**Петряев Владимир Александрович КТбо1-8  
Индивидуальное задание №2 по ПиОТА-2(МТ)**

# Постановка задачи.

На входной ленте машины Тьюринга задано целое неотрицательное число в системе счисления с основанием 3.

Поставить справа от числа знак # и записать представление этого числа в системе счисления с основанием 9.

# Словесное описание

Машине на вход поступает лента с числом в 3-й системе исчисления, проходя до конца числа, после него машина записывает символ «#», после начинается перевод в 9-ю систему.

Пока мы не встретим цифры «0, 1, 2» мы двигаемся влево от «#», встретив их мы берем два символа заменяя их на буквы «a, b, c» соответственно.

Заменив две крайних справа цифры на буквы и перейдя в соответствующее паре цифр состояние

{q6 = 00 = 0; q7 = 01 = 1; q8 = 02 = 2; q9 = 10 = 3; q10 = 11 = 4; q11 = 12 = 5; q12 = 20 = 6; q13 = 21 = 7; q14 = 22 = 8;}, двигаемся вправо до символа «#», после чего начинаем записывать «с нуля» или же перезаписывать число, сдвигая каждую цифру вправо, за счёт переходов по состояниям машины.

Повторяем это пока слева от «#» не останутся только буквы, после начинаем заменять буквы обратно на цифры.

Последним действием возвращаем управляющее устройство в начальное положение и завершаем работу машины.

# Формальное описание автомата.

V = {0, 1, 2}

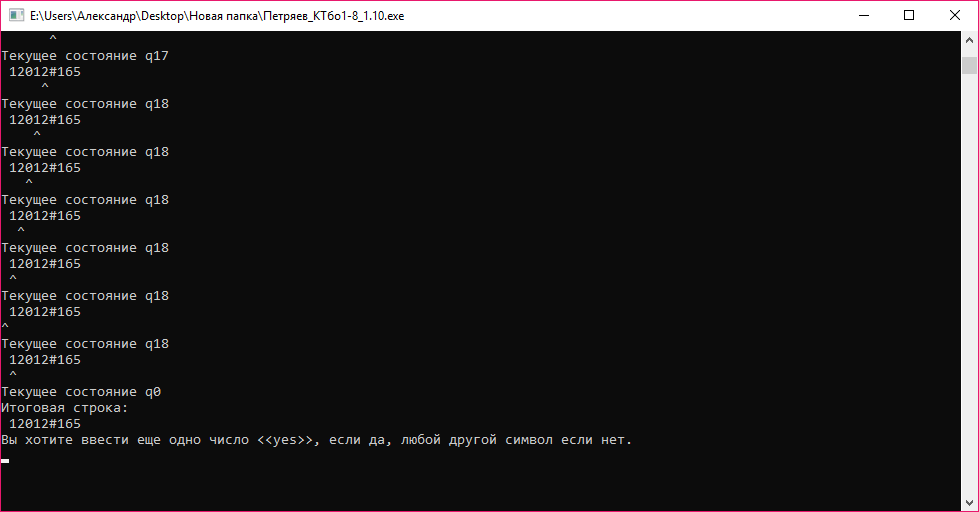
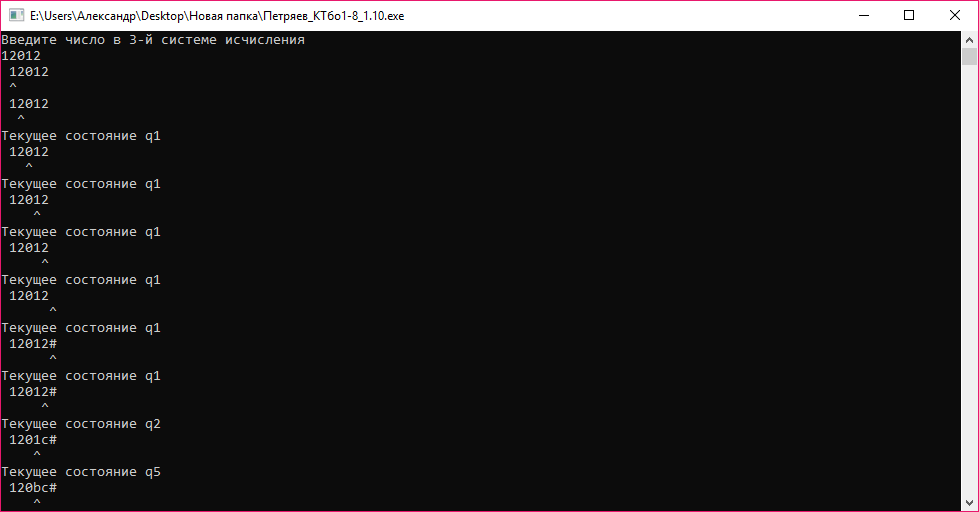
W = {0, 1, 2, 3, 4, 5, 6, 7, 8, #, λ}

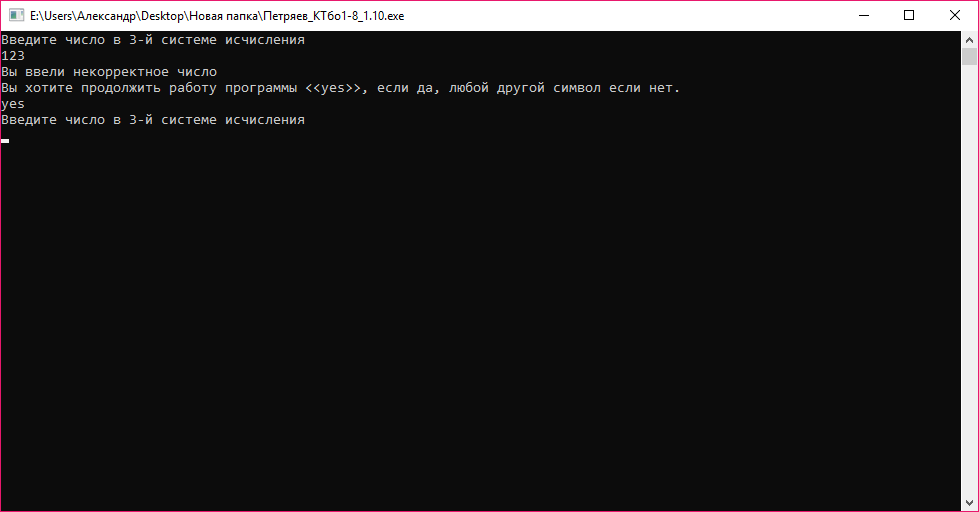
A = {a, b, c}

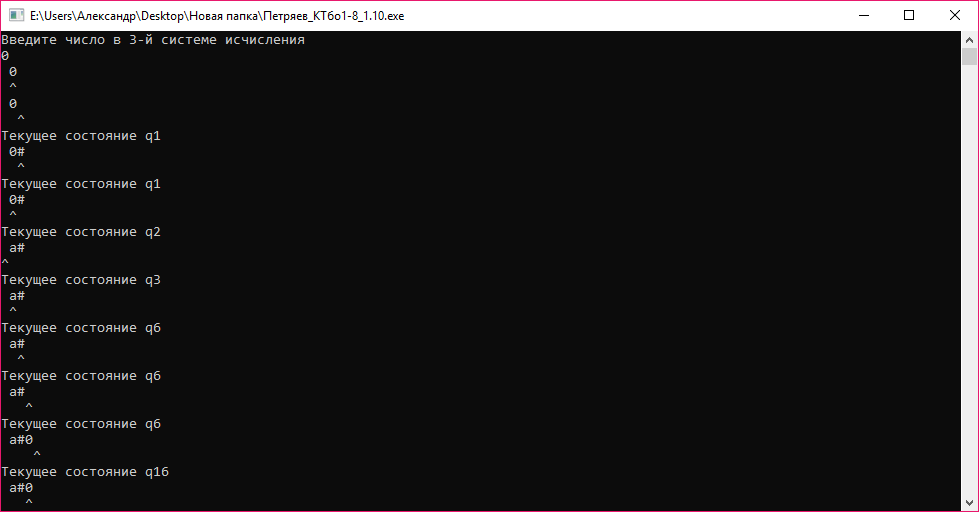
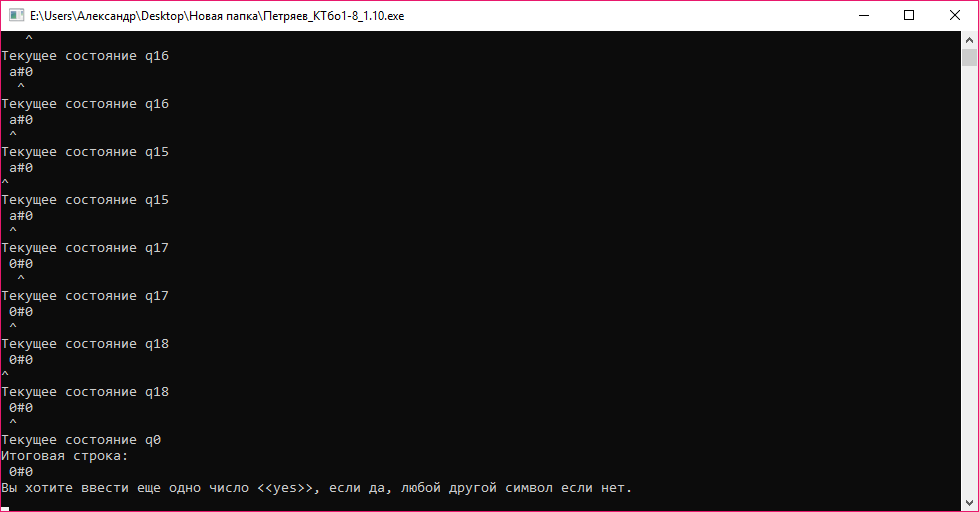
# Система команд машины Тьюринга в виде таблицы

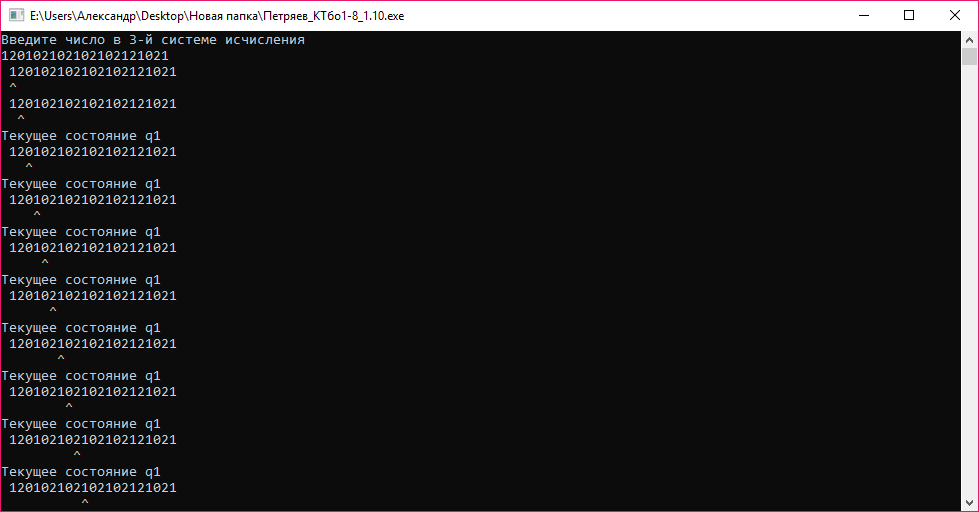
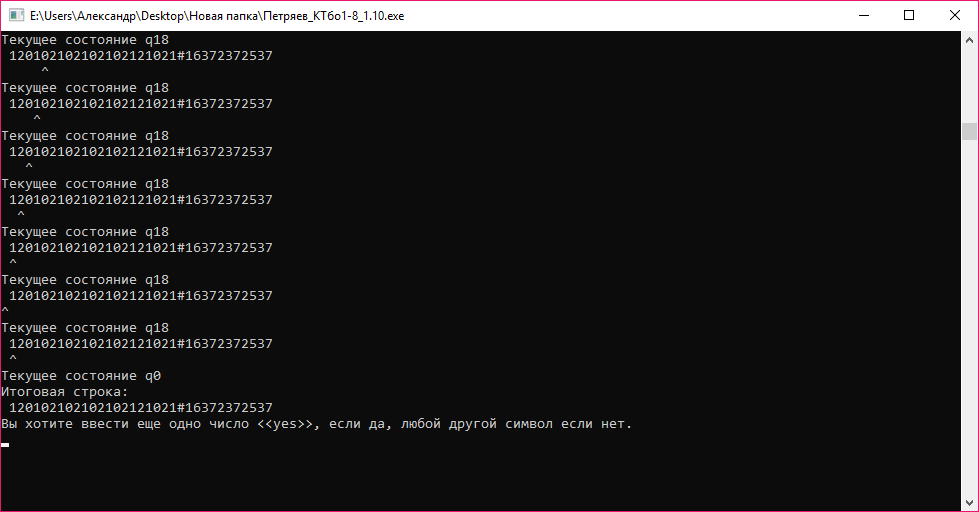
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | q1 | q2 | q3 | q4 | q5 | q6 | q7 | q8 | q9 | q10 | q11 | q12 | q13 | q14 | q15 | q16 | q17 | q18 |
| 0 | П | a, Л, 3 | a, Н, 6 | a, Н, 7 | a, Н, 8 | 0, П, 6 | 1, П, 6 | 2, П, 6 | 3, П, 6 | 4, П, 6 | 5, П, 6 | 6, П, 6 | 7, П, 6 | 8, П, 6 | Н, 2 | Л |  | Л |
| 1 | П | b, Л, 4 | b, Н, 9 | b, Н, 10 | b, Н, 11 | 0, П, 7 | 1, П, 7 | 2, П, 7 | 3, П, 7 | 4, П, 7 | 5, П, 7 | 6, П, 7 | 7, П, 7 | 8, П, 7 | Н, 2 | Л |  | Л |
| 2 | П | c, Л, 5 | c, Н, 12 | c, Н, 13 | c, Н, 14 | 0, П, 8 | 1, П, 8 | 2, П, 8 | 3, П, 8 | 4, П, 8 | 5, П, 8 | 6, П, 8 | 7, П, 8 | 8, П, 8 | Н, 2 | Л |  | Л |
| 3 |  |  |  |  |  | 0, П, 9 | 1, П, 9 | 2, П, 9 | 3, П, 9 | 4, П, 9 | 5, П, 9 | 6, П, 9 | 7, П, 9 | 8, П, 9 |  | Л |  |  |
| 4 |  |  |  |  |  | 0, П, 10 | 1, П, 10 | 2, П, 10 | 3, П, 10 | 4, П, 10 | 5, П, 10 | 6, П, 10 | 7, П, 10 | 8, П, 10 |  | Л |  |  |
| 5 |  |  |  |  |  | 0, П, 11 | 1, П, 11 | 2, П, 11 | 3, П, 11 | 4, П, 11 | 5, П, 11 | 6, П, 11 | 7, П, 11 | 8, П, 11 |  | Л |  |  |
| 6 |  |  |  |  |  | 0, П, 12 | 1, П, 12 | 2, П, 12 | 3, П, 12 | 4, П, 12 | 5, П, 12 | 6, П, 12 | 7, П, 12 | 8, П, 12 |  | Л |  |  |
| 7 |  |  |  |  |  | 0, П, 13 | 1, П, 13 | 2, П, 13 | 3, П, 13 | 4, П, 13 | 5, П, 13 | 6, П, 13 | 7, П, 13 | 8, П, 13 |  | Л |  |  |
| 8 |  |  |  |  |  | 0, П, 14 | 1, П, 14 | 2, П, 14 | 3, П, 14 | 4, П, 14 | 5, П, 14 | 6, П, 14 | 7, П, 14 | 8, П, 14 |  | Л |  |  |
| a |  |  |  |  |  | П | П | П | П | П | П | П | П | П | Л | Л | П |  |
| b |  |  |  |  |  | П | П | П | П | П | П | П | П | П | Л | Л | П |  |
| c |  |  |  |  |  | П | П | П | П | П | П | П | П | П | Л | Л | П |  |
| # | Л, 2 |  |  |  |  | П | П | П | П | П | П | П | П | П |  | Л, 15 | Л, 18 |  |
| λ | #, Н |  | П, 6 | П, 7 | П, 8 | 0, П, 16 | 1, П, 16 | 2, П, 16 | 3, П, 16 | 4, П, 16 | 5, П, 16 | 6, П, 16 | 7, П, 16 | 8, П, 16 | П, 17 | Л |  | П, 0 |

# Набор тестов





# Описание программной реализации

Лента МТ реализована при помощи переменной типа std::string.

Структура tablitsa используется для хранения действий осуществляемых управляющим устройством.

**Алгоритм работы** int main():

Функцией vvod\_tabl() считывается таблица переходов реализованная при помощи map<int, map<char, tablitsa>>.

В консоль выводится сообщение "Введите число в 3-й системе исчисления".

Считывается введенная пользователем строка.

Строка передается в функцию chek\_str.

Если chek\_str возвращает 0, выводятся два сообщения

"Вы ввели некорректное число"

"Вы хотите продолжить работу программы <<yes>>, если да, любой другой символ если нет."

Снова считывается строка.

Если пользователь вводит «yes» перезапускается функция main, если же любой другой символ (сочетание символов) программа заканчивает работу с кодом 0.

Если chek\_str возвращает 1.

Запускается функция obr которая возвращает итоговую строку.

Выводится сообщение "Итоговая строка:" и строка которую вернула функция obr.

Дальше пользователю предлагается продолжить работу программы или закончить её.

Выводится сообщение "Вы хотите ввести еще одно число <<yes>>, если да, любой другой символ если нет."

Считывается строка.

Если пользователь вводит «yes» перезапускается функция main, если же любой другой символ (сочетание символов) программа заканчивает работу с кодом 0.

**Алгоритм работы** map<int, map<char, tablitsa>> vvod\_tabl():

Открывается файл perehodi.txt.

kolvo\_command раз считываются q, simb, pech, sd, a.

Из-за особенности считывания из файла через fstream пробел записан в файл как символ “p”, если

simb или pech равны “p” они приравниваются к пробелу.

Дальше записывается элемент структуры tablitsa t.

Если состояние при считывании не менялось simb передается в качестве ключа в map<char, tablitsa> pretabl, а t в качестве его пары.

Если состояние при считывании изменилось, предыдущее состояние записывается в качестве ключа в map<int, map<char, tablitsa>> tabl, а pretabl в качестве его пары, после чего pretabl очищается и в него добавляются новые считанные значения.

Закрывается считываемый файл.

Возвращается переменная tabl.

**Алгоритм работы** bool chek\_str(string a):

В функцию передается введенная пользователем строка и проверяется на корректность.

Мы проходим по длине всей строки циклом for, если встречаем символ отличный от {0, 1, 2} возвращаем 0, иначе возвращаем 1.

**Алгоритм работы** string obr(string a, map<int, map<char, tablitsa>> tabl):

В функцию передаются введенная пользователем строка (еще при передаче в функцию слева и справа добавляются пробелы) и таблица переходов.

Изначально положение на ленте и состояние равны единице.

Создается итератор iter для tabl.

Выводится начальная строка и положение управляющего устройства в виде “ ^ ”.

Дальше пока мы не достигнем состояния ноль производится обработка строки.

Считывается символ, на котором находится управляющее устройство.

iter приравнивается к tabl.find(sost), мы находим в tabl наше состояние.

Происходит проверка, если выводимый символ не является пробелом, а символ, на котором находится управляющее устройство, им является, справа к строке добавляется пробел.

Символ, на котором находится управляющее устройство, заменяется на символ, который должен быть напечатан по условиям таблицы.

Происходит проверка движения по ленте если r, то положение увеличивается на единицу, если l, то положение уменьшается на единицу.

Меняется состояние на следующее состояние в соответствии с условием таблицы.

Выводится начальная строка и положение управляющего устройства в виде “ ^ ”.

В следующей строке выводится "Текущее состояние q" и номер состояния автомата.

**Структура файла** perehodi.txt:

176 строк содержат описание состояний автомата, 5 символов разделенным пробелом.

Первый символ число, состояние автомата в данный момент.

Второй символ, символ алфавита машины, который будет считываться с ленты.

Третий символ строки, символ записываемый на ленту.

Четвертый символ описывает движение по ленте, буквы r, l, s.

Пятый символ, состояние в которое переходит автомат после выполнения действий с лентой.

# Листинг программы

//--------------------------------------------------------

// ЮФУ, ИКТИБ, МОПЭВМ

// Программирование и основы теории алгоритмов - 2

// Индивидуальное задание 2 - МТ

// КТбо1-8, Петряев Владимир Александрович

// 07.05.2022

//--------------------------------------------------------

#define \_CRT\_SECURE\_NO\_WARNINGS

#include<iostream>

#include<string>

#include<fstream>

#include<map>

#include <windows.h>

using namespace std;

struct tablitsa // Структура для хранения действий машины

{

char pechat; // записываемый на ленту символ

char sdvig; // символ описывающий движение ленты

int next\_sost; // сдледующее состояние автомата

};

bool chek\_str(string a) // Функция проверки корректности ввода строки (передаваетмое значение - введенная пользователем строка)

{ // Возвращаемое значение - 1, если строка корректна, 0, если строка некорректна.

for (int i = 0; i < a.length(); i++)

{

if ((a[i] == '0') or (a[i] == '1') or (a[i] == '2'))

{

}

else

return 0;

}

return 1;

}

map<int, map<char, tablitsa>> vvod\_tabl() // Функция ввода таблицы переходов

{ // Вовращаемое значение таблица переходов МТ

ifstream nani;

tablitsa t;

map<char, tablitsa> pretabl;

map<int, map<char, tablitsa>> tabl;

char simb, pech, sd;

int kolvo\_command = 176;

int q, a, pre\_q = 1;

nani.open("perehodi.txt");// Вводы таблицы переходов из файла

for (int i = 0; i < kolvo\_command; i++)

{

nani >> q >> simb >> pech >> sd >> a;

if (simb == 'p')

simb = ' ';

if (pech == 'p')

pech = ' ';

t.pechat = pech;

t.sdvig = sd;

t.next\_sost = a;

if (pre\_q == q)

{

pretabl.emplace(simb, t);

}

else

{

tabl.emplace(pre\_q, pretabl);

pre\_q = q;

pretabl.clear();

pretabl.emplace(simb, t);

}

}

nani.close();

return tabl;

}

string obr(string a, map<int, map<char, tablitsa>> tabl) //Функция обработки строки

{ // Передаваемые значения

// a - строка введенная пользователем ограниченная пробелами

int pol = 1; // tabl - таблица переходов для МТ

int sost = 1; // Возвращаемое значение - итоговая строка после ее обработки МТ

char sim;

map<int, map<char, tablitsa>> ::iterator iter = tabl.begin();

for (int i = 0; i < a.size(); i++)

cout << a[i];

cout << endl;

for (int i = 0; i < pol; i++)

cout << ' ';

cout << "^";

cout << endl;

while (sost != 0)

{

sim = a[pol];

iter = tabl.find(sost);

if ((((\*iter).second[sim].pechat != ' ') & (sim == ' ')))

{

a += ' ';

}

a[pol] = (\*iter).second[sim].pechat;

if ((\*iter).second[sim].sdvig == 'r')

pol++;

else if ((\*iter).second[sim].sdvig == 'l')

pol--;

sost = (\*iter).second[sim].next\_sost;

for (int i = 0; i < a.size(); i++)

cout << a[i];

cout << endl;

for (int i = 0; i < pol; i++)

cout << ' ';

cout << "^";

cout << endl;

cout << "Текущее состояние q" << sost << endl;

}

return a;

}

int main()

{

setlocale(LC\_ALL, "ru");

map<int, map<char, tablitsa>> tabl;

string str;

tabl = vvod\_tabl();

cout << "Введите число в 3-й системе исчисления" << endl;

cin >> str;

if (chek\_str(str))

{

str = obr(' ' + str + ' ', tabl);

cout << "Итоговая строка:" << endl;

cout << str << endl;

}

else

{

cout << "Вы ввели некорректное число" << endl;

cout << "Вы хотите продолжить работу программы <<yes>>, если да, любой другой символ если нет." << endl;

cin >> str;

if (str == "yes")

return main();

else

return 0;

}

cout << "Вы хотите ввести еще одно число <<yes>>, если да, любой другой символ если нет." << endl;

cin >> str;

if (str == "yes")

return main();

else

return 0;

}