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
«16» февраля 2023 г _____ Денисов М. А.

Проверил:
Старший преподаватель
Кафедры ВС
«16» февраля 2023 г. _____ Фульман В.О.

Оценка «_____»

Новосибирск 2023

ОГЛАВЛЕНИЕ

ЗАДАНИЕ

С помощью отладчика локализовать ошибки и исправить их.

Задание 1.

```
 1  #include <stdio.h>
 2  #include <stdlib.h>
 3  void init(int* arr, int n)
 4  {
 5      arr = malloc(n * sizeof(int));
 6      int i;
 7      for (i = 0; i < n; ++i)
 8      {
 9          arr[i] = i;
10      }
11  }
12  int main()
13  {
14      int* arr = NULL;
15      int n = 10;
16      init(arr, n);
17      int i;
18      for (i = 0; i < n; ++i)
19      {
20          printf("%d\n", arr[i]);
21      }
22      return 0;
23  }
24
25
26
27
28
29
30
31
32
33
```

Задание 2.

```c
#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.

```c
#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.

```c
#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

Описывается ход работы над заданием с приложением снимков экрана.

Задание 1

Ошибка заключается в том, что в переменной arr лежит нулевой указатель. функция init выделяет память под массив и заполняет его внутри себя, но не возвращает его.

```
(gdb) p *arr
Cannot access memory at address 0x0
(gdb) p arr
$1 = (int *) 0x0
(gdb) n
18          for (i = 0; i < n; ++i)
(gdb) n
20              printf("%d\n", arr[i]);
(gdb) n

Program received signal SIGSEGV, Segmentation fault.
0x0000555555400724 in main () at zadanie1gdb.c:20
20              printf("%d\n", arr[i]);
(gdb)
```

Нужно узнать, где данный массив, который заполнила функция init находиться в памяти, для этого, вернем указатель на данную память.

```
int* init(int* arr, int n)
{
    arr = malloc(n * sizeof(int));
    int i;
    for (i = 0; i < n; ++i)
    {
        arr[i] = i;
    }
    return arr;
}
```

После, в main присвоим указатель на память массиву arr.

```
15      int* arr = NULL;
16      int n = 10;
17      arr = init(arr, n);
18      int i;
19      for (i = 0; i < n; ++i)
```

После всех проделанных действий, в массиве arr лежит указатель на память, которую мы и хотели получить

```
(gdb) b 16
Breakpoint 1 at 0x742: file zadanie1.c, line 16.
(gdb) r
Starting program: /home/oteccc/lab_1_prog/gdb1

Breakpoint 1, main () at zadanie1.c:16
16          int n = 10;
(gdb) n
17          arr = init(arr, n);
(gdb) n
19          for (i = 0; i < n; ++i)
(gdb) p arr
$1 = (int *) 0x555555602260
(gdb)
```

Задание 2

В задании 2 под структуру выделяется 8 байт. 3 байта на char str[3],один выравнивающий байт и 4 байта на int number. При попытке записать четырех значное число, возникает проблема в том, что только 3 элемента под запись числа + терминальный ноль, следовательно, когда функция пытается положить ещё значения, портятся другие поля структуры.

```
(gdb) p number.str
$12 = "102"
(gdb) p number.str[3]
$13 = 53 '5'
(gdb) p number.str[4]
$14 = 0 '\000'
(gdb) ptype /o number
type = struct {
/*    0      |     3 */    char str[3];
/* XXX  1-byte hole  */
/*    4      |     4 */    int num;

                            /* total size (bytes):    8 */
                          }
(gdb)
 Breakpoint 1, main () at zadanie2gdb.c:15
 15         printf("str: %s\n", number.str);
 (gdb) p number.num
 $1 = 1024
 (gdb)
```

Выделим 5 байт для массива str.

```
1   #include <stdio.h>
2   typedef struct
3   {
4       char str[5];
5       int num;
6   } NumberRepr;
7   void format(NumberRepr* numbe
8   {
9       sprintf(number->str, "%3d
```

После этого, проверим, не портится ли данные о числе number.num.

```
Breakpoint 1, main () at zadanie2gdb.c:15
15         printf("str: %s\n", number.str);
(gdb) p number.str[3]
$1 = 53 '5'
(gdb) p number.str[4]
$2 = 0 '\000'
(gdb) p number.num
$3 = 1025
(gdb)
```

Как видим, после исправления, все происходит корректно.

Задание 3

Ошибка заключалась в том, что был не расставлен приоритет операций.

```
Breakpoint 1, main () at zadanie3gdb.c:5
5           int y = 5;
(gdb) n
6           int z = SQR(y + 1);
(gdb) n
7           printf("z = %d\n", z);
(gdb) p z
$1 = 11
(gdb)
```

Действие, которые происходит в функции SQR должно быть заключено в скобки, для корректной работы программы.

```c
#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;
}

```

После этого, убедимся, работает ли программа корректно

```
oteccc@DESKTOP-6HJ7UT2:~/lab_1_prog$ ./zadanie3
z = 36
oteccc@DESKTOP-6HJ7UT2:~/lab_1_prog$ gcc -Wall
```

```
Breakpoint 1 at 0x659: file zadanie3.c, line 7.
(gdb) r
Starting program: /home/oteccc/lab_1_prog/zadanie3gdb

Breakpoint 1, main () at zadanie3.c:7
7           int z = SQR((y + 1));
(gdb) n
8           printf("z = %d\n", z);
(gdb) p z
$1 = 36
(gdb)
```

Задание 4

В пузырьковой сортировке, во втором цикле for по j, в условии пропущено действие, из – за этого выходим за границы массива.

```
(gdb) p *array@6
$4 = {10, 5, 4, 15, 7, 21}
(gdb) n
13                  if (array[j] > array[j + 1]) {
(gdb) n
14                      swap(&array[j], &array[j + 1]);
(gdb) n
12              for (j = 0; j < size - i; ++j) {
(gdb) p *array@6
$5 = {10, 5, 4, 15, 7, 0}
(gdb) p *array@6
```

Исправим в цикле пропущенное действие.

```
{
    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]);
            }
        }
    }
```

Проверим корректность работы программы.

```
}
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]);
    }
```

ПРОБЛЕМЫ ВЫХОДНЫЕ ДАННЫЕ КОНСОЛЬ ОТЛАДКИ ТЕРМИНАЛ bash

```
oteccc@DESKTOP-6HJ7UT2:~/lab_1_prog$ ./zadanie4
4 5 7 10 15 21
oteccc@DESKTOP-6HJ7UT2:~/lab_1_prog$
```

```
(gdb) p *array@6
$4 = {10, 5, 4, 15, 7, 21}
(gdb) n
11          for (i = 0; i < size - 1; ++i) {
(gdb) n

Breakpoint 1, bubble_sort (array=0x7fffffffdcb0, size=6) at zadanie4.c:12
12              for (j = 0; j < size - i - 1; ++j) {
(gdb) p *array@6
$5 = {10, 5, 4, 15, 7, 21}
(gdb) n
13                  if (array[j] > array[j + 1]) {
(gdb) n
14                      swap(&array[j], &array[j + 1]);
(gdb) n
12              for (j = 0; j < size - i - 1; ++j) {
(gdb) p *array@6
$6 = {5, 10, 4, 15, 7, 21}
(gdb)
```

ПРИЛОЖЕНИЕ

Исходный код с комментариями.

LabaN1.c

```c
#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]);
    }
    free(arr);
    return 0;
}
```

LabaN2.c

```c
#include <stdio.h>
typedef struct
{
    char str[5];
    int num;
} NumberRepr;
void format(NumberRepr* number)
{
    sprintf(number->str, "%d", number->num);
}
int main()
{
    NumberRepr number = {.num = 1902};
    format(&number);
    printf("str: %s\n", number.str);
    printf("num: %d\n", number.num);
    return 0;
}
```

LabaN3.c

```c
#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;
}
```

LabaN4.c

```c
#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;
}
```