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

Студент группы М80-109Б-22 Серякова Александра Андреевна, № 17

Работа выполнена: «02» октября 2022г.

Преподаватель: каф. 806 Сысоев Максим Алексеевич

Отчет сдан « » _____20__ г., итоговая оценка _____

Подпись преподавателя _____

1. **Тема:** Алгоритмы и структуры данных

2. **Цель работы:** Составить программу на языке Си, выполняющую анализ и обработку вводимого текста в соответствии с выданным преподавателем вариантом задания.

3. **Задание** (*вариант №* 29)**:** 29. Удалить все десятичные числа, не превышающие INT_MAX

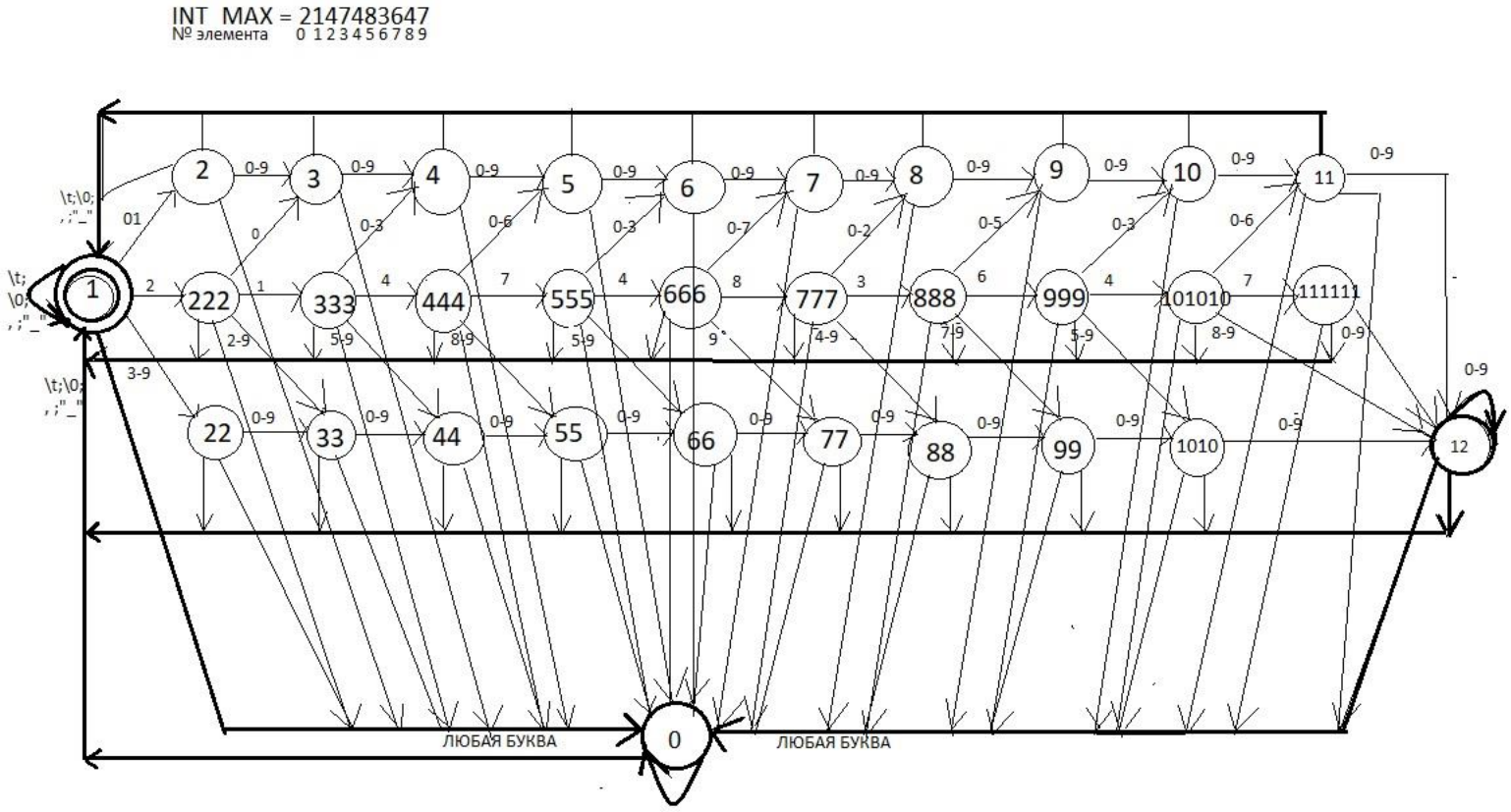
4. **Оборудование** (студента):

   Процессор AMD Ryzen 5 5500U with Radeon Graphics @ 2.100GHz с ОП 9812 Мб, SSD 512 Гб. Монитор 1920x1080

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

   Операционная система семейства: linux, наименование: Arch x86_64
   интерпретатор команд: bash версия 5.1.16
   Система программирования -- версия --, редактор текстов neo vim версия 0.7.2
   Утилиты операционной системы mkdir, cd, touch, ls, echo, cat, find, grep, rm, chmod, bash, pwd
   Прикладные системы и программы –
   Местонахождение и имена файлов программ и данных на домашнем компьютере /home/taida/Programming/MAI_labs/lab5

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



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

Проводилось unit-тестирование, исходный код тестов приложен в пункте №8.

Тесты:

описание	Входные данные	результат
Когда слова-это числа через пробел	Input: 4567887656789 4567656 567 4 4567765678767	4567887656789 4567765678767
Когда слова-это буквы и через пробел	Input: rtyuiutghjkghjk rtyu f fghjk	rtyuiutghjkghjk rtyu f fghjk
Когда и слова, и числа через пробел	Input: r456 5 567 45678765677 56776ty	r456 45678765677 56776ty
Когда и слова, и числа через запятую и пробел	Input: 456765fg,45676567656776567,677 567898 fgh56787 , 45678567876545677	456765fg 45676567656776567 fgh56787 45678567876545677

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

```c
#include <stdio.h>
#include <stdlib.h>
#include <limits.h>
#include <string.h>

int all_number(char c)
{
    return (c>= '0' && c<='9');
}

int less_number(char c, char num_arr)
{
    return (c>= '0' && c<='9' && c< num_arr);
}

int more_number(char c, char num_arr)
{
    return (c>= '0' && c<='9' && c> num_arr);
}

int equal_number(char c, char num_arr)
{
    return (c>= '0' && c<='9' && c== num_arr);
}

int gap(char c)
{
    return (c==' ' || c== '\t' || c==',' || c== '\n');
}

int bukv(char c)
{
    return (!(all_number(c))&& !(gap(c)));
}

int main(void)
{
    printf("%d\n", INT_MAX);
    int int_max = INT_MAX;

    int i= 0;
    int tarr[10] = {0};
    int arr[10] = {};

    while (int_max != 0)
    {
        tarr[i] = int_max % 10;// ЗАПИШЕМ ОСТАТОК ОТ ЭТОГО ЧИСЛА> ЭТО ЧИСЛО ПЕРЕВОРАЧИВАЕМ И ОБРАТНЫЙ ОСТАТОК В МАСС
        ++i;
        int_max /= 10;
    }
    for (; i!= 0; --i)
    {
        arr[10 -i] = tarr[i-1]; // ВЫРОВНЯЕМ ЭТОТ МАСС В arr
    }
    printf("Input: ");

    int state = 1;
    int k;
    char symbol;
    char all[20] = " "; //ТУТ БУДУТ ХРАНИТЬСЯ ВСЕ ЧИСЛА
    int top = 0;//top - ЭТО ИНДЕКС ЭЛЕМЕНТА all
    while ((symbol = getchar()) != EOF){
        switch (state) {
            case 1:
                if (less_number(symbol,arr[0])){
                    top = 0;
                    memset (all, ' ', 20-1);
                    all[top] = symbol;
                    ++top;
                    state = 2;
                }else if (equal_number(symbol, arr[0])){
                    top = 0;
                    memset (all, ' ', 20-1);
                    all[top] = symbol;
                    ++top;
                    state = 222;
                }else if (more_number(symbol, arr[0])){
                    top = 0;
                    memset (all, ' ', 20-1);
                    all[top] = symbol;
                    ++top;
                    state = 22;
                }else if (gap(symbol)){
                    state = 1;
                }else if (bukv(symbol)){
                    top = 0;
                    memset (all, ' ', 20-1);
                    all[top] = symbol;
                    ++top;
                    state = 0;
                }

                break;
            case 0:
                if (gap(symbol)){
                    printf("%s\n", all);
                    state= 1;

                }else{
```

```
                    all[top] = symbol;
                    ++top;
                    state = 0;
                }
                break;
            case 2:
                if (all_number(symbol)){
                    all[top] = symbol;
                    ++top;
                    state = 3;
                }else if(gap(symbol)){
                    state = 1;
                }else if (bukv(symbol)){
                    all[top] = symbol;
                    ++top;
                    state = 0;
                }
                break;
            case 222:
                if (less_number(symbol,arr[1])){
                    all[top] = symbol;
                    ++top;
                    state = 3;
                }else if (equal_number(symbol, arr[1])){
                    all[top] = symbol;
                    ++top;
                    state = 333;
                }else if (more_number(symbol, arr[1])){
                    all[top] = symbol;
                    ++top;
                    state = 33;
                }else if(gap(symbol)){
                    state = 1;
                }else if (bukv(symbol)){
                    all[top] = symbol;
                    ++top;
                    state = 0;
                }
                break;
            case 22:
                if (all_number(symbol)){
                    all[top] = symbol;
                    ++top;
                    state = 33;
                }else if(gap(symbol)){
                    state = 1;
                }else if (bukv(symbol)){
                    all[top] = symbol;
                    ++top;
                    state = 0;
                }
                break;

            case 3:
                if (all_number(symbol)){
                    all[top] = symbol;
                    ++top;
                    state = 4;
                }else if(gap(symbol)){
                    state = 1;
                }else if (bukv(symbol)){
                    all[top] = symbol;
                    ++top;
                    state = 0;
                }
                break;
            case 333:
                if (less_number(symbol,arr[2])){
                    all[top] = symbol;
                    ++top;
                    state = 4;
                }else if (equal_number(symbol, arr[2])){
                    all[top] = symbol;
                    ++top;
                    state = 444;
                }else if (more_number(symbol, arr[2])){
                    all[top] = symbol;
                    ++top;
                    state = 44;
                }else if(gap(symbol)){
                    state = 1;
                }else if (bukv(symbol)){
                    all[top] = symbol;
                    ++top;
                    state = 0;
                }
                break;
            case 33:
                if (all_number(symbol)){
                    all[top] = symbol;
                    ++top;
                    state = 44;
                }else if(gap(symbol)){
                    state = 1;
                }else if (bukv(symbol)){
                    all[top] = symbol;
                    ++top;
                    state = 0;
                }
                break;
```

```
            case 4:
                if (all_number(symbol)){
                    all[top] = symbol;
                    ++top;
                    state = 5;
                }else if(gap(symbol)){
                    state = 1;
                }else if (bukv(symbol)){
                    all[top] = symbol;
                    ++top;
                    state = 0;
                }
                break;
            case 444:
                if (less_number(symbol,arr[3])){
                    all[top] = symbol;
                    ++top;
                    state = 5;
                }else if (equal_number(symbol, arr[3])){
                    all[top] = symbol;
                    ++top;
                    state = 555;
                }else if (more_number(symbol, arr[3])){
                    all[top] = symbol;
                    ++top;
                    state = 55;
                }else if(gap(symbol)){
                    state = 1;
                }else if (bukv(symbol)){
                    all[top] = symbol;
                    ++top;
                    state = 0;
                }
                break;
            case 44:
                if (all_number(symbol)){
                    all[top] = symbol;
                    ++top;
                    state = 55;
                }else if(gap(symbol)){
                    state = 1;
                }else if (bukv(symbol)){
                    all[top] = symbol;
                    ++top;
                    state = 0;
                }
                break;

            case 5:
                if (all_number(symbol)){
                    all[top] = symbol;
                    ++top;
                    state = 6;
                }else if(gap(symbol)){
                    state = 1;
                }else if (bukv(symbol)){
                    all[top] = symbol;
                    ++top;
                    state = 0;
                }
                break;
            case 555:
                if (less_number(symbol,arr[4])){
                    all[top] = symbol;
                    ++top;
                    state = 6;
                }else if (equal_number(symbol, arr[4])){
                    all[top] = symbol;
                    ++top;
                    state = 666;
                }else if (more_number(symbol, arr[4])){
                    all[top] = symbol;
                    ++top;
                    state = 66;
                }else if(gap(symbol)){
                    state = 1;
                }else if (bukv(symbol)){
                    all[top] = symbol;
                    ++top;
                    state = 0;
                }
                break;
            case 55:
                if (all_number(symbol)){
                    all[top] = symbol;
                    ++top;
                    state = 66;
                }else if(gap(symbol)){
                    state = 1;
                }else if (bukv(symbol)){
                    all[top] = symbol;
                    ++top;
                    state = 0;
                }
                break;

            case 6:
                if (all_number(symbol)){
                    all[top] = symbol;
```

```
                    ++top;
                    state = 7;
                }else if(gap(symbol)){
                    state = 1;
                }else if (bukv(symbol)){
                    all[top] = symbol;
                    ++top;
                    state = 0;
                }
                break;
            case 666:
                if (less_number(symbol,arr[5])){
                    all[top] = symbol;
                    ++top;
                    state = 7;
                }else if (equal_number(symbol, arr[5])){
                    all[top] = symbol;
                    ++top;
                    state = 777;
                }else if (more_number(symbol, arr[5])){
                    all[top] = symbol;
                    ++top;
                    state = 77;
                }else if(gap(symbol)){
                    state = 1;
                }else if (bukv(symbol)){
                    all[top] = symbol;
                    ++top;
                    state = 0;
                }
                break;
            case 66:
                if (all_number(symbol)){
                    all[top] = symbol;
                    ++top;
                    state = 77;
                }else if(gap(symbol)){
                    state = 1;
                }else if (bukv(symbol)){
                    all[top] = symbol;
                    ++top;
                    state = 0;
                }
                break;

            case 7:
                if (all_number(symbol)){
                    all[top] = symbol;
                    ++top;
                    state = 8;
                }else if(gap(symbol)){
                    state = 1;
                }else if (bukv(symbol)){
                    all[top] = symbol;
                    ++top;
                    state = 0;
                }
                break;
            case 777:
                if (less_number(symbol,arr[6])){
                    all[top] = symbol;
                    ++top;
                    state = 8;
                }else if (equal_number(symbol, arr[6])){
                    all[top] = symbol;
                    ++top;
                    state = 888;
                }else if (more_number(symbol, arr[6])){
                    all[top] = symbol;
                    ++top;
                    state = 88;
                }else if(gap(symbol)){
                    state = 1;
                }else if (bukv(symbol)){
                    all[top] = symbol;
                    ++top;
                    state = 0;
                }
                break;
            case 77:
                if (all_number(symbol)){
                    all[top] = symbol;
                    ++top;
                    state = 88;
                }else if(gap(symbol)){
                    state = 1;
                }else if (bukv(symbol)){
                    all[top] = symbol;
                    ++top;
                    state = 0;
                }
                break;

            case 8:
                if (all_number(symbol)){
                    all[top] = symbol;
                    ++top;
                    state = 9;
                }else if(gap(symbol)){
                    state = 1;
```

```
            }else if (bukv(symbol)){
                all[top] = symbol;
                ++top;
                state = 0;
            }
            break;
        case 888:
            if (less_number(symbol,arr[7])){
                all[top] = symbol;
                ++top;
                state = 9;
            }else if (equal_number(symbol, arr[7])){
                all[top] = symbol;
                ++top;
                state = 999;
            }else if (more_number(symbol, arr[7])){
                all[top] = symbol;
                ++top;
                state = 99;
            }else if(gap(symbol)){
                state = 1;
            }else if (bukv(symbol)){
                all[top] = symbol;
                ++top;
                state = 0;
            }
            break;
        case 88:
            if (all_number(symbol)){
                all[top] = symbol;
                ++top;
                state = 99;
            }else if(gap(symbol)){
                state = 1;
            }else if (bukv(symbol)){
                all[top] = symbol;
                ++top;
                state = 0;
            }
            break;

        case 9:
            if (all_number(symbol)){
                all[top] = symbol;
                ++top;
                state = 10;
            }else if(gap(symbol)){
                state = 1;
            }else if (bukv(symbol)){
                all[top] = symbol;
                ++top;
                state = 0;
            }
            break;
        case 999:
            if (less_number(symbol, arr[8])){
                all[top] = symbol;
                ++top;
                state = 10;
            }else if (equal_number(symbol, arr[8])){
                all[top] = symbol;
                ++top;
                state = 101010;
            }else if (more_number(symbol, arr[8])){
                all[top] = symbol;
                ++top;
                state = 1010;
            }else if(gap(symbol)){
                state = 1;
            }else if (bukv(symbol)){
                all[top] = symbol;
                ++top;
                state = 0;
            }
            break;
        case 99:
            if (all_number(symbol)){
                all[top] = symbol;
                ++top;
                state = 1010;
            }else if(gap(symbol)){
                state = 1;
            }else if (bukv(symbol)){
                all[top] = symbol;
                ++top;
                state = 0;
            }
            break;

        case 10:
            if (all_number(symbol)){
                all[top] = symbol;
                ++top;
                state = 11;
            }else if(gap(symbol)){
                state = 1;
            }else if (bukv(symbol)){
                all[top] = symbol;
                ++top;
                state = 0;
```

```
                }
                break;
            case 101010:
                if (less_number(symbol,arr[9])){
                    all[top] = symbol;
                    ++top;
                    state = 11;
                }else if (equal_number(symbol, arr[9])){
                    all[top] = symbol;
                    ++top;
                    state = 111111;
                }else if (more_number(symbol, arr[9])){
                    all[top] = symbol;
                    ++top;
                    state = 12;
                }else if(gap(symbol)){
                    state = 1;
                }else if (bukv(symbol)){
                    all[top] = symbol;
                    ++top;
                    state = 0;
                }
                break;
            case 1010:
                if (all_number(symbol)){
                    all[top] = symbol;
                    ++top;
                    state = 12;
                }else if(gap(symbol)){
                    state = 1;
                }else if (bukv(symbol)){
                    all[top] = symbol;
                    ++top;
                    state = 0;
                }
                break;

            case 11:
                if (all_number(symbol)){
                    all[top] = symbol;
                    ++top;
                    state = 12; // в 12 состоянии число > max_int
                }else if(gap(symbol)){
                    //top = top_old;
                    state = 1; // меньше или равно max_int => переходим к следующему числу

                }else if (bukv(symbol)){
                    //top = top_old;
                    all[top] = symbol;
                    ++top;
                    state = 0;
                }
                break;
            case 111111:
                if (all_number(symbol)){
                    all[top] = symbol;
                    ++top;
                    state = 12; // в 12 состоянии число > max_int
                }else if(gap(symbol)){
                    state = 1; // в 13 состоянии число меньше или равно max_int
                }else if (bukv(symbol)){
                    all[top] = symbol;
                    ++top;
                    state = 0;
                }
                break;

            case 12:
                if (all_number(symbol)){
                    all[top] = symbol;
                    ++top;
                    state = 12; // в 12 состоянии число > max_int
                }else if(gap(symbol)){
                    printf("%s\n", all);
                    state = 1;// число полностью считано, переходим к 1 состоянию, чтобы считывать следующее число
                }else if (bukv(symbol)){
                    all[top] = symbol;
                    ++top;
                    state = 0;
                }
                break;

        }
    }
    return 0;
}
```

Тест:

```
void unit_test(){
    assert(all_number('2') == 1);
    assert(all_number('D') == 0);
    assert(all_number(' ') == 0);
    assert(all_number('/') == 0);
    assert(less_number('1', '2') == 1);
    assert(less_number('6', '2') == 0);
    assert(less_number('v', '2') == 0);
    assert(more_number('6', '3') == 1);
    assert(more_number('0', '5') == 0);
    assert(more_number('d', '2') == 0);
    assert(equal_number('4', '4') == 1);
    assert(equal_number('1', '4') == 0);
    assert(equal_number('6', '4') == 0);
    assert(equal_number('h', '4') == 0);
    assert(equal_number('\t', '4') == 0);
    assert(gap(' ') == 1);
    assert(gap('f') == 0);
    assert(bukv('h') == 1);
    assert(bukv('0') == 0);
}

int main(void)
{
    unit_test();
```

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
Дом	31.12.22	16:00	AAaa пытаюсь создать из цифр MAX_INT список(((Сложно. Толи Си ничего не умеет, толи я.	получается, что список получился в обратном порядке(теперь нужно как-то сделать реверс списка, заносив каждый элемент в новый	Грустно, решила, что ничего не умеет ни Си, а я(

10. Замечания автора по существу работы

11. Выводы

В этой лабораторной работе было много боли, пришлось «перелопатить» кучу сайтиков, чтобы понять, как добавлять в си элемент в конец списка, разобраться, как полностью очистить списки (memset ,кстати, очень пригодилась, чтобы каждый раз очищать список, если слово мне не подошло и его не надо выводить). В целом лабораторная мне понравилась, кажется, мы потихоньку находим общий язык с Си ☺

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

Подпись студента _____