Homework

COMPUTING SYSTEMS ARCHITECTURE

Жулин Артем Германович | БПИ204 | 02.09.2021

Вариант – 131 – (5, 10)

1. Условие задачи.

Обобщенный артефакт, используемый в задании	Базовые альтернативы (уникальные параметры, Задающие отличительные признаки альтернатив)	Общие для всех альтернатив переменные	Общие для всех альтернатив функции
Квадратные матрицы с действительными числами	1. Обычный двумерный массив 2. Диагональная (на основе одномерного массива) 3. Нижняя треугольная матрица (одномерный массив с формулой пересчета)	Размерность – целое число	Вычисление среднего арифметического (действительное число)

2.Описание структуры ВС.

Название	Память
Int	4
Double	8
Char	1
Struct matrix: enum {MAX_LENGTH = 1000000} enum matrixKey {SQUARE, DIAGONAL, L_TRIANGLE} key: matrixKey dimension: int correct: bool	9 4 4 4[0] 4[4] 1[8]
Struct square: matrix array: double**	17 8 [9]
Struct lTriamgle: matrix array: double*	17 8[9]
Struct diagonal: matrix array: double*	17 8 [9]
Struct container enum {MAX_LEN = 10000} currentLenght: int cont: matrix*[MAX_LEN]	90004 4 4[0] 90000[4]
main (int argCount, char* argValues[]) argCount: int argValues: char*[] c: container length: int	90020 4[0] 8[4] 90004[12] 4[90016]

return 1..4	int
../matrix/matrix.cpp: In(ifstream &ifStream) command: char*[matrix::Max_LENGTH] key: int dimension: int temp: char*[matrix::Max_LENGTH] m_square: square m_diagonal: diagonal m_lTriangle: lTriangle return matrix*	 8000000[0] 4[8000004] 4[8000008] 8000000[8000012] 17[16000012] 17[16000012] 17[16000012] 8[16000029]
../matrix/matrix.cpp: InRandom() key: int dimension: int m_square: square m_diagonal: diagonal m_lTriangle: lTriangle return matrix*	 4[0] 4[4] 17[8] 17[8] 17[8] 8[25]
../matrix/diagonal.cpp: In m: diagonal temp: char*[matrix::Max_LENGTH] command: char*[matrix::Max_LENGTH]	 8[0] 8000000[8] 8000000[8000008]

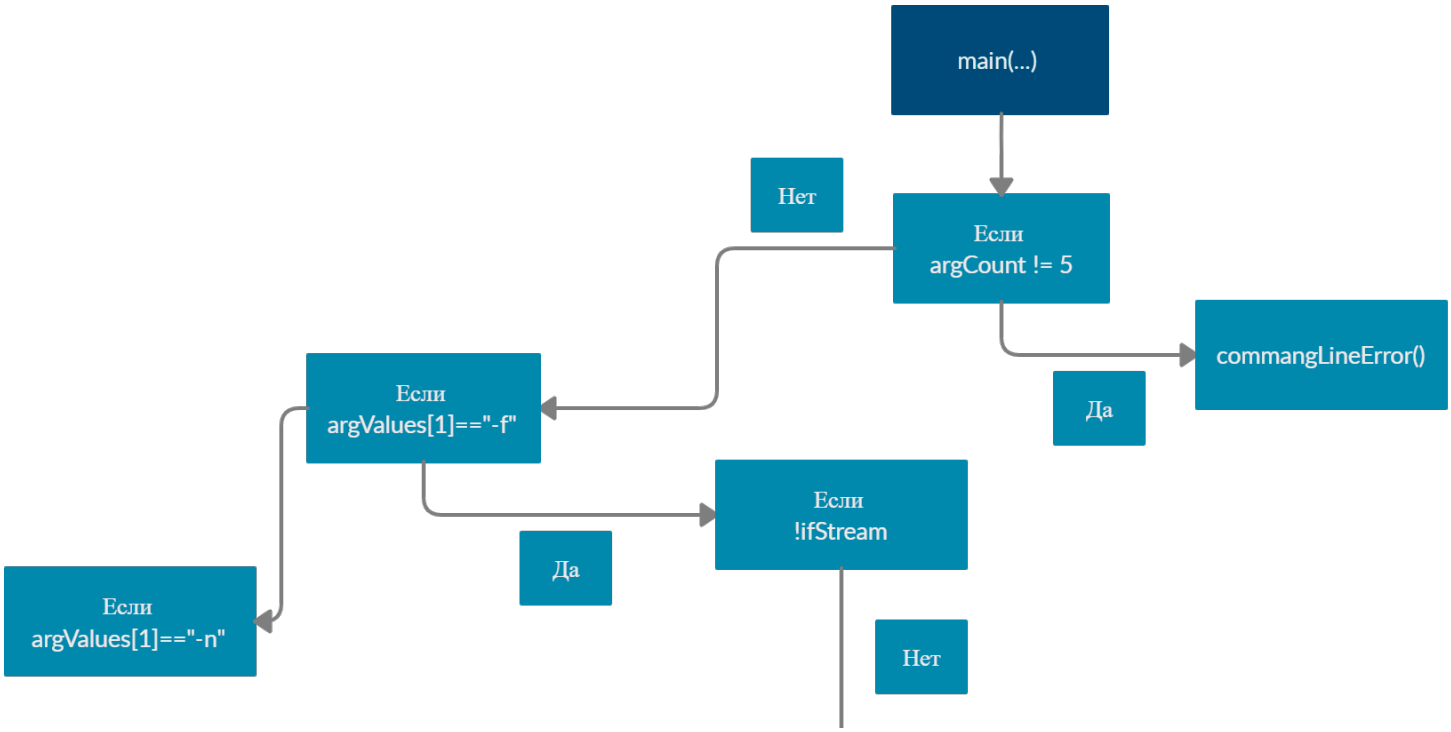


Рис. 1 Начало работы

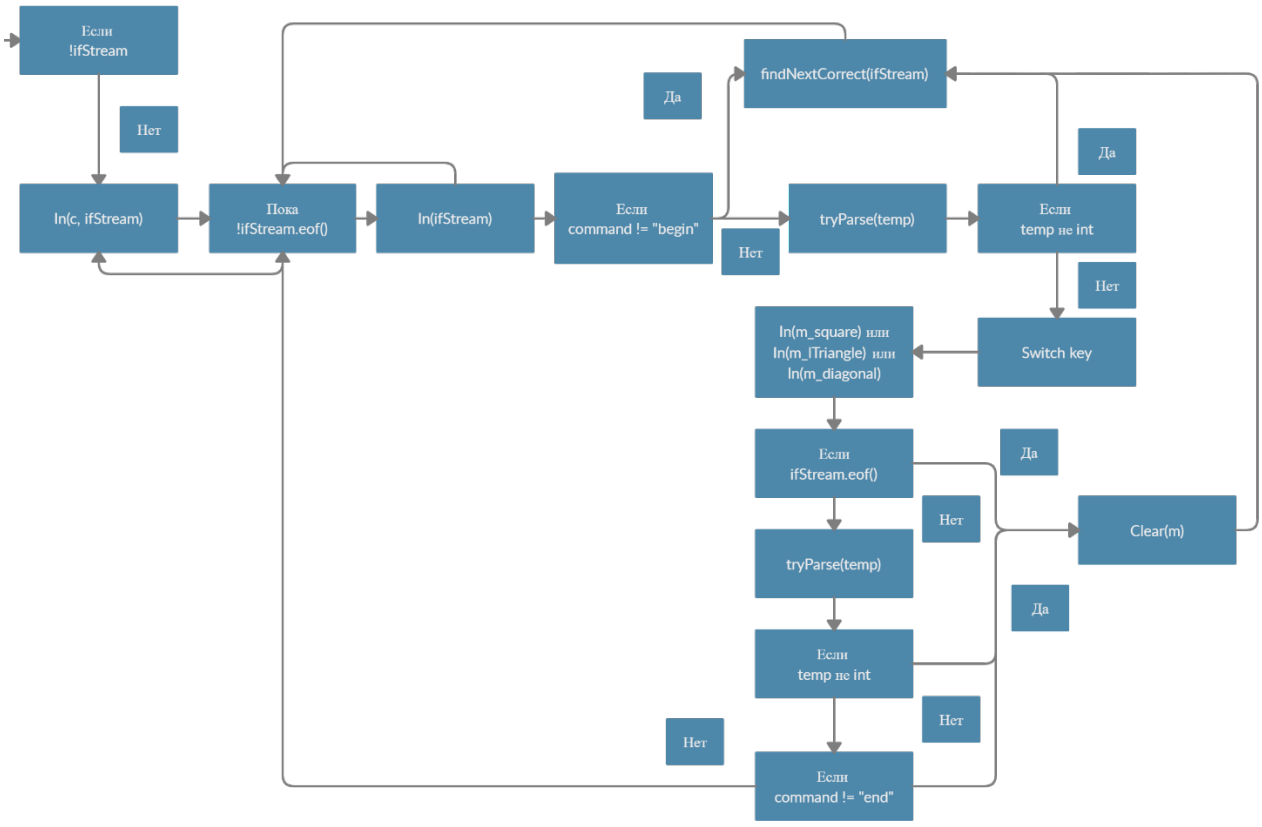


Рис. 2 Блок выполнения ввода данных из файла

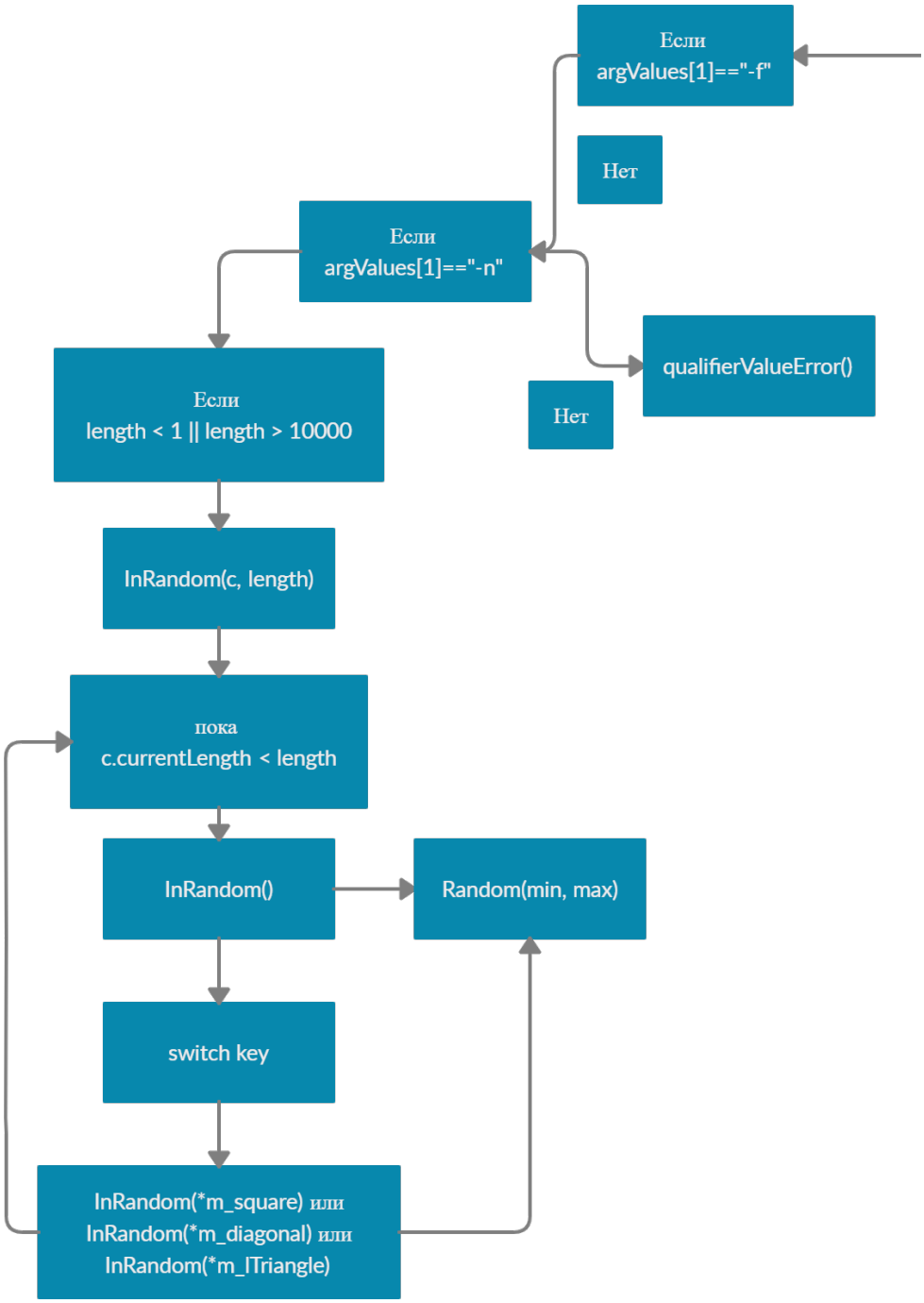


Рис. 3 Блок ввода данных случайным образом

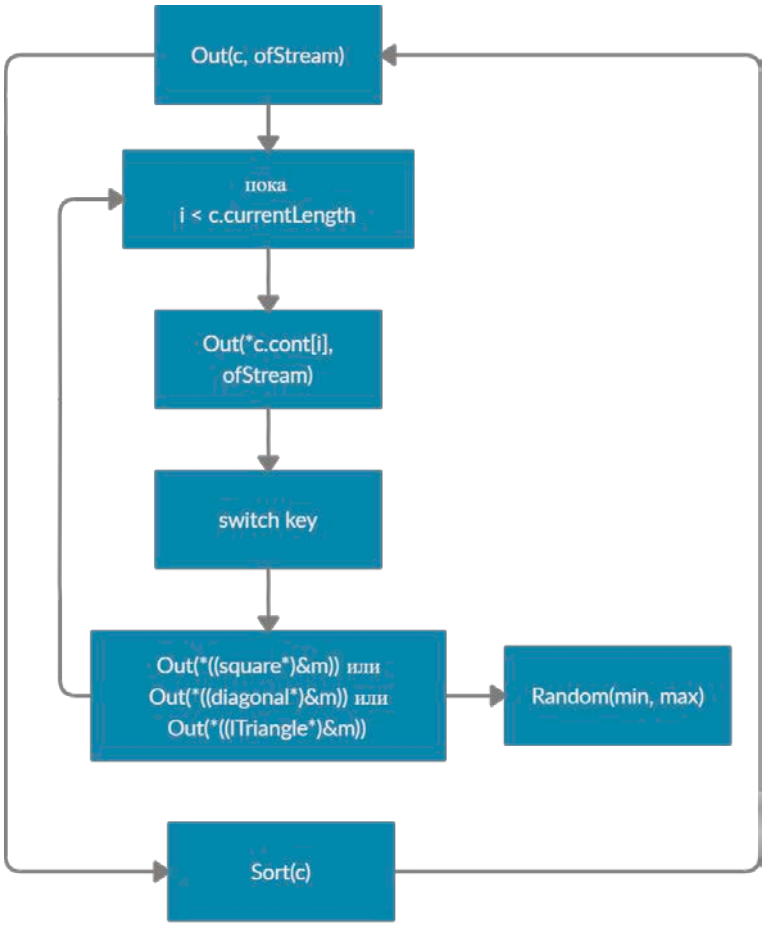


Рис. 4 Блок вывода данных в файл, сортировка, и повторный вывод

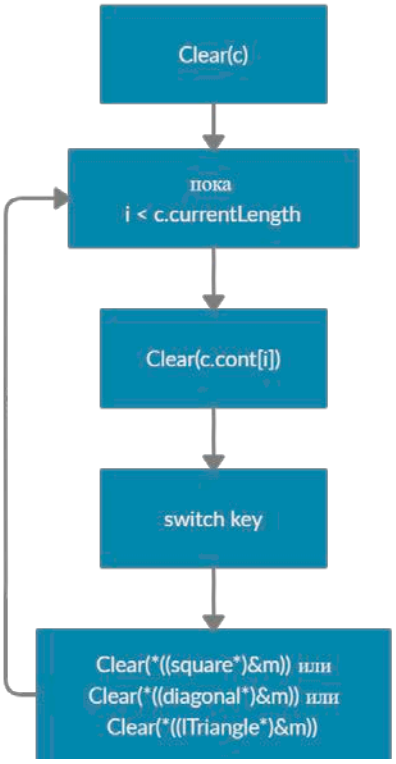


Рис. 5 Блок очистки памяти от элементов контейнера

3. Входные и выходные данные

1. В консоль поступает команда следующего типа:

   Ввод из файл: “ИмяПрограммы.exe -f «ПутьКВходномуФайлу» «ПутьКВыходномуФайлу1» «ПутьКВыходномуФайлу2»”

   Случайный ввод: “ИмяПрограммы.exe -n «КоличествоМатриц» «ПутьКВыходномуФайлу1» «ПутьКВыходномуФайлу2»”

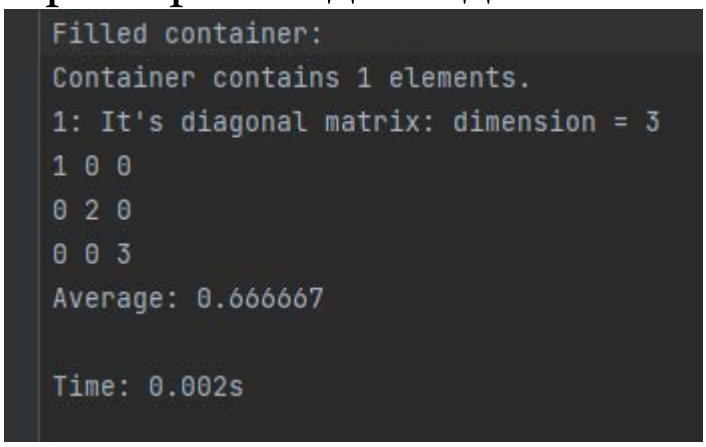
2. Образец входных данных:

   ```
   begin
   1
   2
   1 2
   3 4
   end
   ```

   Где “begin” и “end” означают начало и конец информации о матрице. 1 (2 строка)– это тип матрицы, 2 (3 строка) – это размер матрицы, следом идут элементы самой матрицы.

3. Матрица является некорректной в случае отсутствия хотя бы команды “begin” или “end”, некорректного типа матрицы (от 1 до 3), некорректной размерности матрицы (от 1 до 20), некорректных элементов матрицы. В случае, если матрица является некорректной программа ищет начало следующей матрицы или доходит до конца потока, после чего программа корректно продолжает свое выполнение.

4. После обработки данных, программа выводит их в выходной файл, после чего сортирует и выводит аналогичным образом во второй выходной файл. Пример выходных данных:



В выходном файле также указывается количество времени (в секундах) затраченное программой на обработку данных.

4. Краткий отчет

Количество заголовочных файлов: 7 шт
Количество модулей реализации: 5 шт
Время, затраченное на каждый тест указано в выходном файле соответствующего теста.