Решение задач, основанных на алгоритме, раскладывающем число на цифры

Мельникова Д.Ю.
dmelnikova8@gmail.com,
МАОУ «Физико-технический лицей № 1» г. Саратова, г. Саратов, Россия

В статье рассматриваются особенности арифметического и итерационного циклов, описывается алгоритм, раскладывающий натуральное число на цифры. Приводятся описания разнообразных задач, работающих с цифрами числа. Автор рассматривает ключевые моменты алгоритмов, важные для понимания учащимися.

Ключевые слова: урок, информатика, цифры числа, итерационный цикл.

Введение

Обучение программированию в школе на каком бы то ни было языке начинается с изучения основных алгоритмических конструкций. При знакомстве с циклическими алгоритмами важно, чтобы ученики хорошо усвоили особенности арифметического и итерационного циклов. Информатика является в большой степени практической дисциплиной, опирающейся на навыки решения задач. Для понимания свойств итерационного цикла можно рассмотреть большое количество интересных заданий разной степени сложности, основанных на алгоритме, раскладывающем натуральное число на цифры.

Изучение циклических конструкций

При изучении циклических конструкций часто используется прием решения одной и той же задачи несколькими способами, с использованием разных циклов, например, for и while (Паскаль). Были замечены следующие особенности усвоения материала учащимися. Если сначала изучался арифметический цикл, а потом итерационный, то больше ошибок ученики допускают при использовании итерационного цикла:забывают инициализировать переменную цикла перед циклом,изменять переменную-счётчик внутри цикла. При изменении порядка изучения циклов итерационный цикл while усваивается лучше, но появляются неожиданные ошибки при написании арифметических циклов. Наблюдаются попытки самостоятельного изменения счетчика, особенно в случае шага, отличного от единицы, варианты написания кода без использования «лишней» третьей переменной – непонятного счётчика, ведь в итерационном цикле можно было обойтись двумя переменными, отвечающими за границы цикла.

Возможно, хорошей идеей при изучении циклических конструкций является использование одного, оптимального цикла при решении задач. В задачах, связанных с перебором последовательного числового рядаочевидно удобство арифметического цикла. Хорошими учебными задачами для понимания итерационного цикла являются такие алгоритмы, как поиск наибольшего общего делителя (алгоритм Евклида) и алгоритм, раскладывающий натуральное число на цифры.

Алгоритм, раскладывающий число на цифры

Задача заключается в том, чтобы для натурального числа с произвольным

количеством разрядов рассмотреть все его цифры, например, для вывода на экран или подсчета их количества, суммы или некоторой другой характеристики. Решение основывается на повторяющейся последовательности действий: рассматриваем последнюю (крайнюю справа) цифру числа, выводим на экран или подсчитываем числовую характеристику, отбрасываем от числа последнюю цифру. Количество разрядов числа произвольно, значит, неизвестно, сколько раз будут повторяться действия. В таком случае невозможно решить задачу при помощи арифметического цикла, нужен итерационный, по-другому называемый циклом с условием. Надо остановиться, когда цифр (разрядов) в числе не останется. Выделение цифры выполняется с помощью операции взятия остатка от деления числа на 10, а отделение цифры от числа – с помощью операции целочисленного деления на 10. Результат применения этой операции к однозначному числу (отбрасывание последней оставшейся цифры) – ноль. Значит, надо продолжать выполнять повторяющиеся действия, пока число будет не равно нулю.

Листинг 1. Пример программного кода алгоритма на языке Паскаль

```
k := 0;
while n<> 0 do
begin
     x := n mod 10;  // рассматриваем крайнюю цифру числа
     writeln(x);           // выводим цифру на экран
     k := k + 1;           // подсчитываем количество цифр
     n := ndiv 10;   // отделяем от числа рассмотренную цифру
end;
writeln(k);                // выводим на экран количество цифр
```

При рассмотрении этого базового алгоритма можно обсудить следующие важные для понимания вопросы. Какие команды, выполняющиеся в теле цикла, можно убрать, не нарушая работы алгоритма? (Оставить только последнюю). Почему вывод переменной x происходит в цикле, а переменной k после цикла? (В первом случае выводится каждая цифрана каждом шаге цикла, а во втором – один подсчитанный ответ). Что станет с переменной, отвечающей за исходное число, после цикла, можно ли дальше в тексте программы будет использовать это числовое значение? (Нет, переменная обнулится). Как решить эту проблему? (Завести ещё одну переменную и перед циклом сохранить в нее исходное значение).

Задачи, основанные на алгоритме, раскладывающем число на цифры

Рассмотрим несколько задач, основанных на базовом алгоритме.

1. Вывести все нечетные цифры числа.

Особенностью этой задачи является добавление проверки условия перед выводом цифры числа на экран.

2. Определить, верно ли, что все цифры числа – нечётные.

При самостоятельном решении подобных задач ученики часто допускают ошибки, выводя сразу ответ на экран при проверке условия внутри цикла. Для понимания идеи решения можно обсудить следующие вопросы. Можно ли при любом заданном числе гарантированно дать ответ, не рассмотрев все цифры

числа? (Нет, вдруг единственная четная цифра – в старшем разряде?) На эту задачу даётся один ответ (вывод на экран), или несколько раз может что-то выводиться? (Один ответ, соответственно, вывод на экран должен быть после выполнения цикла). Как дать ответ после цикла, рассмотрев все цифры? (Завести переменную-«счётчик» или переменную-«флажок», которая будет принимать некоторое значение внутри цикла и анализироваться для вывода ответа после выполнения цикла). Как определить, что все цифры нечётные? (Чётных цифр нет).
Вводятся или вспоминаются понятия «счётчика» и «флажка», обсуждаются моменты решения.
Листинг 2. Пример программного кода на языке Паскаль

```
k:=0;            // вариант с использованием «флажка»
while n <> 0 do
begin
     x := n mod 10;
     if x mod 2 = 0 thenk := 1;
     n := n div 10;
end;
if k = 0
then writeln ('da')
else writeln ('net');
```

Как оптимизировать код? (Остановить цикл, если встретили чётную цифру). Как осуществить? (Ввести ещё одно условие в заголовок цикла). Нужен ли тогда «флажок»? Что проверять, чтобы вывести ответ после выполнения цикла? (Проверять значение исходного числа).
Листинг 3. Пример оптимизированного кода

```
while(n <> 0) and (n mod 10 mod 2 <> 0)do
n := n div 10;
if n = 0
then writeln ('da')
else writeln ('net');
```

3. Дано натуральное число. Выяснить, равны ли его крайние цифры.
Как найти последнюю цифру числа? (По формуле, остаток от деления числа на 10). Как найти первую цифру числа с произвольным количеством разрядов? (Применяя алгоритм, раскладывающий число на цифры). Сможем ли мы обратиться к последней цифре после выполнения алгоритма? (Нет, переменная, хранящая исходное число, станет равна нулю. Надо запомнить в переменную до цикла). Как сохранить первую цифру числа? (Два способа: либо завести переменную, записывающую каждую цифру числа, либо остановить цикл, когда от исходного числа останется одна цифра).

Листинг 4. Пример программного кода на языке Паскаль

```
a:=n mod 10;
while n div 10 <> 0 do
```

```
n := n div 10;
if a = n   then writeln (’da’)
      else writeln(‘net’);
```

4. Определить, верно ли, что в записи натурального числа есть две одинаковые цифры, стоящие рядом.

Цифры надо рассматривать попарно – крайняя и стоящая перед ней. Сколько будет таких пар? (На одну меньше, чем цифр). Как это указать в заголовке цикла? (Надо раньше остановиться – когда останется один разряд). Когда даётся ответ на задачу? (После цикла. Переменная «флажок» принимает значение 1, если найдена пара рядом стоящих одинаковых цифр. После цикла анализируется значение «флажка», чтобы дать ответ).

Листинг 5. Пример программного кода на языке Паскаль

```
k:=0;
while (n >9) and (k = 0) do
begin
      x := n mod 10;
      n := n div 10;
      y:= n mod 10;
      if x = y then k:=1;
end;
if k=1     then writeln (’da’)
      else writeln(‘net’);
```

Заключение

Алгоритм, раскладывающий число на цифры, несложен для учащихся средней школы и удобен для отработки навыков применения итерационного цикла. Основываясь на этом алгоритме, можно рассмотреть разнообразные задачи и познакомиться с интересными приёмами программирования.

Список литературы

[1] *Огнева М.В., Кудрина Е.В.* TURBO PASCAL: первые шаги. Примеры и упражнения. Учебное пособие. – Саратов: Изд-во «Научная книга», 2014.С. 84 – 85