок 2. Структурная схема алгоритма функции second\_task().

# Кодпрограммы

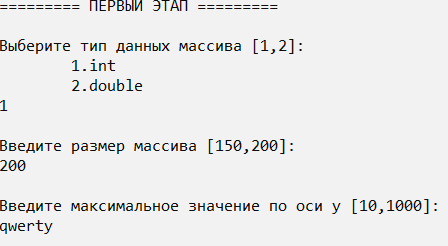
# Тестирование программы

## Тестирование некорректных режимов

### Тест 1

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

**Исходные данные:**



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

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

**Полученный результат:**



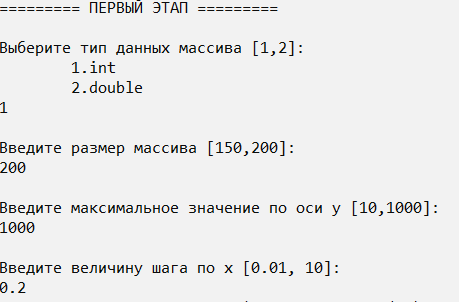
**Вывод по тесту:** Ожидаемый результат совпал с полученным результатом. Тест ошибок не обнаружил.

## Тестирование корректных режимов

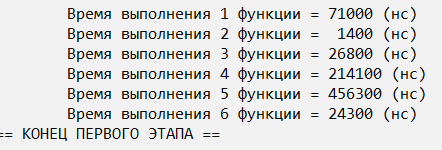
### Тест 1

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

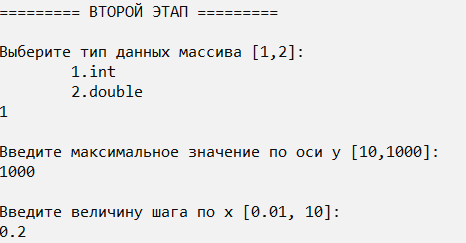
**Исходные данные:**



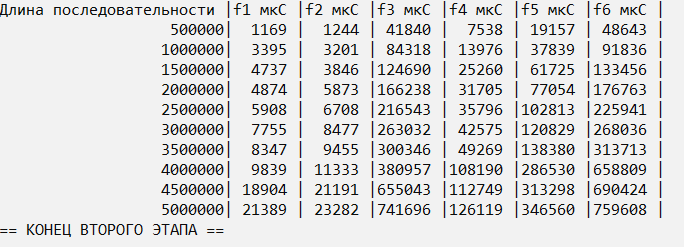
**Ожидаемый Результат:**



**Исходные данные:**



**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.