Бюджетное учреждение высшего образования   
Ханты-Мансийского автономного округа   
«Сургутский государственный университет»

Политехнический институт

Кафедра автоматики и компьютерных систем

**Отчет**  
по лабораторной работе № 8 «Обобщение навыков структурного программирования»

по дисциплине «Программирование и основы алгоритмизации»

Выполнил: Скорба Р. В.

студент группы 609-11

Проверил: Гришмановский П. В.

доцент кафедры автоматики и компьютерных систем

Сургут

2023 г.

**Цель работы:** Закрепление практических навыков технологии структурного программирования, определения и использования структур, массивов и функций.

**Задание:** Разработать программу в соответствии с вариантом индивидуального задания из предыдущей работы, используя, в том числе, уже реализованную структуру и функции.

Вариант 27. Задано расписание мероприятий, каждое из которых имеет текстовое описание и характеризуется датой (год, месяц, день) и временем (час, минута) начала и окончания. Обеспечить упорядочивание записей по времени начала, времени окончания, поиск перекрывающихся записей и максимального незанятого промежутка времени. Максимальное количество записей - не менее 25.

**Вариант 27.**

**Перечень реализованных функций для операций над структурами:**

* **void input(baza \*event, int num)** – ввод мероприятия.
* **void outone(baza \*event)** – вывод в консоль одного мероприятия.
* **int count\_interval(baza \*a, baza \*b)** – подсчет временного промежутка между мероприятиями.
* **int s\_overlapping(baza \*a, baza \*b)** – компаратор проверяющий пересекаются ли мероприятия.
* **int s\_time\_start(baza \*a, baza \*b)** – компаратор проверяющий какое из мероприятия началось раньше
* **int s\_time\_stop(baza \*a, baza \*b)** – компаратор проверяющий какое из мероприятия закончилось раньше

**Перечень реализованных функций для операции над массивом структур:**

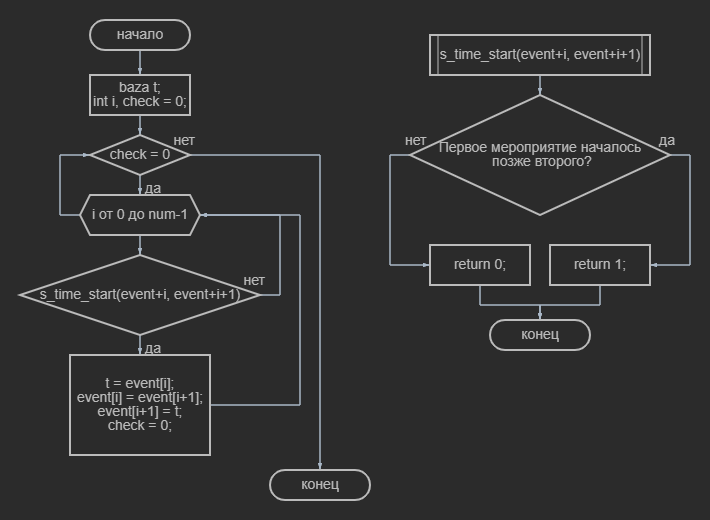
* **void output(baza \*event, int num)** – вывод в консоль массива структур.
* **int del(baza \*event, int num)** – удаление выбранного мероприятия из структуры.
* **int insert(baza \*event, int num)** – вставка мероприятия в структуру.
* **void s\_tStart(baza \*event, int num)** – сортировка мероприятия по времени начала
* **void s\_tStop(baza \*event, int num)** – сортировка мероприятий по времени окончания
* **void s\_Overlap(baza \*event, int num)** – проверка на пересечение мероприятий попарно.
* **void s\_free\_interval(baza \*event, int num)** – считает временной промежуток между не перекрывающимися мероприятиями.

Функции компараторы **s\_overlapping, s\_time\_start, int s\_time\_stop** возвращают значение типа int, 1 или 0 в зависимости от работы условия. Функции **int del, int insert** меняют значение num(количество действительных мероприятий), из-за этого тип возвращаемых данных int. Принимают два аргумента, массив структур и фактическое кол-во мероприятий для удобной работы с их элементами

Функция **int count\_interval** возвращает целове количество секунд, свободного времени между мероприятиями, следовательно – int. Принимают две соседние структуры.

Функции **void input, void outone, void output, void s\_tStart, void s\_tStop, void s\_Overlap, void s\_free\_interval** – имеют тип void, т.к. они ничего не возвращают. Принимают два аргумента, массив структур и фактическое кол-во мероприятий для удобной работы с их элементами

Блок схема алгоритма функции **void s\_tStart(baza \*event, int num):**



**Вывод:** мною была реализована задачу работы с массивом структур, а так же закрепил практические навыки структурного программирования. В результате работы я освоил методы обработки и сортировки массивов структур, а также научился использовать их для решения поставленной задачи.