

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

Студент группы М80-109Б-22 Горохов Михаил Сергеевич, № по списку 4

Контакты e-mail: mgorohow@yandex.ru, telegram: @mcgoroh

Работа выполнена: «24» декабря 2022г.

Преподаватель: каф. 806 Сысоев Максим Алексеевич

Отчет сдан «24» декабря 2022 г., итоговая оценка _____

Подпись преподавателя _____

1. **Тема:** программирование на языке C.
2. **Цель работы:** составление и отладка простейшей программы на языке C с конечными автоматами.
3. **Задание (вариант № 40):** выделить беззнаковые восьмеричные числа и напечатать их цифры в двоичной системе.
4. **Оборудование** (студента):

Процессор *Intel Core i5-7200U CPU @ 2.50 GHz* с ОП 4 Гб, AMD Nainan, 1024 Гб. Монитор 1920x1080

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

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

интерпретатор команд: *bash* версия *4.4.19*.

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

6. Идея, метод, алгоритм

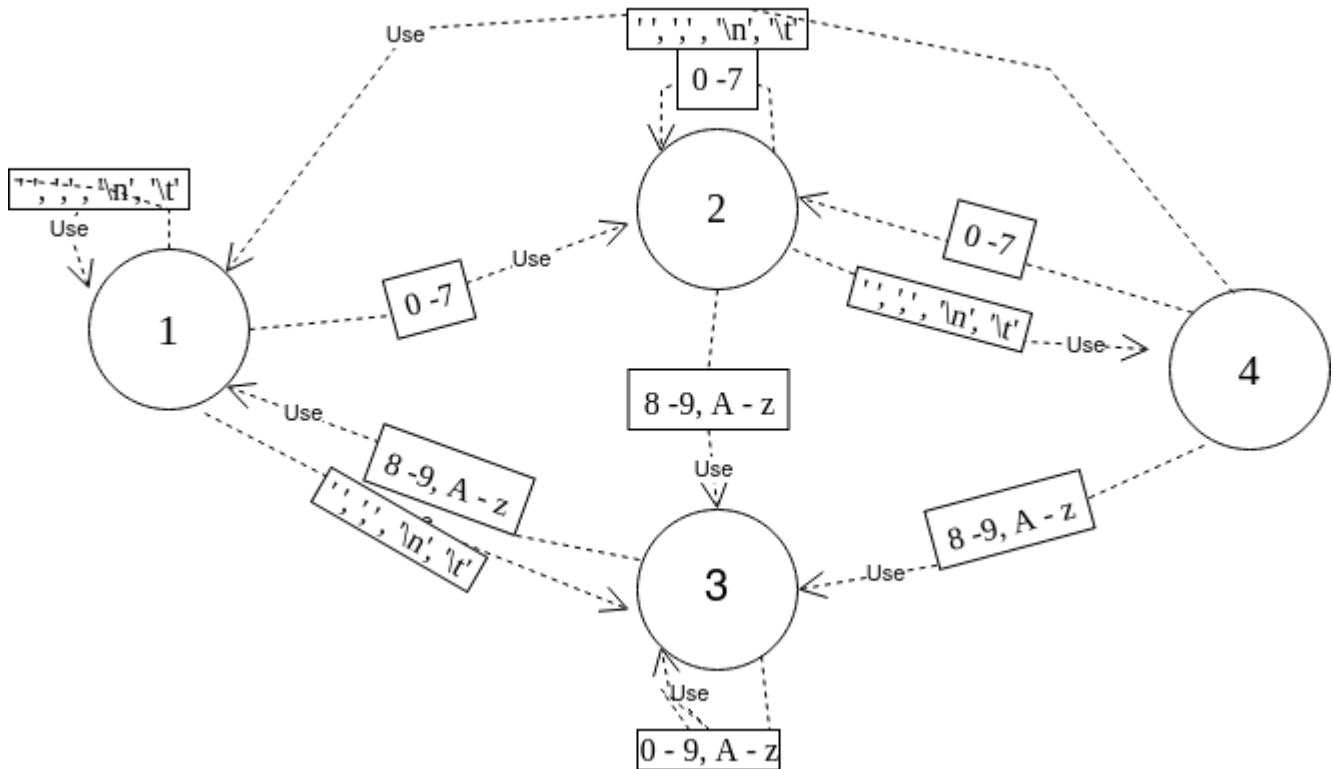
Найти восьмеричные числа, записать посимвольно в массив и потом перевести цифры в двоичные.

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

1. Понять условие.
2. Сделать конечный автомат.
3. Реализовать его на Си.
4. Исключить единичные ошибки.

Входные данные	Выходные данные	Описание тестируемого случая
123 45ret *	1 10 11	Одно число подходит, а другое с буквами
123 45r 45 890 *	1 10 11 100 101	Два числа подходят еще одно с буквами и последнее не подходит(с 8 и 9)

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



```
#include <stdio.h>
#include <assert.h>
```

```
int ascii_to_int(int symbol){
    if (symbol == 48){
        return 0;
    }
    else if (symbol == 49){
        return 1;
    }
    else if (symbol == 50){
        return 2;
    }
    else if (symbol == 51){
        return 3;
    }
    else if (symbol == 52){
        return 4;
    }
    else if (symbol == 53){
        return 5;
    }
    else if (symbol == 54){
        return 6;
    }
    else if (symbol == 55){
        return 7;
    }
}
```

```
void print_binary(int count, int array[50]){
    for (int k = 0; k < count; ++k){
        switch (array[k]) {
            case 0: printf("0 "); break;
            case 1: printf("1 "); break;
            case 2: printf("10 "); break;
            case 3: printf("11 "); break;
            case 4: printf("100 "); break;
```

```

        case 5: printf("101 "); break;
        case 6: printf("110 "); break;
        case 7: printf("111 "); break;
    }
}
printf("\n");
}

```

```

void test_ascii_to_int(){
    assert(ascii_to_int(48) == 0);
    assert(ascii_to_int(49) == 1);
    assert(ascii_to_int(50) == 2);
    assert(ascii_to_int(51) == 3);
    assert(ascii_to_int(52) == 4);
    assert(ascii_to_int(53) == 5);
    assert(ascii_to_int(54) == 6);
    assert(ascii_to_int(55) == 7);
}

```

```

int main(){
    test_ascii_to_int();

    int state = 1, count = 0;
    int symbol;

    while((symbol = getchar()) != '*'){
        int array[50];
        switch (state) {
            case 1:
                if (symbol >= '0' && symbol <= '7'){
                    array[count] = ascii_to_int(symbol);
                    count += 1;
                    state = 2;
                }
                else if ((symbol >= '8' && symbol <= '9') || (symbol >= 'A' && symbol <= 'Z') || (symbol >= 'a' && symbol <= 'z')){
                    state = 5;
                }
                else if (symbol == ' ' || symbol == ',' || symbol == '\n' || symbol == '\t'){
                    count = 0;
                    state = 1;
                }
                break;

            case 2:
                if (symbol >= '0' && symbol <= '7'){
                    array[count] = ascii_to_int(symbol);
                    count += 1;
                    state = 2;
                }
                else if ((symbol >= '8' && symbol <= '9') || (symbol >= 'A' && symbol <= 'Z') || (symbol >= 'a' && symbol <= 'z')){
                    state = 3;
                }
                else if (symbol == ' ' || symbol == ',' || symbol == '\n' || symbol == '\t'){
                    print_binary(count, array);
                    count = 0;
                    state = 4;
                }
                break;

            case 3:
                if ((symbol >= '0' && symbol <= '9') || (symbol >= 'A' && symbol <= 'Z') || (symbol >= 'a' && symbol <= 'z')){
                    state = 3;
                }
                else if (symbol == ' ' || symbol == ',' || symbol == '\n' || symbol == '\t'){
                    state = 1;
                }
            }
        }
    }
}

```

```

    }
    break;

case 4:
    if (symbol >= '0' && symbol <= '7'){
        count = 0;
        array[count] = ascii_to_int(symbol);
        count += 1;
        state = 2;
    }
    else if ((symbol >= '8' && symbol <= '9') || (symbol >= 'A' && symbol <= 'Z') || (symbol >= 'a' && symbol <= 'z')){
        count = 0;
        state = 3;
    }
    else if (symbol == ' ' || symbol == ',' || symbol == '\n' || symbol == '\t'){
        count = 0;
        state = 1;
    }
    break;

default:
    break;
}

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 мя	Событие	Действие по исправлению	Примечание

10. Замечания автора

Замечания отсутствуют.

11. Выводы

Как же долго я сидел над пониманием этого автомата, надеюсь все таки правильно, последняя лаба, но интересная на самом деле. Когда все понимаешь, становится все красиво. Зачти пожалуйста)

Подпись студента
