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. ИС-242

«8» марта 2023 г.

/Воркунов К.С./

Проверил:

Старший Преподаватель Кафедры ВС

«10» марта 2023 г.

/Фульман В.О./

Оценка « »

Новосибирск 2023

# ОГЛАВЛЕНИЕ

[ОГЛАВЛЕНИЕ 1](#_bookmark0)

[ЗАДАНИЕ 2](#_bookmark1)

[ПРИЛОЖЕНИЕ 9](#_bookmark2)

# ЗАДАНИЕ

В лабораторной работе представлены задачи с кодом. Необходимо с помощью отладчика локализовать и исправить ошибки.

Задание 1:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20 | #include <stdio.h>  #include <stdlib.h>  **void** **init**(**int** \*arr, **int** n) {  arr = malloc(n \* **sizeof**(**int**));  **int** i;  **for** (i = **0**; i < n; ++i) {  arr[i] = i;  }  }  **int** **main**() {  **int** \*arr = NULL;  **int** n = **10**;  init(arr, n);  **int** i;  **for** (i = **0**; i < n; ++i) {  printf("%d**\n**", arr[i]);  }  **return** **0**;  } |

Задание 2:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | #include <stdio.h>  **typedef** **struct** {  **char** str[**3**];  **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**;  } |

Задание 3:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | #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**;  } |

Задание 4:

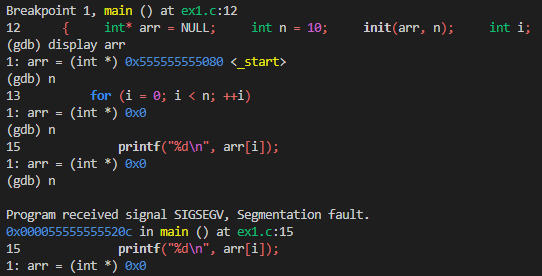
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26 | #include <stdio.h>  **void** **swap**(**int** \*a, **int** \*b) {  **int** tmp = \*a;  \*a = \*b;  \*b = tmp;  }  **void** **bubble\_sort**(**int** \*array, **int** size) {  **int**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

**ВЫПОЛНЕНИЕ РАБОТЫ**

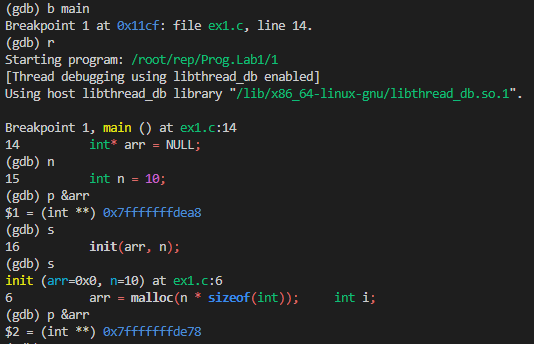
Задание 1:

Компилируем и запускаем программу, получаем ошибку сегментирования.

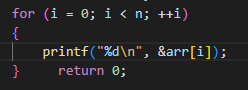
Рассмотрим адреса переменных в функциях main и init c помощью отладчика gdb.



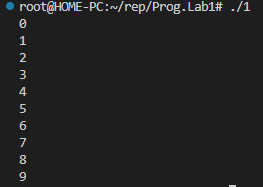
В функции init выделяется память под массив и инициализируются его элементы. Как можно увидеть на скриншоте выше, после выполнения функции init, в переменной arr в функции main остается указатель на нулевую память, что приводит к ошибке. Это происходит из-за того, что в локальной переменной arr в функции init сохранен указатель на выделенную память, но указатель не возвращается в функцию main. Для исправления ошибки нужно возвращать указатель на выделенную память.



На верхнем скриншоте видим, что адрес переменной arr в main и init отличается, так как при передаче переменной в функцию создается временная переменная, срок жизни которой блок данной функции, и после ее завершения переменная аrr опять ссылается на нулевую память.



Посмотрим на результат работы программы после исправления ошибки.

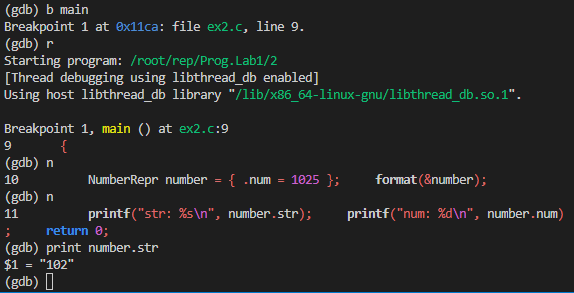


Задание 2:

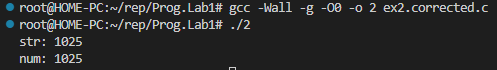
Компилируем и запускаем программу, после выполнения программы элемент num структуры number меняет свое значение на 1024, хотя напрямую мы не меняли значение элемента num.



Вызвано это тем, что мы пытаемся записать в массив char str[3], состоящий из трех элементов, четырехзначное число и символ конца строки. Из чего делаем вывод, что нам нужно увеличить массив str в структуре на 2 ячейки, т.к. массив Char является строчкой с завещающим нулевым символом, для которого требуется еще один байт (ячейка).



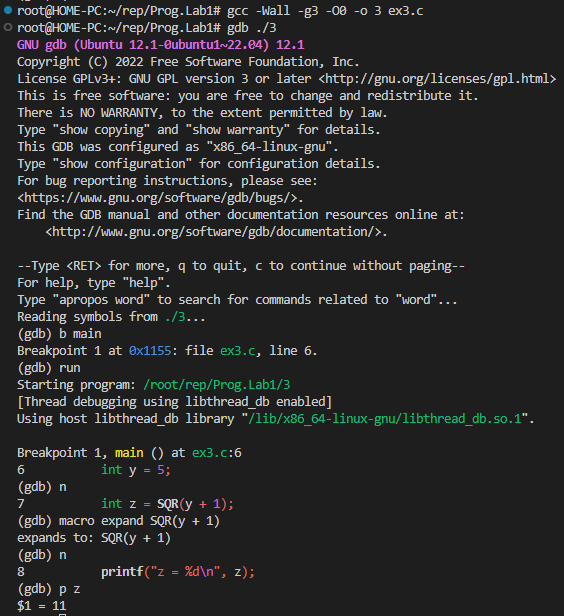
После внесенных изменений получаем такой результат:



Задание 3:

В ходе выполнения программы происходит ошибка в вычислении квадрата.

Посмотрим во что будет развернут макрос SQR. Для этого необходимо скомпилировать программу с ключом –g3. Запустим отладчик и введём команду macro expand + имя макроса и значение для проверки.



Ошибка происходит из-за неправильного порядка выполнения арифметических операций. Для решения предлагаю расставить скобки, чтобы порядок стал правильным.

Посмотрим на результат работы программы после исправления ошибки.



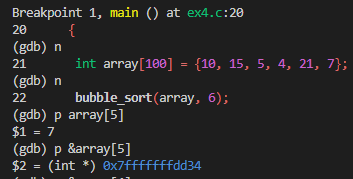
gcc

Задача 4:

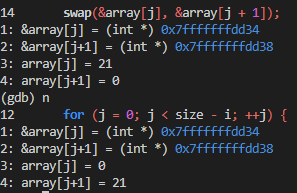
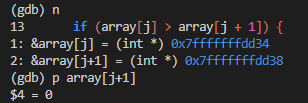
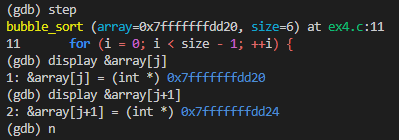
После выполнения программы мы видим неправильно отсортированный массив, а 00также элементы, которых не было в исходном массиве.



Попробуем проследить за процессом сортировки и найти ошибку. Узнаем адрес последнего элемента.



Проходим по сортировке и видим, что сортировка сравнивает следующий элемент, после инициализированного элемента массива, который равен нулю.



Что бы этого избежать, мы добавим -1 в цикле к size

Ошибка происходит из-за того, что совершается лишняя итерация внутреннего цикла пузырьковой сортировки. Для решения необходимо добавить -1 в условии работы цикла.

После исправления ошибки программа правильно сортирует массив.



# ПРИЛОЖЕНИЕ

ex1.corected

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21 | #include <stdio.h>  #include <stdlib.h>  **Int** \***init**(**int** \*arr, **int** n) {  arr = malloc(n \* **sizeof**(**int**));  **int** i;  **for** (i = **0**; i < n; ++i) {  arr[i] = i;  }  **return** arr;  }  **int** **main**() {  **int** \*arr = NULL;  **int** n = **10**;  arr = init(arr, n);  **int** i;  **for** (i = **0**; i < n; ++i) {  printf("%d**\n**", arr[i]);  }  **return** **0**;  } |

ex2.corected

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | #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**;  } |

ex3.corected

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | #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**;  } |

ex4.corected

|  |  |
| --- | --- |
| 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  27 | #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**;  } |