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-222  «4» марта 2023 г. | \_\_\_\_\_\_\_\_\_\_\_\_ | Очнев А.Д. |
|  |  |  |
| Проверил:  Старший преподаватель кафедры вычислительных систем  «\_\_» марта 2023 г. | \_\_\_\_\_\_\_\_\_\_\_\_ | Фульман В.О. |

Оценка «\_\_\_\_\_\_\_\_\_\_\_\_\_»

Новосибирск 2023

**ОГЛАВЛЕНИЕ**

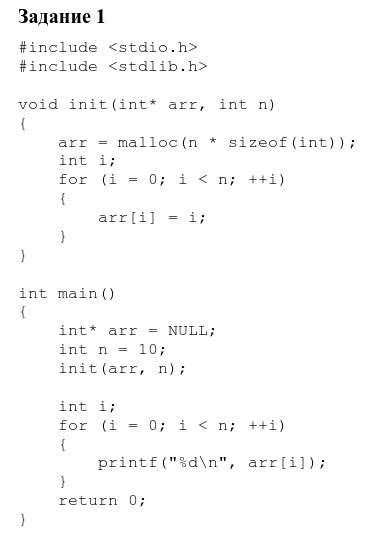
[ЗАДАНИЕ 2](#_Toc459101671)

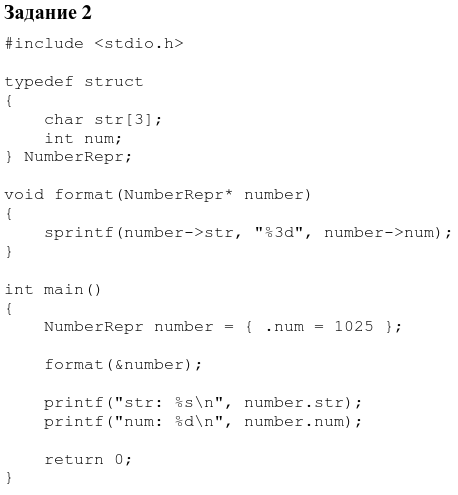
[ВЫПОЛНЕНИЕ РАБОТЫ 4](#_Toc690708757)

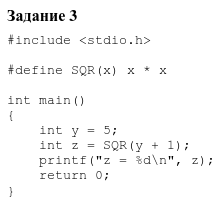
[ПРИЛОЖЕНИЕ 5](#_Toc1359233959)

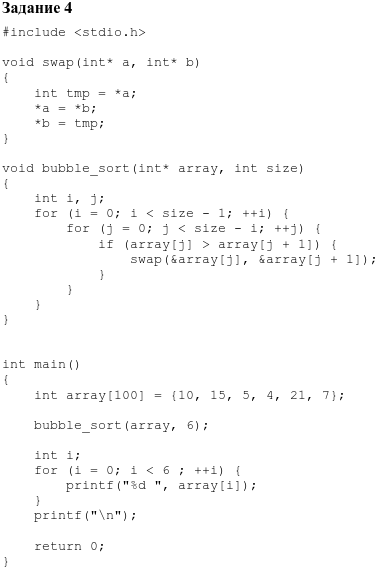
## **ЗАДАНИЕ**

Получение навыков отладки программ на примере использования отладчика GDB.









## **ВЫПОЛНЕНИЕ РАБОТЫ**

Задание 1.

В первом задании происходит заполнение массива в функции init, и последующий вывод данного массива на экран в функции main.



При компиляции ошибок не возникло, но при запуске исполняемого файла происходит ошибка сегментирования.

Компилируем программу с ключами -g, для отображения отладочной информации и -O0, для отключения оптимизации.



Запускаем gdb отладчик, устанавливаем точку остановки на объявлении массива.





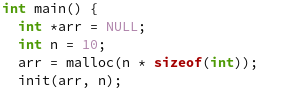
В функции init массив заполняется правильно.



После отработки функции init, массив остался не инициализированным в функции main.

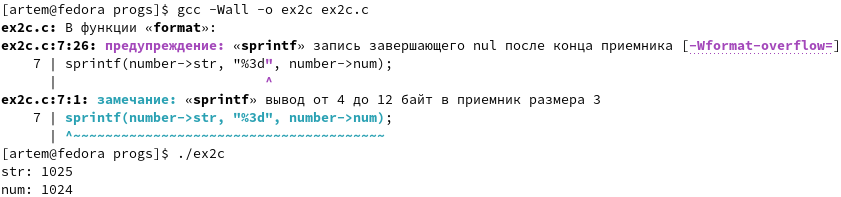


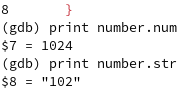
Так как участок памяти, выделяемый malloc в функции init, не распространяется на функцию main, выделяем память для массива в функции main, после чего передаём массив в init.



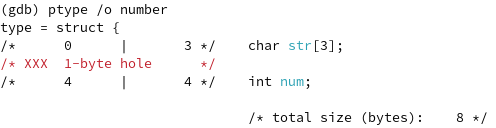
Задание 2.

Во втором задании программа должна инициализировать строковое и целочисленные поля структуры в функции format.

При компиляции видим предупреждение о записи в функции format. Выводятся неправильные значения.

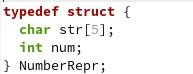


После отработки функции format поля структур инициализируются неправильно, посмотрим как они хранятся в памяти.



Поля располагаются друг за другом с пробелом в байт, так как строка вмещает только три символа, четвёртый символ не записывается.

Для исправления увеличим размер строкового массива до пяти.



Задание 3.

В третьем задании к числу должна прибавляться единица, после чего число возводится в квадрат с помощью макроса.



Программа компилируется без ошибок, но число выводится не правильное.



Если развернуть макрос, то можно увидеть, что последовательность действий макроса неправильная. Исправим последовательность добавив скобки.

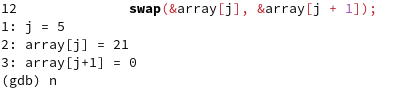


Задание 4.

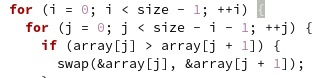
В четвёртом задании массив сортируется пузырьком, после чего выводится на экран.



При компиляции ошибок не возникло, но при запуске выводится неправильный массив.



Если вывести соседние элементы массива и переменной j, можно увидеть, что последний элемент массива меняется местами с неинициализированным элементом массива. Для исправления ошибки поменяем условие цикла.



## **ПРИЛОЖЕНИЕ**

Программа из первого задания:

**ex1c.c**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19 | #include <stdio.h> #include <stdlib.h> void **init**(**int** \*arr, **int** n) {  **int** i;  **for** (i = **0**; i < n; ++i) {  arr[i] = i;  } } int **main**() {  **int** \*arr = NULL;  **int** n = **10**;  arr = malloc(n \* **sizeof**(**int**));  init(arr, n);  **int** i;  **for** (i = **0**; i < n; ++i) {  printf("%d**\n**", arr[i]);  }  **return** **0**; } |

Программа из второго задания:

**ex2c.c**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | #include <stdio.h> typedef **struct** {  **char** str[**5**];  **int** num; } NumberRepr; void **format**(NumberRepr \*number) {  sprintf(number->str, "%3d", number->num);  } int **main**() {  NumberRepr number = {.num = **1025**};  format(&number);  printf("str: %s**\n**", number.str);  printf("num: %d**\n**", number.num);  **return** **0**; } |

Программа из третьего задания:

**ex3c.c**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | #include <stdio.h> #define SQR(x) (x) \*(x) int **main**() {  **int** y = **5**;  **int** z = SQR(y + **1**);  printf("z = %d**\n**", z);  **return** **0**; } |

Программа из четвёртого задания:

**ex4c.c**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26 | #include <stdio.h> void **swap**(**int** \*a, **int** \*b) {  **int** tmp = \*a;  \*a = \*b;  \*b = tmp; } void **bubble\_sort**(**int** \*array, **int** size) {  **int** i, j;  **for** (i = **0**; i < size - **1**; ++i) {  **for** (j = **0**; j < size - i - **1**; ++j) {  **if** (array[j] > array[j + **1**]) {  swap(&array[j], &array[j + **1**]);  }  }  } } int **main**() {  **int** array[**100**] = {**10**, **15**, **5**, **4**, **21**, **7**};  bubble\_sort(array, **6**);  **int** i;  **for** (i = **0**; i < **6**; ++i) {  printf("%d ", array[i]);  }  printf("**\n**");  **return** **0**; } |