**Отчет по лабораторной работе № 11** по курсу “Фундаментальная информатика”

Студент группы: М8О-106Б-21 Петров Илья Олегович, № по списку: 16 Контакты e-mail, telegram, skype: [gtgtr3000@gmail.com](mailto:gtgtr3000@gmail.com)

Работа выполнена: «17» декабря 2021 г.

Преподаватель: каф. 806 Дубинин Алексей Владимирович Отчет сдан « » \_ 20 г., итоговая оценка

Подпись преподавателя

## Тема:

Обработка последовательности литер входного текстового файла. Простейшие приемы лексического анализа. Диаграммы состояний и переходов.

## Цель работы:

Составить программу на языке Си, выполняющую анализ и обработку вводимого текста в соответствии с выданным преподавателем вариантом задания. В программе не должно быть ограничений на кол-во и длину строк исходного текста.

1. **Задание** (*вариант №* **33):**

Выделить все десятичные числа от 17 до 77 по модулю и распечатать их значения в словесной форме по-английски.

1. **Оборудование** (студента): ----------

## Программное обеспечение (лабораторное): ---------

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

Алгоритм:

- Считать число из входного потока при его наличии, иначе ждать появления числа.

- Если оно не соответствует ограничениям – вернуться к началу.

- Если соответствует – проверить цифру десятка и вывести определенное слово на английском.

Далее проверить остаток от деления на 10 и вывести соответствующее слово на английском.

- Начать алгоритм сначала.

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

Написать программу выбора чисел из входного потока.

Написать функцию выбора необходимых чисел.

Продумать и написать функцию вывода значения в словесной форме по-английски.

Протестировать обе функции:

checkNum – если ввести восьмеричное число в диапазоне от 17 до 77 по модулю включая концы – должно вернуть 1, иначе 0

printValue – для каждого числа, которому соответствует 1 из предыдущей ыункции должно выводиться соответствующее число на английском, для проверки будет достаточно проверить число 17, от 20 до 27, числа 30 40 50 60 и 70.

1

*Пункты 1-7 отчета составляются строго до начала лабораторной работы.*

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

#include <stdio.h>  
#include <stdlib.h>  
#include <ctype.h>  
enum states {  
 *not\_num*,  
 *is\_num*,  
 *is\_sign*};  
int space(int c) {  
 return c == ' ' || c == '\n' || c == '\t' || c == EOF || c == ',';  
}  
void printValue(int v) {  
 v = abs(v);  
 if (v == 17) {  
 printf("Seventeen\n");  
 return;  
 }  
 switch (v / 10) {  
 case 2:printf("Twenty"); break;  
 case 3:printf("Thirty"); break;  
 case 4:printf("Forty"); break;  
 case 5:printf("Fifty"); break;   
 case 6:printf("Sixty"); break;   
 case 7:printf("Seventy"); break;   
 }  
 switch (v % 10) {  
 case 0:printf("\n"); break;   
 case 1:printf(" one\n"); break;   
 case 2:printf(" two\n"); break;   
 case 3:printf(" three\n"); break;   
 case 4:printf(" four\n"); break;   
 case 5:printf(" five\n"); break;   
 case 6:printf(" six\n"); break;   
 case 7:printf(" seven\n"); break;   
 }  
}  
int checkNum(int n) {  
 if (abs(n) < 17)  
 return 0;  
 if (abs(n) / 10 < 1 || abs(n) / 10 > 7)  
 return 0;  
 if (abs(n) % 10 > 7)  
 return 0;  
 return 1;  
}  
void build\_int(int \*p\_int, int c) {  
 (\*p\_int) = (\*p\_int) \* 10 + (c - '0');  
}  
int main() {  
 int c;  
 enum states state;  
 int num = 0;  
 do {  
 c = getchar();  
 if (space(c))  
 state = *not\_num*;  
 else if (isdigit(c))  
 state = *is\_num*;  
 else if (c == '+' || c == '-')  
 state = *is\_sign*;  
 else  
 state = *not\_num*;  
  
 switch (state) {  
 case *not\_num*: {  
 if (num && checkNum(num))  
 printValue(num);  
 num = 0; break;  
 }  
 case *is\_num*: {  
 build\_int(&num, c); break;  
 }  
 default: { break; }  
 }  
 } while (c != EOF);  
 return 0;  
}

1. **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 | Время | Событие | Действие по исправлению | Примечание |
| 1 | Дом. | 17.12.21 | 14:00 | Все тесты пройдены | Не требуется. |  |

1. **Замечания автора** по существу работы: отсутствуют.
2. **Выводы**

Проделав данную лабораторную работу, я вспомнил принцип работы switch-case оператора, а также научился потоковой обработке входных данных. Также были улучшены навыки по разбитию задачи на подзадачи и отдельном тестировании функций.

Недочёты при выполнении задания могут быть устранены следующим образом:

Подпись студента