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** 6](#_Toc126776183)

[**ПРИЛОЖЕНИЕ** 10](#_Toc126776184)

# **ЗАДАНИЕ**

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

Задание 1

|  |  |
| --- | --- |
| 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  28 | #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  14  15  16  17  18  19  20  21  22 | #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  10  11  12 | #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**;  } |

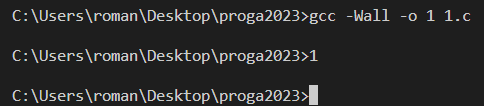
Задание 4

|  |  |
| --- | --- |
| 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  27  28  29  30  31 | #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**

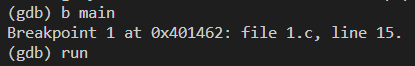
Предполагается, что первая программа, должна выделить память под массив и вывести в консоль числа, но ничего не происходит.



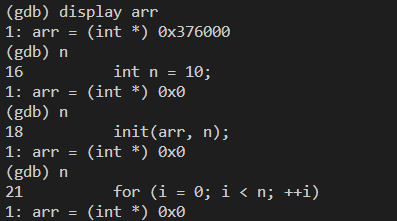
Запускаем отладчик.



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. malloc 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.

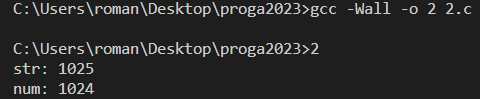
Все что мы не возвращаем из функции стирается из памяти.

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

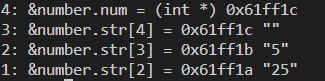
Чтобы изменить саму переменную нам нужно воспользоваться указателями, то есть передать адрес, и через разыменование менять их значения.

**Задание 2**

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



Запускаем отладчик и ставим breakpoint на main, попытаемся вывести адреса элементов структуры.



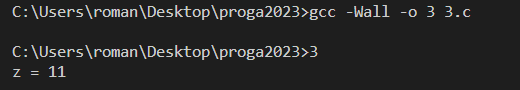
Видим, что когда мы пытаемся записать в строку символов, больше чем выделено, то машина начинает выделять память подряд поверх уже другого поля структуры

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 макроподстановки скомпилируем с ключом:



Проблема в функции SQR(), препроцессор подставляет y+1\*y+1, поэтому мы получаем 11=5+1\*5+1.

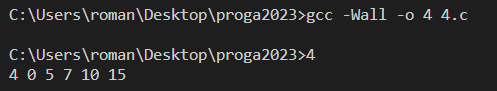


Исправляем макроподстановку.

**Задание 4**

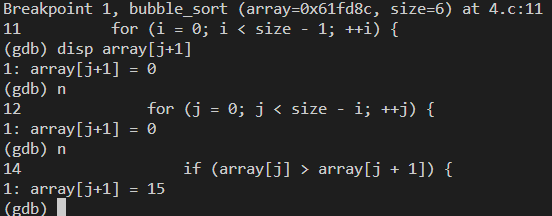
Программа должна сортировать определенное количество элементов массива.





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 отладчик 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**](https://github.com/Herroa/proga2023)

Задание 1

|  |  |
| --- | --- |
| 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  28 | #include <stdio.h>  #include <stdlib.h>  **void** **init**(**int**\*\* arr, **int** n)  //add a pointer  {  \*arr = malloc(n \* **sizeof**(**int**));  //dereferencing  **int** i;  **for** (i = **0**; i < n; ++i)  {  (\*arr)[i] = i;  //dereferencing  }  }  **int** **main**()  {  **int**\* arr = NULL;  **int** n = **10**;  init(&arr, n);  //transfer the address of arr  **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  14  15  16  17  18  19  20  21  22 | #include <stdio.h>  **typedef** **struct**  {  **int** num;  **char** str[**100**];  //allocate more memory and rearrange the line to the end  } 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  10  11  12 | #include <stdio.h>  #define SQR(x) ((x) \* (x))  //fix macro substition  **int** **main**()  {  **int** y = **1**;  **double** z = **1.0** / SQR(y + **1**);  printf("z = %lf**\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  27  28  29  30  31 | #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) {  // limit range of cycle  **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**;  } |