**Московский авиационный институт**

(национальный исследовательский университет)

Институт №8 «Информационные технологии и прикладная математика»

Кафедра 806 «Вычислительная математика и программирование»

Курсовой проект

по курсу «Вычислительные системы»

1 семестр

Задание 3.

Тема: «Программирование в алгоритмической модели Маркова»

|  |  |
| --- | --- |
| **Студент:** | Терентьев М.А. |
| **Группа:** | М8О - 101Б - 23 |
| **Преподаватель:** | Крылов С.С |
| **Подпись:** |  |
| **Оценка:** |  |

Москва 2023

**Оглавление**

[Задание. 2](https://docs.google.com/document/d/1lc655IqNC2tV3y1nvQ2gzOPpxzjwqaKA/edit#heading=h.gjdgxs)

[Формулировка. 2](https://docs.google.com/document/d/1lc655IqNC2tV3y1nvQ2gzOPpxzjwqaKA/edit#heading=h.30j0zll)

[Вариант задания. 3](https://docs.google.com/document/d/1lc655IqNC2tV3y1nvQ2gzOPpxzjwqaKA/edit#heading=h.1fob9te)

[Работа. 3](https://docs.google.com/document/d/1lc655IqNC2tV3y1nvQ2gzOPpxzjwqaKA/edit#heading=h.3znysh7)

Код программы[.](https://docs.google.com/document/d/1lc655IqNC2tV3y1nvQ2gzOPpxzjwqaKA/edit#heading=h.2et92p0) [3](https://docs.google.com/document/d/1lc655IqNC2tV3y1nvQ2gzOPpxzjwqaKA/edit#heading=h.3znysh7)

Описание работы алгоритма.4

Сложностная оценка алгоритма[.](https://docs.google.com/document/d/1lc655IqNC2tV3y1nvQ2gzOPpxzjwqaKA/edit#heading=h.tyjcwt) 5

Тесты[.](https://docs.google.com/document/d/1lc655IqNC2tV3y1nvQ2gzOPpxzjwqaKA/edit#heading=h.3dy6vkm) 5

[Вывод.](https://docs.google.com/document/d/1lc655IqNC2tV3y1nvQ2gzOPpxzjwqaKA/edit#heading=h.1t3h5sf) 5

# **Задание.**

## **Формулировка.**

Составить программу в алгоритмической модели Маркова, выполняющую заданное действие над словами, записанными на ленте. Использование дополнительных букв в ограниченном количестве допускается.

В конечном слове никаких несобственных букв остаться не должно.

## **Вариант задания.**

7. Входное слово представляет собой десятичную запись целого неотрицательного числа в прямой кодировке. Получить дополнительную кодировку для отрицательного числа с тем же абсолютным значением.

# **Работа.**

## **Код программы.**

"e -> .", # конец программы

"+$0)#[0] -> 00000000", # исключительная ситуация - введен 0

"00000000 -> 00000000e",

"$0<#[ -> +$0)#[", # переход из 1 части во 2

"$-0<#[ -> -$0)#[", # переход из 2 части в 3

# декрементирование числа a

"9<# -> 8>#",

"8<# -> 7>#",

"7<# -> 6>#",

"6<# -> 5>#",

"5<# -> 4>#",

"4<# -> 3>#",

"3<# -> 2>#",

"2<# -> 1>#",

"1<# -> 0>#",

"0<# -> ^<9#",

"$-1^< -> $>-",

">- -> ->",

"$1^< -> $>",

"9^< -> 8>",

"8^< -> 7>",

"7^< -> 6>",

"6^< -> 5>",

"5^< -> 4>",

"4^< -> 3>",

"3^< -> 2>",

"2^< -> 1>",

"1^< -> 0>",

"0^< -> ^<9",

">#[ -> #>[",

"#>[ -> #[>",

"0>] -> 1<]",

# инкрементирование числа b

">#[ -> #[>",

"[1\*< -> [\*<0",

"[\*<0 -> [<10",

">0] -> <1]",

">1] -> \*<0]",

"0\*< -> <1",

"1\*< -> \*<0",

">1] -> <10]",

# переход в конец слова a

" < -> >",

">< -> >",

">0 -> 0>",

">0 -> 0>",

">1 -> 1>",

">2 -> 2>",

">3 -> 3>",

">4 -> 4>",

">5 -> 5>",

">6 -> 6>",

">7 -> 7>",

">8 -> 8>",

">9 -> 9>",

# переход из конца слова b

"1< -> <1",

"0< -> <0",

"#[< -> <#[",

">0] -> <0]",

# запись числа b справа, которое будет в итоге двоичным,

"0> -> 0<#[0]",

"1> -> 1<#[0]",

"2> -> 2<#[0]",

"3> -> 3<#[0]",

"4> -> 4<#[0]",

"5> -> 5<#[0]",

"6> -> 6<#[0]",

"7> -> 7<#[0]",

"8> -> 8<#[0]",

"9> -> 9<#[0]",

# ---достраивание числа, вторая часть---

"0)#[ -> 0#[),",

# увеличение счетчика

"$0( -> $1)",

"$1( -> $2)",

"$2( -> $3)",

"$3( -> $4)",

"$4( -> $5)",

"$5( -> $6)",

"$6( -> $7)",

"$7( -> $8)",

# счетчик

")#[ -> #[)",

")1 -> 1)",

")0 -> 0)",

"),1 -> (1,",

"),0 -> (0,",

"1( -> (1",

"0( -> (0",

"#[( -> (#[",

"),] -> a]",

# --- дописавание нулей, третья часть ---

# уменьшение счетчика

"$0a -> $8}",

"$1a -> $7}",

"$2a -> $6}",

"$3a -> $5}",

"$4a -> $4}",

"$5a -> $3}",

"$6a -> $2}",

"$7a -> $1}",

# перемещение a

"1a -> a1",

"0a -> a0",

"#[a -> a#[",

# достраивание нулей

"8}#[ -> 0b#[00000000",

"7}#[ -> 0b#[0000000",

"6}#[ -> 0b#[000000",

"5}#[ -> 0b#[00000",

"4}#[ -> 0b#[0000",

"3}#[ -> 0b#[000",

"2}#[ -> 0b#[00",

"1}#[ -> 0b#[0",

"}1 -> }01",

# --- четвертая часть, дополнительный код ---

"+$0b -> $0+b",

"-$0b -> $0-b",

"+b#[0 -> #[0+b",

"-b#[0 -> #[1-b",

# инвертируем значение, если число отрицательное

"-b0 -> 1-b",

"-b1 -> 0-b",

"-b] -> c]",

# иначе - завершаем

"$0#[0+b -> +b",

"+b1 -> 1+b",

"+b0 -> 0+b",

"+b] -> e",

# прибавили единицу и вызвали сборщик мусора

"0c] -> 1k]",

"1c] -> \*0",

"1\* -> \*0",

"0\* -> 1k",

# концовка, очистка мусора

"$0#[k -> e",

"k] -> k",

"0k -> k0", # переносим сборщик мусора в начало

"1k -> k1",

"- -> $->", # запись коретки в начале слова

" -> $>", # запись коретки в начале слова

## **Описание алгоритма.**

В программе можно выделить 5 основных частей: перевод десятичного числа в двоичное, вычисление его разрядности, достраивание числа до 8 разрядов, перевод в дополнительный код, перевод в обратный код

Перевод десятичного числа в двоичное осуществляется с помощью декрементирования исходного числа в десятичной системе счисления и инкрементирования нуля в двоичной системе счисления.

Вычисление разрядности числа реализовано путем добавления двух доп символов – “)” и “,”. Первый доп символ сдвигает второй на один разряд и инкрементирует обнуленное значение в месте, где появился ноль после перевода десятичного числа.

Затем из 8 вычитается полученное число разрядов, записывается в то же место и с помощью схожей идеи достраиваются нули в начале двоичного числа.

Как только построено двоичное 8 разрядное число, число переводится в дополнительный код путем инверсии его цифр, а затем в обратный – прибавление единицы к полученному числу.

## **Сложностная оценка алгоритма.**

Алгоритма имеет сложность порядка O(n) относительно величины первого числа.

## **Тесты.**

Для проверки работоспособности алгоритма, было проведено его тестирование на следующем наборе тестовых данных:

1. -10 = 11110110
2. -125 = 10000011
3. -99 = 10011101
4. 0 = 00000000
5. 99 = 1100011

Данный набор тестовых данных был успешно пройден алгоритмом.

# **Вывод**

В ходе выполнения данного задания курсового проекта я освоил программирование простейших алгоритмов в алгоритмической модели Маркова. Научился использовать дополнительные буквы, для обработки последовательно расположенных букв входного слова.