



Златопольский Дмитрий Михайлович
Кандидат технических наук, доцент кафедры
информатики и прикладной математики
Московского городского педагогического
университета.

Он дивен, палиндром...

В статье представлена информация о так называемых палиндромах, а также рассматривается ряд задач по программированию, связанных с ними.

Палиндромом называются слово, предложение или стихотворная строка, одинаково читаемые слева направо и справа налево. Например:

ИДИ

ШАЛАШ

АНИКИНА

АРГЕНТИНА МАНИТ НЕГРА

Слово *палиндром* в переводе с греческого буквально означает «бегущий назад». По-русски палиндром часто называют «перевертень».

В древности палиндромам придавали магический или сакральный смысл. Самым древним из магических палиндромов считают такой: SATOR AREPO TENET OPERA ROTAS (в переводе с латыни – «Сеятель Арепо с трудом держит колёса»). Из него складывается магический квадрат, где выражение читается как вертикально, так и горизонтально, как слева направо, сверху вниз, так и наоборот:

S	A	T	O	R
A	R	E	P	O
T	E	N	E	T
O	P	E	R	A
R	O	T	A	S

Из-за магических свойств этот палиндром считали оберегом от болезней и злых духов.



Волшебный смысл палиндромов, видимо, осознавали и русские скomorохи, в своих представлениях выкрикивавшие: «На в лоб, болван».

Широко известен и палиндром русского поэта Гавриила Романовича Державина: «Я иду съ мечемъ судия». Многие, конечно, читали сказку Алексея Николаевича Толстого «Золотой ключик, или Приключения Буратино» и помнят «диктант», который Мальвина уст-

роила главному герою сказки: «Она подняла к потолку хорошенькие глаза. – Пишите: “А роза упала на лапу Азора”. Написали? Теперь прочтите эту волшебную фразу на-оборот». Используя палиндром сочинил другой русский поэт А. Фет. Велимир Хлебников создал целую палиндромическую поэму «Ра-зин».



20 февраля 2002 года (сама по себе запись даты «палиндромична»!) Питер Норвиг разработал компьютерную программу, которая сгенерировала самый длинный палиндром на английском языке. Он состоит из 17 259 слов. Однако данный палиндром можно назвать таковым лишь условно, поскольку он не является логически связанным текстом. Это всего лишь набор слов, разделённых запятыми.

Рассмотрим ряд задач по программированию, связанных с палиндромами. При описании используем школьный язык программирования. Русский синтаксис этого языка делает приведённые программы максимально понятными и легко переносимыми на любой другой язык программирования.



Задача 1. Дано слово. Проверить, является ли оно палиндромом.

В соответствии с определением палиндрома задачу можно решить, сравнив заданное слово и слово, образованное из его букв, прочитанных справа налево. Соответствующие две величины назовём *слово* и *слово_наоборот*. Прежде чем представлять соответствующую программу, заметим, что в школьном языке программирования величина строкового (в терминах языка Паскаль) типа рассматривается как массив, элементами которого являются отдельные символы. Это позволяет обратиться к тому или иному символу строки как к элементу массива – по имени строковой величины и его (символа) номеру¹. Рассмотреть же все символы строки можно с помощью стандартной функции *длин*, возвращающей общее количество символов в строковой величине.

Итак, программа решения задачи:

```
алг Проверка
нач лит слово, слово_наоборот, цел i
. вывод нс, "Введите слово"
```

¹ Также можно обратиться к отдельному символу строковой величины и в программе на языке Паскаль. В языке Бейсик для этого используется функция *MID\$*.

```
. ввод слово
. слово_наоборот := "" |Начальное значение величины
. нц для i от длин(слово) до 1 шаг -1
. . |Рассматриваем все символы, начиная с последнего,
. . |и добавляем их к "старому" значению слово_наоборот
. . слово_наоборот := слово_наоборот + слово[i]
. кц
. |Сравниваем слова
. если слово = слово_наоборот
. . то
. . . вывод нс, "Введенное слово - палиндром"
. . иначе
. . . вывод нс, "Введенное слово палиндромом не является"
. все
кон
```

Можно также решить задачу, не формируя величину *слово_наоборот*, а сравнивая попарно первую и последнюю, вторую и предпоследнюю и т. д. буквы заданного слова. Если встретится хоть одна пара различающихся букв, то заданное слово палиндромом не является. Сравнения букв можно проводить до середины слова. Прежде чем представлять соответствующую программу, установим, с какой буквой сравнивается буква, номер которой (при счёте слева направо) – лев. Для этого составим такую таблицу (в ней n – общее число букв в слове):

Номер буквы	С какой буквой она сравнивается
1	С последней (n -й)
2	С предпоследней, ($n - 1$)-й
3	С ($n - 2$)-й

...	...
$n \div 2 - 1$	С ($n - n \div 2 + 2$)-й
$n \div 2$	С ($n - n \div 2 + 1$)-й

где $\div$ – знак операции целочисленного деления.

Внимание! Индекс в двух последних строках таблицы нельзя рассчитывать с использованием выражения $n/2$, так как значение номера буквы может быть только целым.

Анализ таблицы показывает, что лев-я буква должна сравниваться с ($n - \text{лев} + 1$)-й. Здесь возникает также вопрос: а что будет, если n – нечётное число? Ответ: в этом случае значения в таблице не изменятся, а средний элемент (его индекс $n \div 2 + 1$) меняться не будет (убедитесь в этом, рассмотрев конкретные значения n).

Вся программа имеет вид:

```
алг Проверка2
нач лит слово, цел лев, лог палин
. вывод нс, "Введите слово"
. ввод слово
. |Сначала принимаем, что слово – палиндром
. палин := да
. нц для лев от 1 до  $\text{div}(\text{длин}(\text{слово}), 2)$ 
. . если слово[i] <> слово[длин(слово) - лев + 1]
. . . то
. . . . палин := нет
```

```
. . все
. кц
. если палин
. . то
. . . вывод нс, "Введенное слово - палиндром"
. . иначе
. . . вывод нс, "Введенное слово палиндромом не является"
. все
кон
```

где *div* – функция, возвращающая целую часть частного от деления своего первого аргумента на второй (в других языках программирования для этого используется не функция,

а специальная операция).

Можно также прекратить проверки, как только встретятся разные буквы (применим оператор цикла с условием):

```
алг Проверка3
нач лит слово, цел лев, лог палин
. вывод нс, "Введите слово"
. ввод слово
. |Сначала принимаем, что слово - палиндром
. палин := да
. лев := 1
. нц пока палин и лев <= div(длин(слово), 2)
. . если слово[i] <> слово[длин(слово) - лев + 1]
. . . то
. . . . палин := нет
. . все
. кц
. если палин
. . то
. . . вывод нс, "Введенное слово - палиндром"
. . иначе
. . . вывод нс, "Введенное слово палиндромом не является"
. все
кон
```

Задача 2. Дано предложение. Проверить, является ли оно палиндромом без учёта имеющихся в нём пробелов. Начальных и конечных пробелов и других символов, кроме букв, в предложении нет.

Задачу можно решить в два основных этапа:

1) получить слово (скорее всего,

оно будет бессмысленным), состоящее только из букв заданного предложения (без пробелов);

2) проверить полученное слово по методике, описанной применительно к задаче 1.

Следовательно, начало программы решения задачи может быть таким:

```
алг Задача_2
нач лит предложение, слово, цел i
. вывод нс, "Введите предложение"
```

```
. ввод предложение
. слово := ""
. нц для i от 1 до длин(предложение)
. . если предложение[i] <> " "
. . |Если очередной символ - не пробел
. . . то
. . . . |добавляем его к "старому" значению слова
. . . . слово := слово + предложение[i]
. . все
. кц
...
```

Определение понятия «палиндром» можно применить и к числам. Например, палиндромами являются числа 7, 121, 42024.

Является ли число палиндромом, человек может заметить сразу. Но было бы интересно узнать, является ли конкретное десятичное число палиндромом в какой-либо другой системе счисления. Например, число 45 является палиндромом в двоичной и в восьмеричной системах (проверьте!).

Задача 3. Дано натуральное десятичное число. Найти все его палиндромы в системах счисления с основанием от 2 до 9.

Обозначим заданное натуральное десятичное число n , а основание систем счисления, в которые будет переводиться это число – *основание*. Идея решения первой части задачи – получить все цифры записи числа n в системе с основанием *основание* и записать каждую из них в массив. Затем исследовать отдельные элементы массива с цифрами числа аналогично тому, как мы исследовали отдельные символы за-

данного слова при решении задачи 1 (лучше всего методом, описанным в программе Проверка3).

Напомним методику перевода целых чисел из десятичной системы счисления в систему с другим основанием. Необходимо определять остаток от деления заданного числа и всех промежуточных целочисленных частных на *основание* и делать это до тех пор, пока частное не станет равно нулю.

В приведённом ниже фрагменте программы, в котором происходит заполнение массива цифр, кроме величин n и *основание*, используются также следующие основные переменные величины:

- *цифры* – массив с данными целого типа, в котором будут храниться цифры «нового» числа. Размер этого массива следует определить с учётом возможной длины новой записи числа;

- *кол_цифр* – фактическое количество цифр в новой записи числа.

Обратим внимание на то, что в массив цифры записываются, начиная с последней цифры новой записи.

```
...
кол_цифр := 0
нц пока n > 0
|Увеличиваем значение кол_цифр
кол_цифр := кол_цифр + 1
|Определяем очередную цифру
```

```
|и записываем ее в массив
цифры[кол_цифр] := mod(n, основание)
|Определяем целочисленное частное
n := div(n, основание)
кц
```

где *mod* – функция, возвращающая остаток от деления своего первого аргумента на второй. В других языках программирования для этого используется не функция, а специальная операция (как правило, знак этой операции также обозначается *mod*).

Прежде чем представлять всю программу решения задачи, заме-

тим, что в ней используется величина *копия_n* – так сказать, «копия значения заданного числа *n*» (извините за «масляное масло» – ☺), поскольку при выделении цифр обрабатываемое число меняется, а исходное значение *n* понадобится при различных системах счисления.

Итак, программа:

```
алг Задача_3
нач цел n, основание, копия_n, кол_цифр, лев, i, цел таб
цифры[1:20], лог палин
. вывод нс, "Введите натуральное число"
. ввод n
. |Рассматриваем различные основания
. нц для основание от 2 до 9
. . копия_n := n
. . |Заполняем массив цифры
. . кол_цифр := 0
. . нц пока копия_n > 0
. . . кол_цифр := кол_цифр + 1
. . . цифры[кол_цифр] := mod(копия_n, основание)
. . . копия_n := div(копия_n, основание)
. . кц
. . |Проверяем на "палиндромичность"
. . палин := да
. . лев := 1
. . нц пока палин и лев <= div(длин(слово), 2)
. . . если слово[i] <> слово[длин(слово) - лев + 1]
. . . . то
. . . . . палин := нет
. . . . все
. . . кц
. . если палин
. . . то
. . . . |Выводим число в новой системе счисления
. . . . |цифры в обратном порядке)
. . . . нц для i от кол_цифр до 1 шаг -1
. . . . . вывод цифры[i]
. . . . кц
. . . . вывод " ", основание
. . все
```

. КЦ
кон

Примечание. В программе после числа-палиндрома выводится также

основание системы счисления, при котором такое число получается.

Задания для самостоятельной работы

1. Разработав любую из приведённых программ решения задачи 1 (на языке программирования, который вы изучаете), установите, является ли палиндромом слово «Аникина». Объясните полученный результат.

2. Разработайте программу решения задачи «Дано предложение. Проверить, является ли оно палиндромом без учёта имеющихся в нём пробелов, цифр, русских букв и других символов, кроме латинских букв».

3. Определите максимальное количество систем счисления (с основанием от 2 до 9), в которых числа от 2 до 10000 являются палиндромами.

В заключение заметим, что самым длинным словом-палиндромом в мире считается финское *saipriakauppias* – «продавец мыла», а также приведём изящный пример, принадлежащий Кириллу Решетникову: «Он дивен, палиндром, и ни морд, ни лап не видно»...

Литература

1. Беляева Н. Геометрия в поэзии//Математика. – 2011. – №11.

Юмор Юмор Юмор Юмор Юмор Юмор

Из жизни программистов:

– Предположим, у тебя есть 1000 рублей... Ну, для круглого счёта возьмём 1024!

Вебмастер сдаёт на права, заполняет бланк:

Рост: 175

Вес: 80

Цвет глаз: #44AAFF

Сын приходит к папе, постоянно сидящему в Интернете:

– Папа, а «адрес» пишется с одной «с» или с двумя?

– С тремя «w», сынок.

Разговор двух заядлых посетителей Интернета:

– Слышь... это... www.zarplaty.net?

– www.kak.vsegda.net!

– www.kak.zhe.ya.domoy.po.edu?

– www.pesh.com.