**Отчет по лабораторной работе №** 8по курсу “Фундаментальная информатика”

Студент группы М80-103Б-21 Березнев Никита Вадимович, № по списку 4

Контакты e-mail: nikita.berezneff@yandex.ru

Работа выполнена: «22» октября 2021г.

Преподаватель: каф. 806 Севастьянов Виктор Сергеевич

Отчет сдан « » \_\_\_\_\_\_\_\_\_20\_\_ г., итоговая оценка \_\_\_\_\_\_\_\_

Подпись преподавателя \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* 1. **Тема:** Cистемы программирования на языке Си.

1. **Цель работы:** Изучение конкретной системы программирования на Си и получение навыков подготовки текстов и отладки программ.
2. **Задание А:** Найти и вывести сумму двх целых чисел, не меньших -32000 и не превосходящих 32000.

**Задание B:** Найти сумму последовательности из не более чем 40 целых чисел со значениями от -20000000 до 20000000.

**Задание C:** Найти сумму последовательности из не более чем 40 целых чисел со значениями от -20000000 до 20000000. Суммировать необходимо только числа, превосходящие или равные 10.

1. **Оборудование** (студента):

Процессор *Intel Core i3-3217U @ 4x 1.80GH* с ОП *7851* Мб, НМД *512* Гб. Монитор *1366x768*

1. **Программное обеспечение (**студента**):**

Операционная система семейства: *linux*, наименование: *ubuntu*, версия *18.10 cosmic*

интерпретатор команд: *bash* версия *4.4.19*.

Система программирования -- версия --

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).

- Изучить и освоить возможности лабораторной СП по содержащейся в практикуме документации и другим источникам, в том числе основные этапы процесса компиляции и подготовки программ к выполнению.

- Составить и отладить простейшую программу на Си в *терминальном классе*.

- Изучить различные системы программирования на Си на других платформах: GNU, MS Visual Studio и др. на *домашних компьютерах.*

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

- **Запуск редактора текстов** для редактирования или первоначального набора программы с помощью команды emacs.

- **Компиляция программы** (в ключах компилятора обязательно задать проверку соблюдения стандарта языка) и создание (в случае отсутствия синтаксических ошибок) выполнимой программы (a.out).

- При обнаружении синтаксических ошибок компилятор выдаёт сообщения, и необходимо вернуться к пункту 1 (возобновив emacs) для их исправления. Если ошибок много, то начинать исправления надо с первой ошибки, так как многие последующие ошибки часто являются следствием неправильного восстановления состояния компилятора после первых ошибок (так называемые *наведенные ошибки*). Если список ошибок не помещается в окно терминала, то его можно рассматривать по частям, прокручивая вывод терминала или перенаправляя вывод компилятора (поток ошибок) на пейджер *more*  (или *less*).

- В случае, если ошибок не было, **запуск исполняемого файла** a.out из *текущего* каталога для тестирования программы.

- В случае обнаружения семантических (смысловых) ошибок, необходимо проанализировать текст программы с целью их обнаружения. Если не удаётся найти ошибки по тексту программы, следует воспользоваться приёмами отладки с помощью ЭВМ.

- После исправления ошибок в исходном тексте программу следует **заново** *откомпилировать*.

- Составить ряд тестов, с помощью которых можно было бы проверить корректность работы составленного алгоритма.

Задача А

|  |  |
| --- | --- |
| **Входные данные** | **Выходные данные** |
| 0 0 | 0 |
| 123 45 | 168 |
| 12 31890 | 31902 |
| 13879 -17781 | -3902 |

Задача B

|  |  |
| --- | --- |
| **Входные данные** | **Выходные данные** |
| 4 1 2 3 4 | 10 |
| 3 -123 542 -78 | 341 |
| 2 0 0 | 0 |
| 8 -7983 7843 12 -1245 -12345 7843 -1345511 1 | -1351385 |

Задача C

|  |  |
| --- | --- |
| **Входные данные** | **Выходные данные** |
| 4 1 2 3 4 | 0 |
| 1 1000 | 1000 |
| 5 12 7 -20 16 278 | 306 |
| 8 -7983 7843 12 -1245 -12345 7843 -1345511 1 | 15698 |

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

Задача А

#include <stdio.h>

int main(void)

{

int a;

int b;

scanf("%d", &a);

scanf("%d", &b);

printf("%d\n", a + b);

return 0;

}

Задача B

#include <stdio.h>

int main(void)

{

int sum;

int x;

int k;

scanf("%d", &k);

sum = 0;

for (int i = 1 ; i <= k ; i++) {

scanf("%d", &x);

sum = sum + x;

}

printf("%d\n", sum);

return 0;

}

Задача C

#include <stdio.h>

int main(void)

{

int sum;

int k;

int x;

scanf("%d", &k);

sum = 0;

for (int i = 1 ; i <= k ; i++) {

scanf("%d", &x);

if (x >= 10) {

sum = sum + x;

}

}

printf("%d\n", sum);

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. | Дата | Время | Событие | Действие по исправлению | Примечание |
|  | или |  |  |  |  |  |
|  | дом. |  |  |  |  |  |
| 1 | дом | 22.10.21 | 3:51 | При прохождении чекера программа выпадала в ошибку “Нарушение правил оформления программ” | Редактирование кода программы согласно правилам оформления | Это был увлекательнейший час поиска лишних пробелов, за время которого я переслушал весь Black Album Metallica. |
| 2 | дом | 22.10.21 | 2:37 | При проверке программы с помощью чекера, на некотором тесте происходила ошибка компиляции | Указание void в качестве аргумента функции main | Если void – это пустота, то функция main(void) пустая внутри. Выходит функция main(void) – dead inside? Или Hollow function? Звучит как название метроидвании. |

1. **Замечания автора** по существу работы

Работу считаю крайне полезной (в конце концов, Си с нами до конца семестра) и очень интересной (ввиду того, что из всех ЯП я был знаком лишь с Паскалем, мне было очень интересно опробовать новый язык программирования). Создавая элементарные программы на Си я не только обрел новые знания в синтаксисе нового для меня языка, но и хорошо выучил правила оформления программ с помощью чекера (Да, без чекера жизнь была бы куда скучнее. И проще).

1. **Выводы**

Работа мне очень понравилась. Было очень занимательно писать алгоритм, “обкатывая” неизвестный ранее ЯП. Также весьма порадовал тот факт, что мы, изучив алгоритмические языки, перешли к работе с одним из самых распространённых ЯП. Благодаря данной работе я научился создавать простейшие программы, реализуемые на языке программирования Си. Знания, полученные мной в ходе выполнения данной лабораторной работы, помогут мне в дальнейшем при составлении более сложных алгоритмов.

Подпись студента \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_