

Отчет по лабораторной работе № 10 по курсу “Фундаментальная информатика”

Студент группы М80-109Б-22 Любарский Иван Владимирович, № по списку 8

Контакты ivanred289@gmail.com, @rMeDGranD

Работа выполнена: «22» ноября 2022г.

Преподаватель: каф. 806 Сысоев Максим Алексеевич

Отчет сдан « » _____ 20__ г., итоговая оценка _____

Подпись преподавателя _____

1. **Тема:** Программирование на языке Си

2. **Цель работы:** Получить и исправить 6 ошибок (2 CE, 2 RE, 2 UB)

3. **Задание:** Получить и исправить 6 ошибок (2 CE, 2 RE, 2 UB)

4. **Оборудование:**

Процессор *Intel Core i5-4210U @ 4x 1.7GH* с ОП 15873 Мб, НМД 512 Гб. Монитор 1600x900

5. **Программное обеспечение:**

Операционная система семейства: *linux*, наименование: *ubuntu*, версия 22.04 LTS

интерпретатор команд: *bash* версия 4.4.19.

Система программирования -- версия --, редактор текстов *emacs* версия 25.2.2

Утилиты операционной системы --

Прикладные системы и программы --

Местонахождение и имена файлов программ и данных на домашнем компьютере --

6. **Идея, метод, алгоритм** решения задачи (в формах: словесной, псевдокода, графической [блок-схема, диаграмма, рисунок, таблица] или формальные спецификации с пред- и постусловиями)

Задача состоит в том, чтобы найти способы «поломать» программу на языке Си и дальше решить её.

7. **Сценарий выполнения работы** [план работы, первоначальный текст программы в черновике (можно на отдельном листе) и тесты либо соображения по тестированию].

План работы:

1. Вызвать ошибку.
2. Решить ошибку.

8. Распечатка протокола (подклеить листинг окончательного варианта программы с тестовыми примерами, подписанный преподавателем).

Ошибка СЕ — попытка изменения константы.

```
#include <stdio.h>
int main (){
    const int i = 0;
    for(i = 0; i < 10; i++) { printf("%i", i); }
    return 0;
}
```

TaskC.c: In function 'main':

TaskC.c:4:15: error: assignment of read-only variable 'i'

```
4 |     for(i = 0; i < 10; i++) { printf("%i", i); }
  |           ^
```

TaskC.c:4:29: error: increment of read-only variable 'i'

```
4 |     for(i = 0; i < 10; i++) { printf("%i", i); }
  |                               ^
```

Решение — изменить тип i.

```
#include <stdio.h>
int main (){
    int i = 0;
    for(i = 0; i < 10; i++) { printf("%i", i); }
    return 0;
}
```

Ошибка СЕ — неправильный тип аргумента в прототипе функции.

```
#include <stdio.h>
void DoAFlip(int a);
int main (){
    DoAFlip();
    return 0;
}
void DoAFlip(){ printf("%s", "Flip"); }
```

TaskC.c: In function 'main':

TaskC.c:4:9: error: too few arguments to function 'DoAFlip'

```
4 |     DoAFlip();
  |     ^~~~~~
```

TaskC.c:2:6: note: declared here

```
2 | void DoAFlip(int a);
  |     ^~~~~~
```

TaskC.c: In function 'DoAFlip':

TaskC.c:7:1: error: number of arguments doesn't match prototype

```
7 | void DoAFlip(){ printf("%s", "Flip"); }
  | ^~~~
```

TaskC.c:2:6: error: prototype declaration

```
2 | void DoAFlip(int a);
  |     ^~~~
```

Решение — удаление аргумента.

```
#include <stdio.h>
void DoAFlip();
int main (){
    DoAFlip();
    return 0;
}
```

```
}  
void DoAFlip(){ printf("%s", "Flip"); }
```

Ошибка RE — деление на 0.

```
#include <stdio.h>  
int main (){  
    int i = 1;  
    int j = 1;  
    printf("%i", i/(--j));  
    return 0;  
}
```

Решение — увеличить j.

```
#include <stdio.h>  
int main (){  
    int i = 1;  
    int j = 2;  
    printf("%i", i/(--j));  
    return 0;  
}
```

Ошибка RE — ошибка сегментирования из-за рекурсивного вызова функции main.

```
#include <stdio.h>  
int main (){  
    printf("%s", "What?");  
    main();  
    return 0;  
}
```

Решение — удалить и не использовать вызов функции main.

```
#include <stdio.h>  
int main (){  
    printf("%s", "What?");  
    return 0;  
}
```

Ошибка UB — выход за пределы массива.

```
#include <stdio.h>  
int main(){  
    int mas[1]={ 1 };  
    printf("%i", mas[1]);  
    return 0;  
}
```

Решение — правильное индексирование по массиву (1ый элемент не первый в индексе).

```
#include <stdio.h>  
int main(){  
    int mas[1]={ 1 };  
    printf("%i", mas[0]);  
    return 0;  
}
```

Ошибка UB — вывод неправильного типа.

```
#include <stdio.h>  
int main (){
```

```
int i = 10;
float j = 2.5;
printf("%i", i/j);
return 0;
}
```

Решение - изменить тип вывода на %f.

```
#include <stdio.h>
int main (){
    int i = 10;
    float j = 2.5;
    printf("%f", i/j);
    return 0;
}
```

9. Дневник отладки должен содержать дату и время сеансов отладки и основные события (ошибки в сценарии и программе, нестандартные ситуации) и краткие комментарии к ним. В дневнике отладки приводятся сведения об использовании других ЭВМ, существенном участии преподавателя и других лиц в написании и отладке программы.

№	Лаб. или дом.	Дата	Вре м я	Событие	Действие по исправлению	Примечание
0						

10. Замечания автора по существу работы

11. Выводы

Для выполнения работы потребовалось разобраться в видах ошибок и понять, что их может вызвать. В будущем, возможно, это понадобится для того, чтобы как раз таки избегать таких ошибок или хотя бы представлять, что они из себя представляют.

Недочёты при выполнении задания могут быть устранены следующим образом: --

Подпись студента _____