Федеральное государственное бюджетное образовательное учреждение высшего образования

«Сибирский государственный университет телекоммуникаций и информатики»

(СибГУТИ)

Кафедра вычислительных систем

**ОТЧЕТ**

по практической работе 1

по дисциплине «**Программирование**»

|  |  |  |
| --- | --- | --- |
| Выполнил:  студент гр. ИВ-221  «8» февраля 2023 г. | \_\_\_\_\_\_\_\_\_\_\_\_ | Гордов Р.С. |
|  |  |  |
| Проверил:  Старший преподаватель кафедры  «13» февраля 2023 г. | \_\_\_\_\_\_\_\_\_\_\_\_ | Фульман В.О. |

Оценка «\_\_\_\_\_\_\_\_\_\_\_\_\_»

Новосибирск 2023

**ОГЛАВЛЕНИЕ**

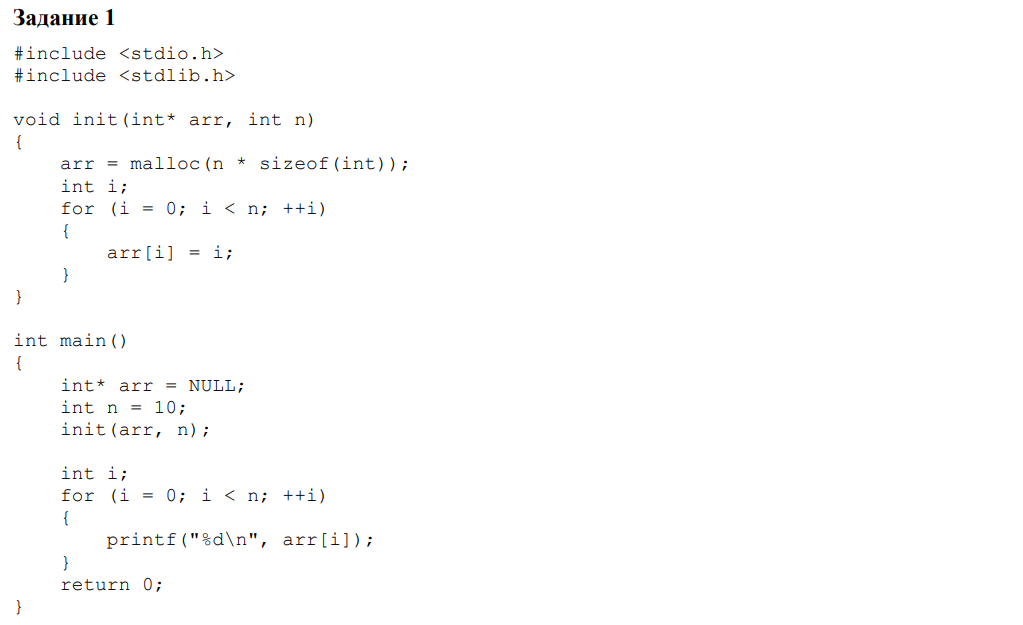
[**ЗАДАНИЕ** 3](#_Toc126776182)

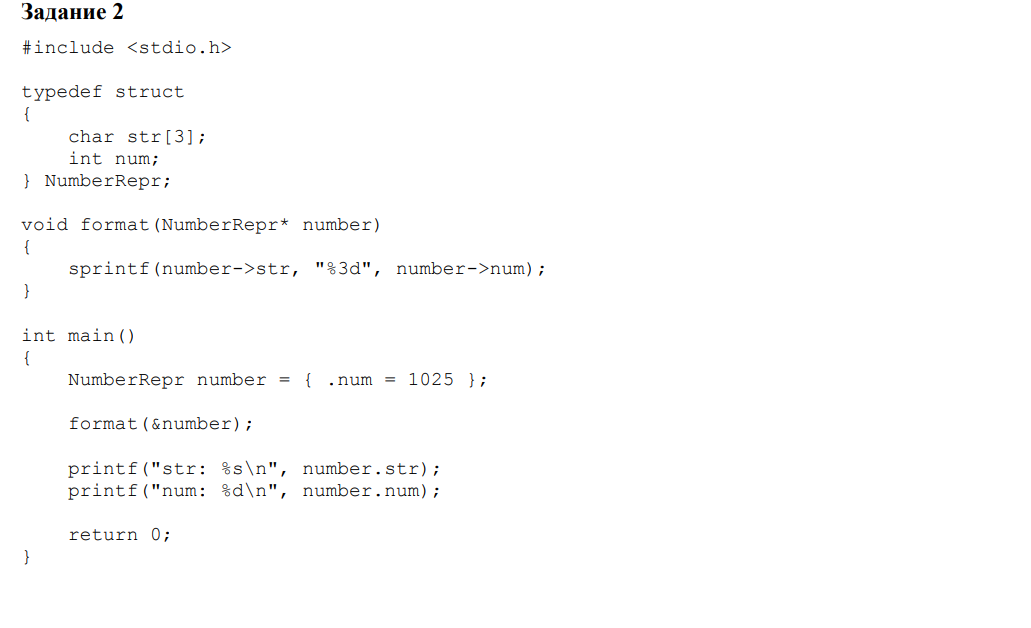
[**ВЫПОЛНЕНИЕ РАБОТЫ** 5](#_Toc126776183)

[**ПРИЛОЖЕНИЕ** 12](#_Toc126776184)

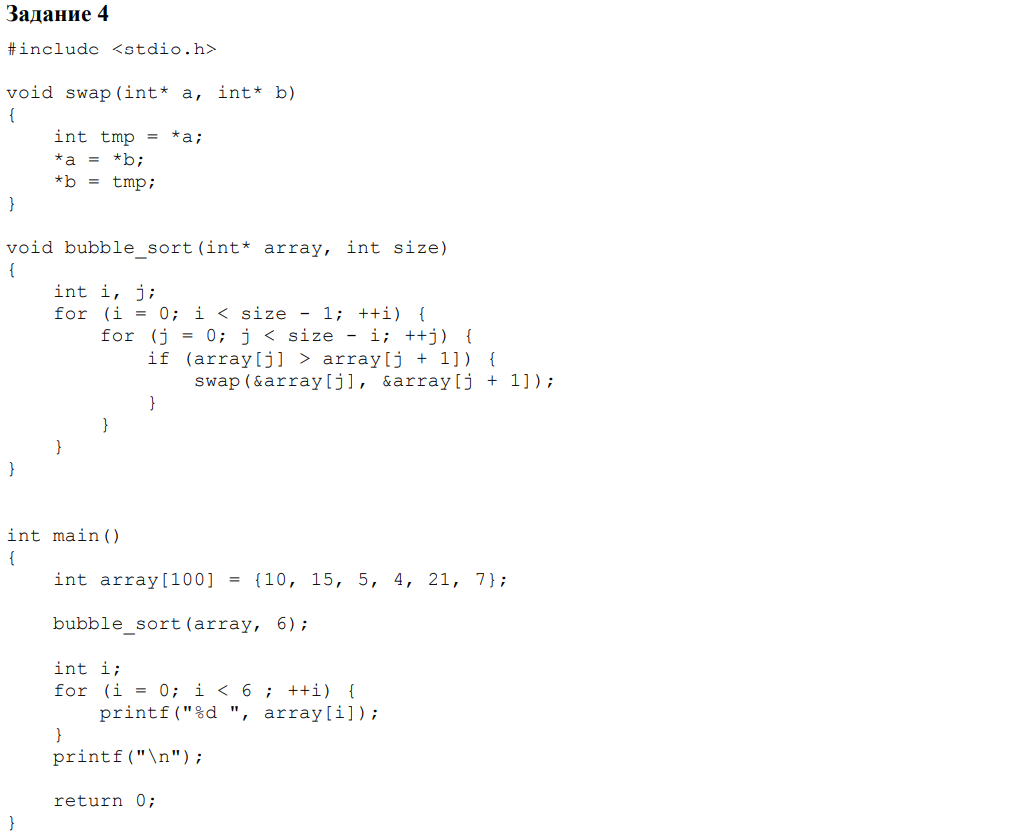
# **ЗАДАНИЕ**

В приведенных программах содержатся ошибки. Необходимо с помощью отладчика локализовать и исправить их.





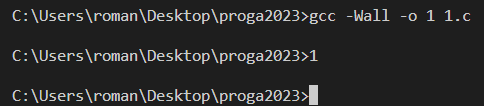




**ВЫПОЛНЕНИЕ РАБОТЫ**

**Задание 1**

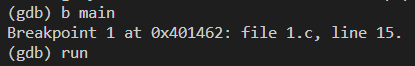
Предполагается, что первая программа, должна выделить память под массив и вывести в консоль числа, но ничего не происходит.



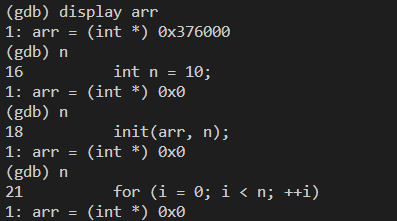
Запускаем дебагер.



Ставим breakpoint на main и запускаем.



Попробуем выводить значение переменной arr, которая должна содержать адрес первого элемента массива.



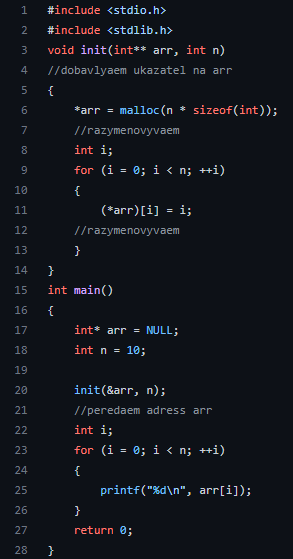
Видим, что после функции init адрес не меняется, а должен т.к. маллок выделяет память и возвращает новый адрес.

Дело в том, что при передаче значений переменных в функцию, передаются только их значения, это не те же самые переменные, а только их значение.

Все что мы не возвращаем из функции стирается из памяти.

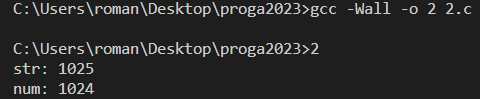
Поэтому не имеет смысла менять адреса внутри функции, нужно помнить, что внутри функции мы работаем не с самими переменными, а только с их прототипами.

Чтобы изменить саму переменную нам нужно воспользоваться указателями, то есть передать адрес, и через разыменование менять их значения.

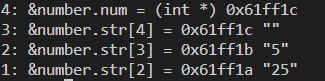


**Задание 2**

Предполагается, что функция должна вывести в консоль два раза число 1025.

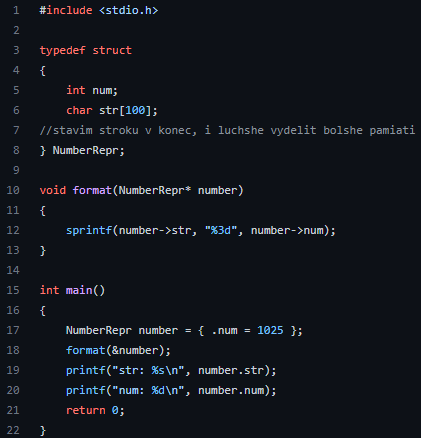


Запускаем дебагер и ставим брейкпоинт на main, попытаемся вывести адреса элементов структуры.



Видим, что когда мы пытаемся записать в строку символов, больше чем выделено, то машина начинает выделять память подряд поверх уже другого поля структуры

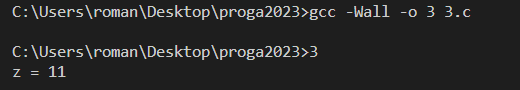
Исправляем так, выделяем больше памяти, а также можно поставить, в конец поле со строкой.



**Задача 3**

Предполагается, что директива препроцессора, должна заменять SQR(x) на x\*x, то есть возводить число в квадрат. Мы должны получить квадрат 6, то есть 36.





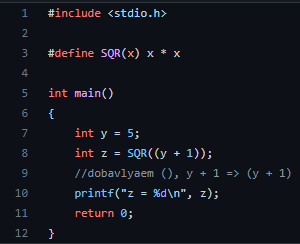
Чтобы посмотреть, что творится, после макроподсанвки скомпилируем с ключем:



Проблема в функции sqr,,а точнее в человеческом факторе, дело в том что препроцессор подставляет y+1\*y+1, поэтому мы получаем 11=5+1\*5+1.



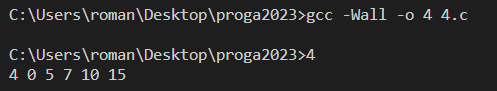
Внутри нашей функции пишем выражение в скобках.



**Задание 4**

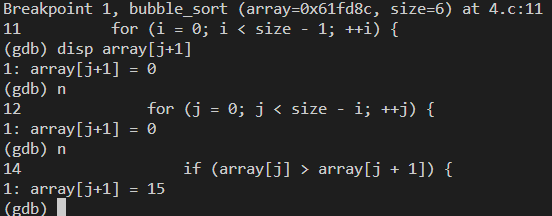
Программа должна сортировать определенное количество элементов массива.





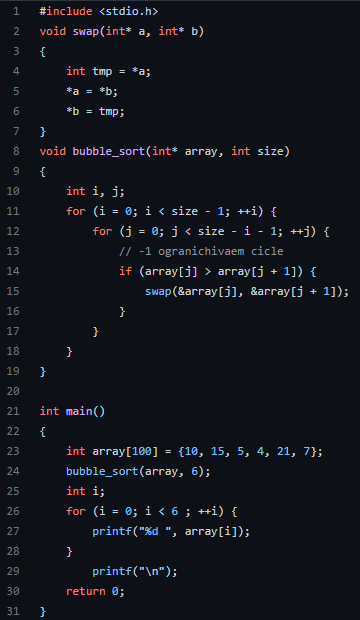
Не работает. Я САМ писал сортировку пузырьком и долго мучился с этой душещипательной проблемой. У нас появляется, элемент, который выходит за пределы цикла. Давайте запустим дебагер и посмотрим, что же не так.

Сразу посмотрим последний элемент, который мы сортируем



Видим, что у нас «всплывает» 15 и «тонет» 0, которого и не должно было быть.

Исправляем, так как на первом проходе главного цикла i=0, то size-i=size, а на последнем шаге мы выходим за пределы массива, т.к. j=(size-1)+1=size, элементы начинаются с нуля, а значит элемента с индексом size не может существовать.



# **ПРИЛОЖЕНИЕ**

**https://github.com/Herroa/proga2023**