

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

Студент группы М80-109Б-22 Серый Никита Олегович, № по списку 16

Контакты:

email — [nikita.seryj@mail.ru](mailto:nikita.seryj@mail.ru)

telegram - @hukumkas

Работа выполнена: «4» декабря 2022г.

Преподаватель: каф. 806 Сысоев Максим Алексеевич

Отчет сдан «5» декабря 2022 г., итоговая оценка \_\_\_\_\_

Подпись преподавателя \_\_\_\_\_

## 1. Тема: Отладчик системы программирования

2. Цель работы: Обработка ошибок и отладка программы на языке C.

3. Задание: Составить программы на языке C с различными типами ошибок.

4. Оборудование (студента):

Процессор AMD Ryzen 5 5500U @ 6x 2.1GH с ОП 15360 Мб, НМД (?) 512 Гб. Монитор 1920x1080

5. Программное обеспечение (студента):

Операционная система семейства: *linux*, наименование: *mint*, версия *21 Cinnamon*

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)

Привести примеры программ, исполнение которых приводит к ошибке, а затем показать, как их нужно исправить.

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

СЕ (compilation error):

1. Лишняя скобка, допущенная по невнимательности, вызывает СЕ.

```
1 int main(){
2     printf("%d", 17283 + 5* 6*7*8);
3     return 0;
4 }
5
```

```
1 int main(){
2     printf("%d", 17283 + 5* 6*7*8);
3     return 0;
4 }
5
```

File	Line	Message
=== Build: Release in Cprogramm (compiler: GNU GCC Compiler) ===		
C:\Cprogram...		In function 'main':
C:\Cprogram... 2		warning: implicit declaration of function 'printf' [-Wimplicit-function-dec...
C:\Cprogram... 2		warning: incompatible implicit declaration of built-in function 'printf'
C:\Cprogram... 2		note: include '<stdio.h>' or provide a declaration of 'printf'
C:\Cprogram... 2		warning: format '%d' expects a matching 'int' argument [-Wformat=]
C:\Cprogram... 2		warning: right-hand operand of comma expression has no effect [-Wunused-value]
C:\Cprogram... 2		error: expected ';' before ')' token
C:\Cprogram... 2		error: expected statement before ')' token

## 2. First use in this function.

Возникает, когда забыли объявить переменную (в данном случае не объявили avg)

```
1  #include <stdio.h>
2
3  int main(){
4      float num1 = 0;
5      float num2 = 0;
6      float num3 = 0;
7
8      scanf("%f%f%f", &num1, &num2, &num3);
9
10     avg = (num1 + num2 + num3)/3;
11     printf("The avg value of 3 numbers is %f\nValue of avg is %d", avg, avg > 15);
12
13     return 0;
14 }
```

File	Line	Message
		=== Build: Release in Cprogramm (compiler: GNU GCC Compiler) ===
C:\Cprogram...		In function 'main':
C:\Cprogram...	10	error: 'avg' undeclared (first use in this function)
C:\Cprogram...	10	note: each undeclared identifier is reported only once for each function it...
		=== Build failed: 1 error(s), 0 warning(s) (0 minute(s), 0 second(s)) ===

```
1  #include <stdio.h>
2
3  int main(){
4      float num1 = 0;
5      float num2 = 0;
6      float num3 = 0;
7      float avg = 0;
8
9      scanf("%f%f%f", &num1, &num2, &num3);
10
11     avg = (num1 + num2 + num3)/3;
12     printf("The avg value of 3 numbers is %f\nValue of avg is %d", avg, avg > 15);
13
14     return 0;
15 }
```

RE (runtime error):

## 1. Бесконечная рекурсия.

Исправляется, если задать условие завершения функции (в моём примере его нет).

```
1  #include <stdio.h>
2
3  void f(){
4      f();
5  }
6
7
8  int main(){
9      f();
10
11     return 0;
12 }
13
```

```
1  #include <stdio.h>
2
3  int x = 2;
4
5  void f(){
6
7
8      --x;
9      if (x == 0) {
10         return ;
11     }
12     f();
13 }
14
15
16 int main(){
17     f();
18
19     return 0;
20 }
21
```

## 2. Бесконечный цикл (согласно стандарту программа должна либо завершаться, либо выводить значение).

```
1  #include <stdio.h>
2
3  int main(){
4      float km, mi;
5      float low, step, upper;
6
7      low = 0.0;
8      upper = 300.0;
9      step = 20.0;
10
11     km = low;
12
13     while (km <= upper)
14         printf("%0.2f : %0.2f\n", km, mi);
15         km += step;
16         mi += step*0.621;
17 }
```

```
0.00 : 0.00
0.00 : 0.00
0.00 : 0.00
0.00 : 0.00
0.00 : 0.00
0.00 : 0.00
0.00 : 0.00
0.00 : 0.00
0.00 : 0.00
0.00 : 0.00
0.00 : 0.00
0.00 : 0.00
0.00 : 0.00
0.00 : 0.00
0.00 : 0.00
```

```
1  #include <stdio.h>
2
3  int main(){
4      float km, mi;
5      float low, step, upper;
6
7      low = 0.0;
8      upper = 300.0;
9      step = 20.0;
10
11     km = low;
12
13     while (km <= upper){
14         printf("%0.2f : %0.2f\n", km, mi);
15         km += step;
16         mi += step*0.621;
17     }
18 }
19
```

```
0.00 : 0.00
20.00 : 12.42
40.00 : 24.84
60.00 : 37.26
80.00 : 49.68
100.00 : 62.10
120.00 : 74.52
140.00 : 86.94
160.00 : 99.36
180.00 : 111.78
200.00 : 124.20
220.00 : 136.62
240.00 : 149.04
260.00 : 161.46
280.00 : 173.88
300.00 : 186.30

Process returned 0 (0x0)   execution time : 0.063 s
Press any key to continue.
```

UB (undefined behavior):

## 1. Переменные без присвоенного значения

В результате исполнения программа выдаст неправильное значение, не соответствующее решению задачи. Чтобы исправить, присвоим 0 переменным.

```
1 #include <stdio.h>
2
3 int main() {
4
5     int kolvo;
6     int chislo;
7     int kolvo62;
8     int numb62;
9     int numb31;
10    int numb2;
11    int numb0;
12    int i;
13
14    scanf("%d", &kolvo);
15
16    for (int i = 0; i < kolvo; ++i) {
17        scanf("%d", &chislo);
18        if (chislo % 62 == 0) {
19            numb62 += 1;
20        }
21        else if (chislo % 31 == 0) {
22            numb31 += 1;
23        }
24        else if (chislo % 2 == 0) {
25            numb2 += 1;
```

```
1 #include <stdio.h>
2
3 int main() {
4
5     int kolvo = 0;
6     int chislo = 0;
7     int kolvo62 = 0;
8     int numb62 = 0;
9     int numb31 = 0;
10    int numb2 = 0;
11    int numb0 = 0;
12    int i = 0;
13
14    scanf("%d", &kolvo);
15
16    for (int i = 0; i < kolvo; ++i) {
17        scanf("%d", &chislo);
18        if (chislo % 62 == 0) {
19            numb62 += 1;
20        }
21        else if (chislo % 31 == 0) {
22            numb31 += 1;
23        }
24        else if (chislo % 2 == 0) {
25            numb2 += 1;
```

## 2. Целочисленное переполнение (выдаёт значение в дополнительном коде).

Выход — использовать значение в диапазоне от -32768 до 32767

```
1 #include <stdio.h>
2
3 int main() {
4
5     short int gamma = 52000;
6     printf("%hd", gamma );
7     return 0;
8
9 }
10
```

```
1 #include <stdio.h>
2
3 int main() {
4
5     short int gamma = 520;
6     printf("%hd", gamma );
7     return 0;
8
9 }
10
```

```
-13536
Process returned 0 (0x0)   execution time : 0.024 s
Press any key to continue.
```

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

**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
1	дом	20.11.22	2:02	Всё идёт по плану	-	Просматриваю коды своих задач

#### **10. Замечания автора** по существу работы

Замечаний по существу нет. Материалы для данной работы накопились сами собой благодаря решению других задач при подготовке к работам 8-9, что даёт небольшую разгрузку.

#### **11. Выводы**

Лабораторная работа научила меня внимательнее относиться к коду программы перед компиляцией. Этот навык пригодится при программировании в «полевых» условиях — при отсутствии под рукой IDE, в частности на экзамене.

Недочёты при выполнении задания могут быть устранены следующим образом: больше решать задач.

Подпись студента

---