Московский авиационный институт

(национальный исследовательский университет)

Институт №8 «Информационные технологии и прикладная математика»

Кафедра 806 «Вычислительная математика и программирование»

Курсовая работа

по курсу «Фундаментальная информатика»

1 семестр

Задание 2 «Программирование в алгоритмической модели Маркова»

Студент:	Дробышев Е. П.
Группа:	М8О-114БВ-24
Преподаватель:	Никулин С.П.
Подпись:	
Оценка:	
Дата сдачи:	1.11.2024

Содержание

Введение и формулировка задания

Программирование с использованием алгоритмической модели Маркова — это один из ключевых подходов для описания и выполнения вычислительных операций через формальные грамматики и правила замены. Эта модель, предложенная выдающимся русским математиком Андреем Марковым, позволяет строить алгоритмы, используя простые правила преобразования, что дает возможность эффективно анализировать и моделировать вычислительные процессы.

В данной курсовой работе мы исследуем, как с помощью модели Маркова можно решить задачу перевода чисел из одной системы счисления в другую — а именно из четверичной в шестнадцатеричную.

Вариант задания

Вариант 33: Составить алгоритм перевода числа из четверичной системы счисления в шестнадцатиричную.

Использованное оборудование и ПО

Оборудование ПЭВМ студента (лабораторное):

Процессор Intel Core i5, ОП 8ГБ, SSD 256ГБ, монитор 1920x1080 ~60Hz. Другие устройства не использовались.

Программное обеспечение ПЭВМ студента (лабораторное):

Операционная система семейства Linux, наименование Ubuntu версия 24.04

Интерпретатор команд GNU bash версия 5.2.21(1).

Редактор текстов: emacs версия 27.2

Утилиты операционной системы: ls, cd

Прикладные системы и программы: emacs, nam

Местоположение файлов /home/tru

Описание алгоритма

Выполним перевод числа из 4 в 2 систему счисления, а затем из 2 в 16 систему счисления:

1. Введем символ «*» для начала движения вправо и заменим его на символ «/».
2. Далее при переходе вправо выполним соответствующий правилам (/0 -> 0/ и т.д.) перевод из 4 в 2 систему счисления.

   При первом переводе исходного числа в 4 систему счисления переведем его в 2 систему счисления, используя следующие значения:

4 сс	2 сс
0	0
1	1
2	10
3	11

3. По окончании первого перевода заменим символ «/» на «\», двигаясь влево, и начнем перевод из 2 в 16 систему счисления по соответствующим правилам (1111\ -> \F, и т.д.). Во избежание ошибок при переводе начнем перевод с последнего знака.

   Второй перевод из 2 системы счисления в 16 систему счисления, начиная с последнего знака, используя следующие значения:

2 сс	16 сс
1111	F
1110	E
1101	D
1100	C
1011	B
1010	A
1001	9
1000	8
111	7
110	6
101	5

100	4
11	3
10	2
1	1
0	0

4. После второго перевода заменим символ «\», стоящий перед полученным числом на « ». На рабочей строке будет выведен результат перевода из 4 в 16 систему счисления.

Код программы

	Образец		Замена
1	*0	→	*
2	*1	→	/1
3	*2	→	/2
4	*3	→	/3
5	/0	→	0/
6	/1	→	1/
7	/2	→	10/
8	/3	→	11/
9	/	→	\
10	1111\	→	\F
11	1110\	→	\E
12	1101\	→	\D
13	1100\	→	\C
14	1011\	→	\B
15	1010\	→	\A
16	1001\	→	\9
17	1000\	→	\8
18	111\	→	\7
19	110\	→	\6
20	101\	→	\5
21	100\	→	\4
22	11\	→	\3
23	10\	→	\2
24	1\	→	\1
25	0\	→	\0
26	\1	→	1
27	\2	→	2
28	\3	→	3
29	\4	→	4
30	\5	→	5
31	\6	→	6
32	\7	→	7
33	\8	→	8
34	\9	→	9
35	\10	→	A
36	\11	→	B
37	\12	→	C
38	\13	→	D
39	\14	→	E
40	\15	→	F
41		→	*

Протокол выполнения программы

213 -> 27

Протокол замен

```
41: "" -> "*"
    "213" -> "*213"
3: "*2" -> "/2"
    "*213" -> "/213"
7: "/2" -> "10/"
    "/213" -> "10/13"
6: "/1" -> "1/"
    "10/13" -> "101/3"
8: "/3" -> "11/"
    "101/3" -> "10111/"
9: "/" -> "\"
    "10111/" -> "10111\"
18: "111\" -> "\7"
    "10111\" -> "10\7"
23: "10\" -> "\2"
    "10\7" -> "\27"
27: "\2" -> "2"
    "\27" -> "27"
```

01213 -> 67

```
41: "" -> "*"
    "01213" -> "*01213"
1: "*0" -> "*"
    "*01213" -> "*1213"
2: "*1" -> "/1"
    "*1213" -> "/1213"
6: "/1" -> "1/"
    "/1213" -> "1/213"
7: "/2" -> "10/"
    "1/213" -> "110/13"
6: "/1" -> "1/"
    "110/13" -> "1101/3"
8: "/3" -> "11/"
    "1101/3" -> "110111/"
9: "/" -> "\"
    "110111/" -> "110111\"
18: "111\" -> "\7"
    "110111\" -> "110\7"
19: "110\" -> "\6"
    "110\7" -> "\67"
31: "\6" -> "6"
    "\67" -> "67"
```

322 -> 3A

```
41: "" -> "*"
    "322" -> "*322"
4: "*3" -> "/3"
    "*322" -> "/322"
8: "/3" -> "11/"
    "/322" -> "11/22"
7: "/2" -> "10/"
    "11/22" -> "1110/2"
7: "/2" -> "10/"
    "1110/2" -> "111010/"
9: "/" -> "\"
    "111010/" -> "111010\"
15: "1010\" -> "\A"
    "111010\" -> "11\A"
22: "11\" -> "\3"
    "11\A" -> "\3A"
28: "\3" -> "3"
    "\3A" -> "3A"
```

123013 -> 6C7

```
41: "" -> "*"
    "123013" -> "*123013"
2: "*1" -> "/1"
    "*123013" -> "/123013"
6: "/1" -> "1/"
    "/123013" -> "1/23013"
7: "/2" -> "10/"
    "1/23013" -> "110/3013"
8: "/3" -> "11/"
    "110/3013" -> "11011/013"
5: "/0" -> "00/"
    "11011/013" -> "1101100/13"
6: "/1" -> "1/"
    "1101100/13" -> "11011001/3"
8: "/3" -> "11/"
    "11011001/3" -> "1101100111/"
9: "/" -> "\"
    "1101100111/" -> "1101100111\"
18: "111\" -> "\7"
    "1101100111\" -> "1101100\7"
13: "1100\" -> "\C"
    "1101100\7" -> "110\C7"
19: "110\" -> "\6"
    "110\C7" -> "\6C7"
31: "\6" -> "6"
    "\6C7" -> "6C7"
```

2322020 -> 2E88

```
41: "" -> "*"
    "2322020" -> "*2322020"
3: "*2" -> "/2"
    "*2322020" -> "/2322020"
7: "/2" -> "10/"
    "/2322020" -> "10/322020"
8: "/3" -> "11/"
    "10/322020" -> "1011/22020"
7: "/2" -> "10/"
    "1011/22020" -> "101110/2020"
7: "/2" -> "10/"
    "101110/2020" -> "10111010/020"
5: "/0" -> "00/"
    "10111010/020" -> "1011101000/20"
7: "/2" -> "10/"
    "1011101000/20" -> "101110100010/0"
5: "/0" -> "00/"
    "101110100010/0" -> "10111010001000/"
9: "/" -> "\"
    "10111010001000/" -> "101110100010000\"
17: "1000\" -> "\8"
    "101110100010000\" -> "1011101000\8"
17: "1000\" -> "\8"
    "1011101000\8" -> "101110\88"
11: "1110\" -> "\E"
    "101110\88" -> "10\E88"
23: "10\" -> "\2"
    "10\E88" -> "\2E88"
27: "\2" -> "2"
    "\2E88" -> "2E88"
```

32333 -> 3BF

```
41: "" -> "*"
    "32333" -> "*32333"
4: "*3" -> "/3"
    "*32333" -> "/32333"
8: "/3" -> "11/"
    "/32333" -> "11/2333"
7: "/2" -> "10/"
    "11/2333" -> "1110/333"
8: "/3" -> "11/"
    "1110/333" -> "111011/33"
8: "/3" -> "11/"
    "111011/33" -> "11101111/3"
8: "/3" -> "11/"
    "11101111/3" -> "1110111111/"
9: "/" -> "\"
    "1110111111/" -> "1110111111\"
10: "1111\" -> "\F"
    "1110111111\" -> "111011\F"
14: "1011\" -> "\B"
    "111011\F" -> "11\BF"
22: "11\" -> "\3"
    "11\BF" -> "\3BF"
28: "\3" -> "3"
    "\3BF" -> "3BF"
```

133 -> 1F

```
41: "" -> "*"
    "133" -> "*133"
2: "*1" -> "/1"
    "*133" -> "/133"
6: "/1" -> "1/"
    "/133" -> "1/33"
8: "/3" -> "11/"
    "1/33" -> "111/3"
8: "/3" -> "11/"
    "111/3" -> "11111/"
9: "/" -> "\"
    "11111/" -> "11111\"
10: "1111\" -> "\F"
    "11111\" -> "1\F"
24: "1\" -> "\1"
    "1\F" -> "\1F"
26: "\1" -> "1"
    "\1F" -> "1F"
```

202020 -> 888

```
41: "" -> "*"
    "202020" -> "*202020"
3: "*2" -> "/2"
    "*202020" -> "/202020"
7: "/2" -> "10/"
    "/202020" -> "10/02020"
5: "/0" -> "00/"
    "10/02020" -> "1000/2020"
7: "/2" -> "10/"
    "1000/2020" -> "100010/020"
5: "/0" -> "00/"
    "100010/020" -> "10001000/20"
7: "/2" -> "10/"
    "10001000/20" -> "1000100010/0"
5: "/0" -> "00/"
    "1000100010/0" -> "100010001000/"
9: "/" -> "\"
    "100010001000/" -> "100010001000\"
17: "1000\" -> "\8"
    "100010001000\" -> "10001000\8"
17: "1000\" -> "\8"
    "10001000\8" -> "1000\88"
17: "1000\" -> "\8"
    "1000\88" -> "\888"
33: "\8" -> "8"
    "\888" -> "888"
```

Выводы

В ходе выполнения данного задания я научился переводить числа из 4 в 16 систему счисления, используя нормальные алгоритмы Маркова.