

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
«21» апреля 2023 г.

/Журбенко В. Е./

Проверил:
Ст. преподаватель Кафедры ВС
«__» апреля 2023 г.

/Фульман В.О./

Оценка « _____ »

Новосибирск 2023

ОГЛАВЛЕНИЕ

ЗАДАНИЕ	3
ВЫПОЛНЕНИЕ РАБОТЫ.....	5
ПРИЛОЖЕНИЕ.....	9

ЗАДАНИЕ

Найти и исправить с помощью отладчика ошибки в программах :

Задание 1

```
#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

```
#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

```
#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

```
#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

При запуске программы мы получаем ошибку

```
xcredo@xcredo-Pc:~/Documents/work1/Prog$ ./prog1
Segmentation fault (core dumped)
```

что означает попытку обращения к недоступным участкам памяти.

Запускаем отладчик gdb:

```
Program received signal SIGSEGV, Segmentation fault.
0x000055555555520c in main () at prog1.c:24
24      printf("%d\n", arr[i]);
```

Он показывает нам ошибку в 24 строчке printf. Если мы попробуем вывести arr[i] то получаем ошибку: Не удастся получить доступ к памяти по адресу 0x0.

Это дает понять что адрес в функции и main не совпадают, а это значит, что перед void нужно поставить * и добавить return arr(иначе выдает предупреждение при компиляции и ошибку при запуске)

```
Program received signal SIGSEGV, Segmentation fault.
0x000055555555520c in main () at prog1.c:24
24      printf("%d\n", arr[i]);
(gdb) p arr[i]
Cannot access memory at address 0x0
```

Так как init теперь у нас указатель то нужно изменить обращение к функции и добавить arr =.

```
21      arr=init(arr, n);
```

после запускаем программу и получаем вывод массива

```
xcredo@xcredo-Pc:~/Documents/work1/Prog$ ./prog1.1
0
1
2
3
4
5
6
7
8
9
```

Задание 2

В ходе выполнения программа должна вывести цифру 1025 в виде строки и цифры, но в строчку выводит 1025 и цифрой 1024.

```
prog1.c: In function 'format':
prog1.c:42:30: warning: 'sprintf' writing a terminating nul past the end of the destination [-Wformat-overflow=]
   42 |     sprintf(number->str, "%3d", number->num);
      |                               ^
prog1.c:42:5: note: 'sprintf' output between 4 and 12 bytes into a destination of size 3
   42 |     sprintf(number->str, "%3d", number->num);
      |     ^~~~~~
xcredo@xcredo-Pc:~/Documents/work1/Prog$ ./pro
str: 1025
num: 1024
```

Запускаем отладчик gdb и ставим точку останова на вывод строки (строка 49), а также смотрим, что находится в number.str . Там находятся 3 символа.

```
Reading symbols from pro...
(gdb) b 49
Breakpoint 1 at 0x11f4: file prog1.c, line 49.
(gdb) run
Starting program: /home/xcredo/Documents/work1/Prog/pro
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".

Breakpoint 1, main () at prog1.c:49
49     printf("str: %s\n", number.str);
(gdb) p number.str
$1 = "102"
```

Из чего делаем вывод, что нам нужно увеличить массив str в структуре, но не на 1 ячейку, а на 2 т.к. массив Char является строкой с завершающим нулевым символом, для которого требуется еще один байт (ячейка). В итоге структура у нас будет выглядеть так

```
35     typedef struct
36     {
37         char str[5];
38         int num;
39     } NumberRepr;
```

И программа начнет выводить 1025 в строке и 1025 цифрой

```
str: 1025
num: 1025
[1] + Done
"/usr/bin/gdb" --interpreter=mi --tty=${DbgTerm} 0<"/tmp/Microsoft-MIEngine
-In-c22ub0xc.zmi" 1>"/tmp/Microsoft-MIEngine-Out-elttvf30.usp"
xcredo@xcredo-Pc:~/Documents/work1/Prog$
```

Задание 3

После выполнения, программа должна нам вывести квадрат $y+1$. Так как $y=5$ то в ответе мы должны получить 36 ($(5+1)^2 = 36$). Программа не выдает ошибок при запуске, но в ответ выводит 11.

```
xcredo@xcredo-PC:~$ ./ex3
z = 11
```

Компилируем программу, но с указателем `-g3`, так как проблема, скорее всего с макросом.

Запускаем отладчик и вводим команду `macro expand` + имя макроса и значение для проверки

```
(gdb) macro expand SQR(6)
expands to: 6 * 6
```

Вроде выглядит правильно, но если мы подставим нашу формулу, то тут и начнутся проблемы.

```
(gdb) macro expand SQR(5+1)
expands to: 5+1 * 5+1
```

Как известно умножение выполняется в первую очередь, поэтому у нас получается $5+6=11$. Чтобы этого не было стоит поставить скобки в макросе.

```
#define SQR(x) ((x) * (x))
```

Теперь программа работает правильно.

```
z = 36
[1] + Done "/usr/bin/gdb" --interpreter=mi --tty=${DbgTerm} 0<"/tmp/Microsoft-MIEngine-In-3n3t1ce0.0yd" 1>"/tmp/Microsoft-MIEngine-Out-moksbz5e.udy"
xcredo@xcredo-PC:~/Documents/work1/Prog$
```

Задание 4

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

```
4 0 5 7 10 15
[1] + Done "/usr/bin/gdb" --interpreter=mi --tty=${DbgTerm} 0<"/tmp/Microsoft-MIEngine-In-3zhjg01n.4ff" 1>"/tmp/Microsoft-MIEngine-Out-kppj4z2f.ybh"
xcredo@xcredo-PC:~/Documents/work1/Prog$
```

Установим break point на 85 строке, а также пропишем display для проверки.

```
(gdb) b 85
Breakpoint 1 at 0x555555551cd: file prog1.c, line 85.
(gdb) r
Starting program: /home/xcredo/Documents/work1/Prog/pro2
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".

Breakpoint 1, bubble_sort (array=0x7fffffff920, size=6) at prog1.c:85
85         for (i = 0; i < size - 1; ++i)
(gdb) display array[1]
1: array[1] = 15
(gdb) display array[2]
2: array[2] = 5
(gdb) display array[3]
3: array[3] = 4
(gdb) display array[4]
4: array[4] = 21
(gdb) display array[5]
5: array[5] = 7
(gdb) display array[6]
6: array[6] = 0
```

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. Всё работает исправно.

```
4 5 7 10 15 21
[1] + Done
"/usr/bin/gdb" --interpreter=mi --tty=${DbgTerm} 0<"/tmp/Microsoft-MIEngine
-In-ctfwkxs0.pla" 1>"/tmp/Microsoft-MIEngine-Out-0gqkxdbp.q1w"
xcredo@xcredo-Pc:~/Documents/work1/Prog$
```


ПРИЛОЖЕНИЕ

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 }
```

Задание 2

```
1  #include <stdio.h>
2  typedef struct
3  {
4      char str[5];
5      int num;
6  } NumberRepr;
7  void format(NumberRepr *number)
8  {
9      sprintf(number->str, "%3d", number->num);
10 }
11 int main()
12 {
13     NumberRepr number = {.num = 1025};
14     format(&number);
15     printf("str: %s\n", number.str);
16     printf("num: %d\n", number.num);
17     return 0;
18 }
```

Задание 3

```
1  #include <stdio.h>
2  #define SQR(x) ((x) * (x))
3  int main()
4  {
5      int y = 5;
6      int z = SQR(y + 1);
7      printf("z = %d\n", z);
8      return 0;
9  }
```

Задание 4

```
1  #include <stdio.h>
2  void swap(int* a, int* b)
3  {
4      int tmp = *a;
5      *a = *b;
6      *b = tmp;
7  }
8  void bubble_sort(int* array, int size)
9  {
10     int i, j;
11     for (i = 0; i < size - 1; ++i)
12     {
13         for (j = 0; j < size - i - 1; ++j)
14         {
15             if (array[j] > array[j + 1])
16             {
17                 swap(&array[j], &array[j + 1]);
18             }
19         }
20     }
21 }
22 int main()
23 {
24     int array[100] = {10, 15, 5, 4, 21, 7};
25     bubble_sort(array, 6);
26     int i;
27     for (i = 0; i < 6 ; ++i)
28     {
29         printf("%d ", array[i]);
30     }
31     printf("\n");
32     return 0;
33 }
```