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  «13» марта 2023 г. | \_\_\_\_\_\_\_\_\_\_\_\_ | Павленко П. В. |
|  |  |  |
| Проверил:  Старший преподаватель Кафедры ВС  «13» марта 2023 г. | \_\_\_\_\_\_\_\_\_\_\_\_ | Фульман В.О. |

Оценка «\_\_\_\_\_\_\_\_\_\_\_\_\_»

Новосибирск 2023

**ОГЛАВЛЕНИЕ**

[**ЗАДАНИЕ** 3](#_Toc127112463)

[Задание 1 3](#_Toc127112464)

[Задание 2 4](#_Toc127112465)

[Задание 3 4](#_Toc127112466)

[Задание 4 5](#_Toc127112467)

[**ВЫПОЛНЕНИЕ РАБОТЫ** 6](#_Toc127112468)

[Задание 1 6](#_Toc127112469)

[Задание 2 8](#_Toc127112470)

[Задание 3 10](#_Toc127112471)

[Задание 4 11](#_Toc127112472)

[**ПРИЛОЖЕНИЕ** 12](#_Toc127112473)

[1 задание (исправленный код) 12](#_Toc127112474)

[2 задание (исправленный код) 13](#_Toc127112475)

[3 задание (исправленный код) 13](#_Toc127112476)

[4 задание (исправленный код) 14](#_Toc127112477)

# **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 | #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  23 | #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 | #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  32  33  34  35 | #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;  } |

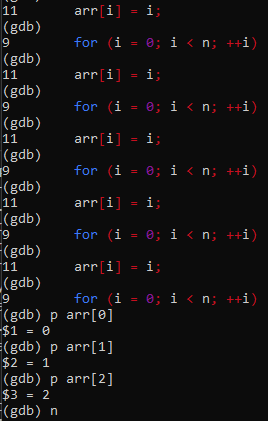
**ВЫПОЛНЕНИЕ РАБОТЫ**

## Задание 1

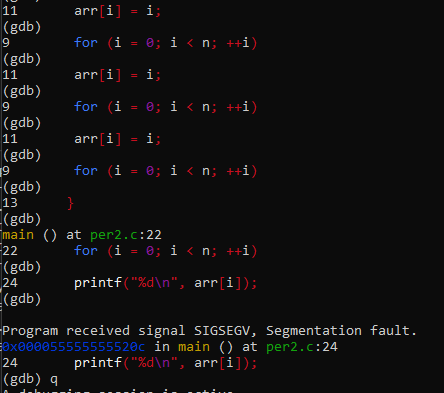
Программа 1 инициализирует нулевой указатель arr на int, а затем с помощью функции init должен быть создан динамический массив размером n и заполнен числами от 0 до n. При запуске программы выдается ошибка сегментирования.



При помощи GDB видно, что массив внутри функции init заполняется успешно.



При выходе из функции init и переходе в цикл в функции main на экран выводится ошибка Segmentation fault с сигналом SIGSEGV (сигнализирует об ошибке сегментации).



Как можно понять, ошибка в этом коде кроется в функции init, а именно в том, что массив остается внутри функции и никуда далее не выходит, поэтому в main при попытке использовать printf программа выдает ошибку. Чтобы код работал исправно нужно выводить массив из функции init. Просто добавим вывод массива.

## Задание 2

Задумка программиста — представить число 1025 в строковом и целочисленном представлении. При сборке программы, компилятор предупреждает нас о возможной ошибке — признак конца строки записывается за пределами назначения.

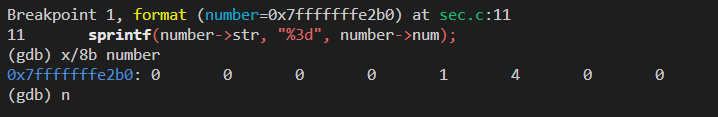
## 

При запуске программы, можно увидеть, что строковое и целочисленное представление числа не совпадают.

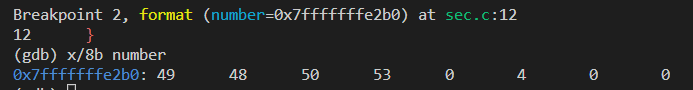


Выведем память нашей структуры на экран. На первый взгляд, она занимает в памяти только семь байт — но это не правда, поскольку поля структуры выравниваются в памяти.

Выведем восемь байт памяти структуры number.



Запускаем программу и видим, как функция sprintf невозбранно перезаписывает младший разряд признаком конца строки.



Все из-за того, что sprintf записывает строку длиной в 5 символов. Четыре символа это “1025” и пятый символ это конец строки. В то время как выделяется место для трех символов. Два лишних символа перезаписывают в пустое место в памяти оставленное для выравнивания, от чего и изменятся значение num.

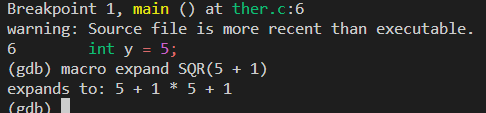
Для решения нужно выделить больше памяти для строки. Из “str[3]” сделать “str[5]”

## Задание 3

Программа 3 содержит макрос, подразумевающий возведение числа в квадрат, но при запуске программы результат не совпадает с ожиданиями.



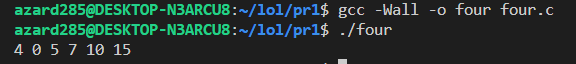
Используя команду macro expand, можно увидеть ошибку. Она заключается в том, что арифметические действия идут не по порядку.



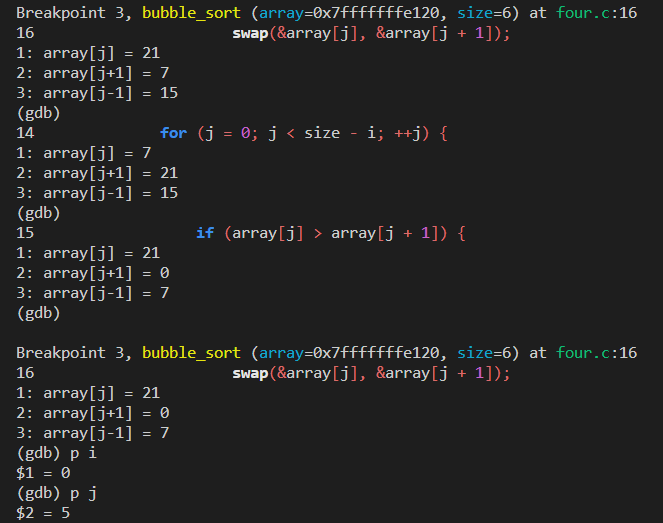
Чтобы программа работала правильно нужно упорядочить действия макроса. Этого можно добиться путем исправления выражения в макросе.

## Задание 4

Программа 4 подразумевает пузырьковую сортировку массива. При запуске программы массив сортируется некорректно.



На выводе видно, что происходит выход за границу массива. Проверим где и в какой момент он выходит



Видно, что после шестого числа берётся еще одно (седьмое, array[j+1], индекс 6). После массива числа инициализируются нулями. Чтобы все работало, нужно изменить условия циклов: в первом – с “i < size - 1” на “i < size”, во втором – с “j < size – i” на “j < size – i – 1”.

# **ПРИЛОЖЕНИЕ**

## 1 задание (исправленный код)

|  |  |
| --- | --- |
| 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>  #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;  } |

## 2 задание (исправленный код)

|  |  |
| --- | --- |
| 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 | #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;  } |

## 3 задание (исправленный код)

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

## 4 задание (исправленный код)

|  |  |
| --- | --- |
| 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  32  33  34  35  36  37 | #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; ++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**");  } |

P.S: [azard285/laba1 (github.com)](https://github.com/azard285/laba1)