

## МЕТОД ВИГЛЕЗИО ИЗВЛЕЧЕНИЯ КВАДРАТНОГО КОРНЯ\*

СТЕФАН ПАШКОВСКИ

**1. Введение.** Из методов извлечения квадратного корня наиболее известны три метода:

а) метод, основанный на формуле

$$\sum_{k=1}^p (2k-1) = p^2,$$

применимый в таких цифровых вычислительных машинах, в которых извлечение корней является элементарной операцией, как сложение или умножение,

б) тесно связанный с предыдущим, „школьный“ метод вычисления последовательных цифр корня, обычно применяемый при расчётах вручную,

в) итерационный метод Ньютона, применяемый, чаще всего, в вычислениях на цифровых вычислительных машинах.

Иногда, напр. в некоторых задачах теории чисел, требуется получить корень с очень большой точностью — до нескольких сот значащих цифр. В таком случае вышеуказанные методы непригодны, так как приходилось бы производить арифметические действия над чрезмерно длинными числами, что, разумеется, неудобно не только при расчётах вручную, но и для большинства цифровых вычислительных машин.

В 1923 г. Элиза Виглезио предложила в работе [2] очень простой метод извлечения квадратного корня из натурального числа.

Методом Виглезио (в дальнейшем мы будем короче его называть методом  $\sqrt{\cdot}$ ) вычисляются очередные значения цифры разложения корня при произвольном установленном базиссе. В противоположность методам а) и б), вспомогательные величины, получаемые в ходе вычислений, возрастают очень медленно, когда возрастает

\* Работа была представлена 30 января 1963 г. на семинаре кафедры вычислительной математики Вроцлавского университета.

число цифр — и напр. 10 000 десятичных цифр корня можно вычислить, оперируя, самое большое, шестизначными числами.

Ниже в предлагаемой работе мы припомним метод  $\mathcal{V}$  (§ 2), изложим вытекающий из него, но короче осуществляющийся, метод  $\mathcal{W}$  (§§ 3-5), приведём написанный на алгоритмическом языке АЛГОЛ алгоритм этого метода (§ 6), программу, составленную для цифровой вычислительной машины Эллиотт 803, и примеры корней, вычисленных с помощью этой программы (§ 7).

**2. Метод  $\mathcal{V}$ .** Пусть  $n$  будет неквадратное натуральное число ( $n = 2, 3, 5, 6, \dots$ ). Обозначим целую часть  $\sqrt{n}$  буквой  $p$ , а очередные десятичные цифры дробной части  $\sqrt{n}$  — символами  $c_1, c_2, \dots$ . Итак

$$\sqrt{n} = p + 0,1c_1 + 0,1^2c_2 + 0,1^3c_3 + \dots$$

Отсюда следует, что

$$(1) \quad n = p^2 + 0,1Pc_1 + 0,1^2c_1^2 + 0,1^2Pc_2 + 0,1^3 \cdot 2c_1c_2 + \\ + 0,1^3Pc_3 + 0,1^4(2c_1c_3 + c_2^2) + \dots = p^2 + \sum_{j=1}^{\infty} 0,1^{j+1}(10Pc_j + s_j)$$

где

$$P = 2p, \quad s_j = \sum_{i=1}^j c_i c_{j+1-i}$$

В методе  $\mathcal{V}$  из числа  $n$  попеременно вычтываются слагаемые правой части формулы (1), т. е. числа  $p^2, 0,1Pc_1, 0,1^2s_1, 0,1^2Pc_2, 0,1^3s_2, \dots$  и определяются (далее поясним, каким образом) цифры  $c_1, c_2, c_3, \dots$  Так как удобнее всего пользоваться целыми числами, то получаемые разности умножают на соответственные степени 10. Эти разности обозначим следующими символами:

$$r_1 = 10(n - p^2), \quad q_k = r_k - P c_k, \quad r_{k+1} = 10q_k - s_k \quad (k = 1, 2, \dots).$$

Легко проверить, что

$$(2) \quad r_{k+1} = 10^{k+1} \left( n - p^2 - \sum_{j=1}^k 0,1^{j+1}(10Pc_j + s_j) \right) = \\ = 10^{k+1} \sum_{j=k+1}^{\infty} 0,1^{j+1}(10Pc_j + s_j), \\ q_k = 0,1(r_{k+1} + s_k),$$

следовательно, разности  $q_k$  должны быть положительными. Метод  $\mathcal{V}$  и состоит в том, что пробное значение цифры  $c_k$  ( $k = 1, 2, \dots$ ) опре-

деляют как наибольшее целое число, меньшее 10 и такое, что  $q_k > 0$ , т. е.

$$(3) \quad c_k < \frac{r_k}{P}.$$

Пример 1. Если  $n = 200$ , то  $p = 14$ ,  $P = 28$ ,

$$r_1 = 10(200 - 14^2) = 40,$$

$$r_1/P = \frac{40}{28}, \quad c_1 = 1, \quad s_1 = 1 \cdot 1 = 1,$$

$$q_1 = 40 - 28 \cdot 1 = 12, \quad r_2 = 10 \cdot 12 - 1 = 119,$$

$$r_2/P = \frac{119}{28}, \quad c_2 = 4, \quad s_2 = 1 \cdot 4 + 4 \cdot 1 = 8,$$

$$q_2 = 119 - 28 \cdot 4 = 7, \quad r_3 = 10 \cdot 7 - 8 = 62,$$

$$r_3/P = \frac{62}{28}, \quad c_3 = 2, \quad s_3 = 1 \cdot 2 + 4 \cdot 4 + 2 \cdot 1 = 20,$$

$$q_3 = 62 - 28 \cdot 2 = 6, \quad r_4 = 10 \cdot 6 - 20 = 40, \dots$$

и, действительно, имеем  $\sqrt{200} = 14,142\dots$

Определённая из неравенства (3) цифра  $c_k$  может быть неверной, ибо из этого неравенства ещё не следует неравенство  $r_{k+1} > 0$ , которое, как знаем, должно выполняться.

Пример 2. Если  $n = 6$ , то  $p = 2$ ,  $P = 4$ ,

$$r_1 = 10(6 - 2^2) = 20,$$

$$r_1/P = \frac{20}{4}, \quad c_1 = 4, \quad s_1 = 4 \cdot 4 = 16,$$

$$q_1 = 20 - 4 \cdot 4 = 4, \quad r_2 = 10 \cdot 4 - 16 = 24,$$

$$r_2/P = \frac{24}{4}, \quad c_2 = 5, \quad s_2 = 4 \cdot 5 + 5 \cdot 4 = 40,$$

$$q_2 = 24 - 4 \cdot 5 = 4, \quad r_3 = 10 \cdot 4 - 40 = 0.$$

Действительно, легко проверить, что  $\sqrt{6} < 2,45$ .

Из примера 2 следует необходимое дополнение метода  $\mathcal{V}$ : если  $r_{k+1} \leq 0$  для некоторого  $k$ , то цифру  $c_k$  следует уменьшить на 1 и вторично вычислить  $s_k$  и  $r_{k+1}$ . Иногда цифру приходится корректировать несколько раз.

Пример 3. Если  $n = 3$ , то  $p = 1$ ,  $P = 2$ ,  $r_1 = 20$ ,

$$c_1 = 9, \quad s_1 = 81, \quad q_1 = 2, \quad r_2 = -61 \quad (c_1 \text{ неверное}),$$

$$c_1 = 8, \quad s_1 = 64, \quad q_1 = 4, \quad r_2 = -24 \quad (c_1 \text{ неверное}),$$

$$c_1 = 7, \quad s_1 = 49, \quad q_1 = 6, \quad r_2 = 11,$$

$$c_2 = 5, \quad s_2 = 70, \quad q_2 = 1, \quad r_3 = -60 \quad (c_2 \text{ неверное}).$$

Следующий пример показывает, что неверной цифрой может быть 0. Тогда надо вернуться к последней цифре, отличной от нуля, уменьшить её на 1 и вторично считать следующие за ней цифры.

Пример 4. Если  $n = 2499$ , то  $p = 49$ ,  $P = 98$ ,  $r_1 = 980$ ,

$$c_1 = 9, s_1 = 81, q_1 = 98, r_2 = 899,$$

$$c_2 = 9, s_2 = 162, q_2 = 17, r_3 = 8,$$

$$c_3 = 0, s_3 = 81, q_3 = 8, r_4 = -1 \text{ (} c_2 \text{ и } c_3 \text{ неверные),}$$

$$c_2 = 8, s_2 = 144, q_2 = 115, r_3 = 1006,$$

$$c_3 = 9, s_3 = 226, q_3 = 124, r_4 = 1014,$$

$$c_4 = 9, s_4 = 306, q_4 = 132, r_4 = 1014, \dots$$

Другие примеры позволяют полагать, что для произвольного натурального числа  $l$  существует такое число  $n$ , что, применяя метод  $\mathcal{V}$ , нужно корректировать  $l$  очередных цифр  $\sqrt[n]{n}$ , определённых из неравенства (3). Как же проверить, которые из вычисленных цифр квадратично корня верны, а которые, быть может, ещё будут изменены? В § 4 мы ответим на этот вопрос, но сказанное нами будет относиться к методу Виглезио, модифицированному таким образом, чтобы реже приходилось корректировать цифры.

**3. Метод  $\mathcal{W}$ .** Изложение метода  $\mathcal{W}$ , представляющего собой некоторую модификацию метода  $\mathcal{V}$ , начнём с вычисления цифры  $c_1$ .

Известно, что

$$r_2 = 10(r_1 - P c_1) - s_1 = 10(10(n - p^2) - 2pc_1) - c_1^2 = 100(n - (p + 0,1c_1)^2).$$

Так как должно быть

$$\sqrt[n]{n} > p + 0,1c_1,$$

то  $r_2 > 0$ , если цифра  $c_1$  определена правильно, а если  $c_1$  слишком велика, то  $r_2 \leq 0$ . Поэтому  $c_1$  можно определить как наибольшее целое число  $c$ , удовлетворяющее квадратному неравенству

$$10(r_1 - P c) - c^2 > 0.$$

Во избежание действий над иррациональными числами напишем это неравенство в виде

$$10r_1 - 10P(1 + 1 + \dots + 1) - (1 + 3 + \dots + (2c - 1)) > 0.$$

Цифру  $c_1$  получим отсюда путём итерации, проверяя, сколько слагаемых типа  $10P + 1, 10P + 3, \dots$  можно вычесть из числа  $10r_1$  без перемены его знака.

Перейдём теперь к методу определения следующих цифр  $\sqrt[n]{n}$ . В методе  $\mathcal{W}$  пробное значение цифры  $c_k$  ( $k = 2, 3, \dots$ ) определим как

наибольшее целое число, меньшее 10, и такое, что  $r_{k+1} > 0$ . Так как это неравенство сильнее неравенства  $q_k > 0$ , применяющегося в методе  $\mathcal{W}$ , можно полагать, что теперь реже придётся корректировать цифры. Примеры подтверждают это.

Известно, что

$$(4) \quad r_{k+1} = 10(r_k - P c_k) - s_k.$$

Для  $k > 1$  имеем

$$s_k = \sum_{i=1}^k c_i c_{k+1-i} = 2c_1 c_k + t_k,$$

где

$$t_k = \sum_{i=2}^{k-1} c_i c_{k+1-i}$$

(в частности  $t_2 = 0$ ). Итак, имеем равенство

$$(5) \quad r_{k+1} = 10r_k - t_k - (10P + 2c_1)c_k,$$

в котором сумма  $t_k$  уже не зависит от искомой цифры  $c_k$ .

Пусть

$$Q = 10P + 2c_1, \quad R_k = 10r_k - t_k.$$

Таким образом, цифру  $c_k$  при методе  $\mathcal{W}$  определяем как наибольшее целое число, меньшее 10, и такое, что

$$(6) \quad c_k < \frac{R_k}{Q}.$$

Пример 5. Если  $n = 6$ , то (ср. пример 2)  $p = 2$ ,  $P = 4$ ,  $c_1 = 4$ ,  $r_2 = 24$ , значит  $Q = 48$ ,  $R_2 = 240$ , откуда  $c_2 = 4$ , а не — как в примере 2 —  $c_2 = 5$ .

Пример 6. Если  $n = 2499$ , то (ср. пример 4)  $p = 49$ ,  $P = 98$ ,  $c_1 = 9$ ,  $r_2 = 899$ , значит  $Q = 998$ ,  $R_2 = 8990$ , откуда  $c_2 = 9$ . Мы уже знаем, что эта цифра неверна. Отсюда следует, что и определение (6) цифры  $c_k$  может быть неокончательным и что вопрос, поставленный в конце § 2, является важным и для метода  $\mathcal{W}$ .

**4. Исследование корректности цифр в методе  $\mathcal{W}$ .** Пусть  $k > 1$  и  $l > 1$  — натуральные числа. Предположим, что цифры  $c_1, c_2, \dots, c_{k+l-1}$  квадратного корня из числа  $n$  определены методом  $\mathcal{W}$ . Стало быть, выполняется между прочим, неравенство  $r_{k+l} > 0$ . Пример 6 показывает, что эти цифры могут быть неверными. Очевидна, однако, следующая теорема:

Цифры  $c_1, c_2, \dots, c_k$  верны, если к ним можно присоединить цифры  $c'_{k+1}, c'_{k+2}, \dots, c'_{k+l-1}$  (тождественные с цифрами, уже вычисленными методом  $\mathcal{W}$  или отличные от них) и  $c'_{k+l}, c'_{k+l+1}, \dots$  таким образом, что соответствующие им числа  $r'_{k+2}, r'_{k+3}, \dots$  положительные.

Для того, чтобы применить эту теорему, воспользуемся равенством

$$r_{k+m+1} = 10^{m+1} \left( r_k - P \sum_{j=0}^m 0,1^j c_{k+j} - 0,1 \sum_{j=0}^m 0,1^j s_{k+j} \right)$$

( $m$  — целое неотрицательное число), полученным путём итерации формулы (4). В связи с этим, предположение  $r_{k+m+1} > 0$  равносильно неравенствам

$$(7) \quad r_k > P \sum_{j=0}^m 0,1^j c''_{k+j} + 0,1 \sum_{j=0}^m 0,1^j s''_{k+j} \quad (m = 1, 2, \dots),$$

где

$$c''_i = \begin{cases} c_i & (i \leq k), \\ c'_i & (i > k), \end{cases}$$

$$s''_{k+j} = \sum_{i=1}^{k+j} c''_i c''_{k+j+1-i}.$$

С помощью несложных рассуждений получим неравенство, из которого вытекают все неравенства (7) и которое обеспечивает лёгкую проверку, верны ли цифры  $c_1, c_2, \dots, c_k$ .

Заметим сперва, что неравенства (7) следуют из неравенства

$$(8) \quad r_k > P \sum_{j=0}^{\infty} 0,1^j c''_{k+j} + 0,1 \sum_{j=0}^{\infty} 0,1^j s''_{k+j}.$$

Мы будем преобразовывать это неравенство в предположении, что

$$(9) \quad c'_i = 0 \quad (i > k).$$

Ибо, если в этом случае неравенство (8) справедливо, то оно справедливо тоже для некоторой последовательности цифр  $c_{k+1}, c_{k+2}, \dots$ , не всех равных 0, которые могут быть уже верными цифрами  $\sqrt{n}$ .

В случае (9) имеем

$$\sum_{j=0}^{\infty} 0,1^j c''_{k+j} = c_k,$$

$$s''_{k+j} = \begin{cases} \sum_{i=j+1}^k c_i c_{k+j+1-i} & (j < k), \\ 0 & (j \geq k) \end{cases}$$

и неравенство (8) принимает вид

$$(10) \quad r_k > P c_k + 0,1 \sum_{j=0}^{k-1} 0,1^j \sum_{i=j+1}^k c_i c_{k+j+1-i}.$$

Предположение  $r_{k+l} > 0$ , сформулированное в начале настоящего параграфа, можно написать в виде, аналогичном (7):

$$r_k > P \sum_{j=0}^{l-1} 0,1^j c_{k+j} + 0,1 \sum_{j=0}^{l-1} 0,1^j s_{k+j}.$$

В связи с этим, неравенство (10) следует из неравенства

$$P \sum_{j=0}^{l-1} 0,1^j c_{k+j} + 0,1 \sum_{j=0}^{l-1} 0,1^j \sum_{i=1}^{k+j} c_i c_{k+j+1-i} \geq P c_k + 0,1 \sum_{j=0}^{k-1} 0,1^j \sum_{i=j+1}^k c_i c_{k+j+1-i}$$

т. е., при добавочном предположении

$$(11) \quad l \leq k,$$

оно следует из неравенства

$$(12) \quad P \sum_{j=1}^{l-1} 0,1^j c_{k+j} + 0,1 \sum_{j=0}^{l-1} 0,1^j \sum_{1 \leq i \leq j, k+1 \leq i \leq k+j} c_i c_{k+j+1-i} \geq \\ \geq 0,1 \sum_{j=l}^{k-1} 0,1^j \sum_{i=j+1}^k c_i c_{k+j+1-i}.$$

Оценивая сверху слагаемые правой части этого неравенства:

$$\sum_{i=j+1}^k c_i c_{k+j+1-i} \leq 81(k-j) \leq 81(k-l),$$

снизу — первый член левой части:

$$P \sum_{j=1}^{l-1} 0,1^j c_{k+j} \geq 0,1 P c_{k+1}$$

(предполагается, что  $l > 1$ ) и опуская второй член, находим, что неравенство (12) следует из неравенства

$$0,1 P c_{k+1} \geq 8,1(k-l) \sum_{j=l}^{k-1} 0,1^j,$$

а тем более из неравенства

$$0,1 P c_{k+1} \geq 8,1(k-l) \sum_{j=l}^{\infty} 0,1^j = 9(k-l)0,1^l.$$

Отсюда и из предположения (11) вывод таков: если  $r_{k+l} > 0$ , то верны такие цифры  $c_1, c_2, \dots, c_k$ , для которых

$$l \leq k \leq l + \frac{1}{9} P c_{k+1} 10^{l-1}.$$

Это неравенство имеет наибольшее практическое значение тогда, когда  $c_{k+1} > 0$ , а  $l = 2$  (корректность цифры  $c_1$  известна):

$$(13) \quad 2 \leq k \leq 2 + \frac{10}{9} P c_{k+1}.$$

Как видим, интервал изменения индекса  $k$  тем больше, чем больше  $P$ , а следовательно, чем больше число  $n$ . Поэтому, прежде чем извлечь квадратный корень, следует число  $n$  умножить на возможную высокую степень числа 100. В частности, в программе, составленной для цифровой вычислительной машины Эллиотт 803 (§ 7) создается число  $n' = 100^d n$  ( $d$  — целое), такое, что  $2^{38}/100 < n' < 2^{38}$ . Затем методом  $\mathcal{W}$  вычисляется  $\sqrt{n'}$  и применяется формула  $\sqrt{n} = 10^{-d} \sqrt{n'}$ .

Тогда  $P = 2[\sqrt{n'}] \geq 2[2^{19}/10] = 104856, \frac{10}{9} P > 116506$  и полученный выше результат можно сформулировать так:

Если  $r_{k+2} > 0, c_{k+1} \neq 0$  и  $2 \leq k \leq 116508$ , то цифры  $c_1, c_2, \dots, c_k$  числа  $\sqrt{n}$  верны.

**5. Оценка вспомогательных величин.** В введении было сказано, что вспомогательные величины, получаемые в ходе вычисления  $\sqrt{n}$  методом  $\mathcal{W}$ , возрастают очень медленно вместе с возрастанием числа цифр. Покажем сейчас, что при предпосылке (13) то же самое относится и к методу  $\mathcal{W}$ .

Прежде всего ясно, что

$$t_k \leq 81(k-2).$$

Оценивая  $r_{k+1}$  рассмотрим несколько случаев:

I. Цифры  $c_1, c_2, \dots, c_k$ , определённые методом  $\mathcal{W}$  верны. Число  $r_{k+1}$  выражается тогда формулой (2) через эти и следующие, ещё неизвестные цифры, откуда

$$\begin{aligned} r_{k+1} &< 10^{k+1} \sum_{j=k+1}^{\infty} 0,1^{j+1} (90P + 81j) = \\ &= \sum_{j=1}^{\infty} 0,1^j (90P + 81(j+k)) = 10P + 9k + 81 \sum_{j=1}^{\infty} 0,1^j \end{aligned}$$

и окончательно

$$(14) \quad r_{k+1} \leq 10P + 9(k+1).$$

II. Не все цифры  $c_1, c_2, \dots, c_k$  верны.

II. I.  $r_{k+1} > 0, c_k \neq 0$ . Согласно результату § 4, цифры  $c_1, c_2, \dots, c_{k-1}$  верны, а цифра  $c_k$  слишком большая. В связи с этим, вычисленное  $r_{k+1}$  меньше верного значения, для которого была дана оценка (14).

II.2.  $r_{k+1} > 0$ ,  $c_k = 0$ . Из способа определения цифр по методу  $\mathcal{W}$  следует, что

$$r_{k+1} = 10r_k - t_k \leqslant 10P + 2c_1 \leqslant 10P + 18$$

и неравенство (14) справедливо.

II.3.  $r_{k+1} \leqslant 0$ . Так же в этом случае имеем

$$r_{k+1} = 10r_k - t_k,$$

а так как  $r_k > 0$ , то

$$|r_{k+1}| \leqslant t_k - 10 \leqslant 81k - 172.$$

Мы доказали таким образом, что всегда

$$|r_{k+1}| \leqslant \max\{10P + 9(k+1), 81k - 172\}.$$

В случае вычислений на машине Эллиотт 803 рассмотренном в конце § 4, имеем

$$P \leqslant 2(2^{19} - 1) = 1048574,$$

$$10P + 9(k+1) \leqslant 10485740 + 9(116508 + 1) = 11534321,$$

$$81k - 172 \leqslant 9436976, t_k \text{ и } |r_{k+1}| \leqslant 11534321.$$

При вычислении цифр  $c_k$ , стало быть, даже не использована вся длина машинного слова (39 двоичных разрядов).

6. Продедура **Виглезио  $\mathcal{W}$** . Все подробности вычисления  $\sqrt[n]{n}$  методом  $\mathcal{W}$  разъясняет ниже описание процедуры Виглезио  $\mathcal{W}$ , изложенное на алгоритмическом языке АЛГОЛ:

```
procedure Viglesio  $\mathcal{W}$  ( $n, p, g, c, K$ );
comment Заданы натуральное число  $n$  и целая часть  $p$  квадратного корня из  $n$ . Процедура Viglesio  $\mathcal{W}$  вычисляет цифры  $c[1], c[2], \dots, c[K]$  ( $K$  — дано) дробной части этого корня в позиционной системе с основанием  $g$ . Параметры  $p, g$  и  $K$  должны удовлетворять неравенствам  $2 \times g \times p \geqslant (g-1) \times (K-2)$  и (в вычислениях на цифровой вычислительной машине)  $g \times (2 \times g \times p + (g-1) \times (K-1)) \leqslant I_{max}$ ,  $g \times g \times g \times (K-2) \leqslant I_{max}$ , где  $I_{max}$  наибольшее допустимое целое число. Наименее значащие цифры  $c[k]$ , от последней, отличной от нуля, до  $c[K]$ , могут быть неверными. Значение функции  $intdiv(x, y)$ , выступающей в содержании процедуры, является целой частью частного  $x \geqslant 0$  на  $y > 0$ ;
```

```
value  $n, p, g, K$ ;
integer  $n, p, g, K$ ;
integer array  $c$ ;
begin
    integer  $C, k, Q, r, R, t$ ;
    procedure sum;
    begin
        integer  $i, j$ ;
```

```

    t := 0;
    i := 2;
1 : j := k + 2 - i;
    if i < j
        then begin
            t := t + 2 × c[i] × c[j];
            i := i + 1;
            go to 1
        end i < j
    else if i = j
        then t := t + c[i] × c[i]

    end sum;
r := g × g × (n - p × p);
if r ≠ 0 then go to 2;
for k := 1 step 1 until K do c[k] := 0;
go to 10;
2 : C := 0;
    Q := 2 × g × p + 1;
3 : r := r - Q;
    if r ≤ 0 then go to 4;
    C := C + 1;
    Q := Q + 2;
    go to 3;
4 : r := r + Q;
    Q := Q - 1;
    for k := 1 step 1 until K do
begin
    c[k] := C;
    sum;
5 : R := g × r - t;
    if R > 0 then go to 8;
6 : if c[k] ≠ 0 then go to 7;
    k := k - 1;
    sum;
    r := intdiv(r + t, g);
    go to 6;
7 : if k = 2 then t := t + 1;
    t := t - 2 × c[2];
    c[k] := c[k] - 1;
    r := r + Q;
    go to 5;
8 : C := 0;
9 : R := R - Q;

```

```

if  $R > 0$ 
  then begin
     $C := C + 1;$ 
    if  $C < g$ 
      then go to 9
      else  $r := R$ 
    end  $R > 0$ 
  else  $r := R + Q$ 
end  $k$ ;
10 : end Viglesio  $\mathbb{W}$ 

```

К комментарию, помещённому в описании, прибавим ещё несколько примечаний:

(а) Процедурой не охватывается вычисление  $p$ . Это можно сделать с помощью стандартной функции  $\sqrt{q}$  (исправляя, если понадобится, её приближённое значение) или с помощью какой-то процедуры, содержание которой представлено в коде цифровой вычислительной машины.

(б) §§ 2-5 относились только к десятичному разложению дробной части  $\sqrt{n}$ , но переход от этих рассуждений к произвольному основанию  $g$  позиционной системы (как в описании процедуры) очень лёгок.

(в) Процедура sum вычисляет величину

$$t_{k+1} = \sum_{i=2}^k c_i c_{k+2-i}$$

Если цифра  $c_k > 0$  неверна, то её следует уменьшить на 1, т. е.  $t_{k+1}$  уменьшить на число  $2c_2$  (если  $k > 2$ ) или  $2c_2 - 1$  (если  $k = 2$ ). Этому служат три соседние операторы, первый из которых имеет метку 7.

(г) Если цифра  $c_k = 0$  неверна, то надо изменить цифру  $c_{k-1}$ , а для этого нужен, прежде всего, возврат от величины  $r_{k+1}$ , вычисленной последней, к  $r_k$  по формуле

$$r_k = (r_{k+1} + t_k)/10$$

вытекающей из (5). Соответственный оператор — третий после оператора с меткой 6.

**7. Программа для цифровой вычислительной машины Эллиотт 803.** Первый вариант программы для машины Эллиотт 803, основанной на методе  $\mathbb{W}$ , был составлен и проверен в январе 1963 г. Вычисленные с помощью этой программы (и проверенные иным методом) 3922 значащих цифры  $\sqrt{2}$  указаны в [1]. Второй вариант программы послужил для составления таблиц квадратных корней из натуральных чисел от 1 до 9999, с точностью до 50 цифр после запятой.

Отпечатанные на телетайпе таблицы находятся в кафедре вычислительной математики Вроцлавского университета. Взятые из этой таблицы корни 120 начальных простых чисел приведены дальше.

Третий вариант программы, разработанный в январе 1964 г., хранится в библиотеке подпрограмм машины Эллиотт 803 под названием *WU1*. С помощью *WU1* можно вычислить 3946 цифр  $s_k$  корня из  $n'$ . Время вычислений — около 9 ч.; 350 начальных цифр получается в 4 мин. 35 сек.

Программа *WU1* приведена на стр. 42. Для произведения пуска этой программы следует с пульта управления заслать в машину команду передачи управления 40 5, установить на пульте число  $n$  и выполнить ту команду. Результаты печатаются в виде, который поясняется следующим примером:

$$SQRT(3) = 1.7320508075688772935274463415058723669428052538103 \\ 8062805580697945193301690880003708114618675724857 \\ 5675626141415406703029969945094998952478 \text{ итд.}$$

$\sqrt{n}$

2	1,41421	35623	73095	04880	16887	24209	69807	85696	71875	37694
3	1,73205	08075	68877	29352	74463	41505	87236	69428	05253	81038
5	2,23606	79774	99789	69640	91736	68731	27623	54406	18359	61152
7	2,64575	13110	64590	59050	16157	53639	26042	57102	59183	08245
11	3,31662	47903	55399	84911	49327	36670	68668	39270	88545	58935
13	3,60555	12754	63989	29311	92212	67470	49594	62512	96573	84524
17	4,12310	56258	17660	54982	14098	55974	07702	51471	99225	37362
19	4,35889	89435	40673	55223	69819	83859	61565	91370	03925	23244
23	4,79583	15233	12719	54159	74380	64162	69391	99967	07041	90412
29	5,38516	48071	34504	03125	07104	91540	32955	62951	20161	64478
31	5,56776	43628	30021	92211	94712	98918	54952	04763	93377	57041
37	6,08276	25302	98219	68899	96842	45202	06706	20849	70094	78641
41	6,40312	42374	32848	68648	82176	74621	81326	45204	20132	62101
43	6,55743	85243	02000	65234	41099	97636	00162	79269	66319	88378
47	6,85565	46004	01044	12493	58714	49084	84896	04606	43461	00132
53	7,28010	98892	80518	27109	73024	91527	03279	37776	69682	57647
59	7,68114	57478	68608	17576	96870	21731	37247	30624	51077	61488
61	7,81024	96759	06654	39412	97227	35759	10141	35683	05136	64856
67	8,18535	27718	72449	96995	37037	24733	92945	88804	86815	49803
71	8,42614	97731	76358	63063	41399	06202	73603	16080	02401	56075
73	8,54400	37453	17531	16787	16483	26239	70643	45944	55329	53328
79	8,88819	44173	15588	85009	14416	75408	72781	70764	50603	72952
83	9,11043	35791	44298	88194	56261	04688	66919	00991	39168	26495
89	9,43398	11320	56603	81132	06603	77622	64071	69836	22633	41512
97	9,84885	78017	96104	72174	62114	14917	62448	16961	36287	44276
101	10,04987	56211	20890	27021	92649	12759	57618	69450	23470	02637
103	10,14889	15650	92219	46864	85201	18935	87438	35819	22501	88840
107	10,34408	04327	88600	46973	85994	42626	99736	83020	22374	97168
109	10,44030	65089	10550	17975	77540	22548	04767	82898	49837	71979
113	10,63014	58127	34649	40799	91219	14929	31422	19672	64258	58259

<i>n</i>	<i>Vn</i>												
127	11,26942	76695	84644	88285	00036	91742	81735	47453	11603	74752			
131	11,44552	31422	59597	03904	34342	88192	18413	20491	11807	01030			
137	11,70469	99107	19625	10910	20982	74228	62108	88233	10518	28184			
139	11,78982	61225	51595	96846	91837	51358	48942	61080	49943	10659			
149	12,20655	56157	33702	95189	78552	56622	95413	60871	27820	38084			
151	12,28820	57274	44507	59181	21635	54641	85137	44987	48978	99300			
157	12,52996	40861	41667	78849	53648	15752	86586	10216	36354	31072			
163	12,76714	53348	03704	66171	09520	09780	89234	73823	63780	30125			
167	12,92284	79833	20085	46884	48108	04605	84094	96249	37656	15290			
173	13,15294	64379	65905	43996	24871	93129	09318	00991	08972	72938			
179	13,37908	81602	59652	01502	68813	54786	30705	79814	04629	17532			
181	13,45362	40470	73710	31716	30854	62170	40419	38709	62385	52012			
191	13,82027	49610	85253	31390	44805	67753	45600	52191	89818	08841			
193	13,89244	39894	49804	50843	25470	41028	55442	93173	93362	32773			
197	14,03566	88476	18199	63051	94613	86407	04570	85634	42943	67413			
199	14,10673	59796	65884	42523	21636	90877	32647	79387	08129	20384			
211	14,52583	90463	33950	06832	87578	45437	27547	11787	71555	43121			
223	14,93318	45230	68078	66292	22905	13494	60490	24616	80005	47787			
227	15,06651	91733	19363	61545	74987	07326	73426	94646	73691	81746			
229	15,13274	59504	21555	92719	18620	49341	06521	17248	75483	85197			
233	15,26433	75224	73748	02525	59487	00600	47958	73286	93478	60982			
239	15,45962	48337	40306	51965	08428	83328	15250	08347	27989	67737			
241	15,52417	46962	60023	72886	24617	67172	98070	29524	22726	77067			
251	15,84297	95177	54859	48475	67960	52047	58481	88506	97302	70981			
257	16,03121	95418	81397	36487	13547	57688	48220	80975	88144	16754			
263	16,21727	47402	26854	77424	26831	73270	43388	45988	30279	10847			
269	16,40121	94668	56725	16307	47132	27392	48461	03023	94288	81248			
271	16,46207	76331	54327	94767	01816	28178	44821	18218	62632	96002			
277	16,64331	69770	93238	06892	82147	85345	98654	80892	46338	60120			
281	16,76305	46142	40210	12844	56955	56518	07275	86006	84987	54639			
283	16,82260	38412	60722	02620	47905	41291	20142	88899	01879	52346			
293	17,11724	27686	23689	76315	29522	94429	30964	07420	99478	28008			
307	17,52141	54679	35231	85874	59627	34792	17795	22553	52451	02902			
311	17,63519	20885	48397	77642	28836	07629	19173	26792	36317	50352			
313	17,69180	60129	54132	54679	00351	71089	80047	70902	90248	79081			
317	17,80449	38147	64855	59454	65587	81474	23613	35625	80577	58925			
331	18,19340	53986	60251	91738	44462	09150	26571	26055	01887	15845			
337	18,35755	97506	85819	29849	17195	18707	17255	12290	08256	98319			
347	18,62793	60101	97157	64258	63277	60956	53449	25746	70373	97870			
349	18,68154	16922	69404	34847	08194	84683	05817	12138	29456	20971			
353	18,78829	42280	55935	99904	52048	38698	96352	47523	28989	14220			
359	18,94729	53214	96416	68217	16255	67751	96936	52689	37188	64068			
367	19,15724	40606	68016	66041	81623	01717	93733	54906	79204	91728			
373	19,31320	79158	27965	83858	74750	87774	20364	52608	83343	76486			
379	19,46792	23339	31785	02794	39928	93822	98082	70211	46324	04207			
383	19,57038	57907	80926	75911	00943	35362	40340	03349	65368	70099			
389	19,72308	29233	16019	96806	39990	83247	80379	97324	68009	44913			
397	19,92485	88451	71275	13996	62591	89487	10972	22091	13072	03723			
401	20,02498	43945	00785	72769	72121	48322	60542	14864	51312	91593			
409	20,22374	84161	56684	37951	29875	59240	03257	94388	17108	42299			

*n*

$\sqrt{n}$

419	20,46948	94904	58720	27997	97225	35443	04990	30593	58059	08710
421	20,51828	45286	83191	06251	97381	99141	05908	85257	55356	25120
431	20,76053	94920	26694	43962	24962	85341	07173	52946	46809	92962
433	20,80865	20466	84811	63077	00015	22103	35293	24738	35516	95997
439	20,95232	68397	56962	89777	66779	85587	94557	55368	29287	91844
443	21,04756	51798	49188	43710	11425	43474	38025	04077	60078	44961
449	21,18962	01004	17090	77171	76589	51520	04088	33114	44424	65392
457	21,37755	83264	31950	11895	61020	37832	27440	27245	66939	12150
461	21,47091	05535	83888	46920	96987	81039	02216	55573	38500	13356
463	21,51743	47913	50013	52310	77296	89730	04275	55997	74038	75043
487	21,61018	27849	74309	42079	37154	19611	15452	03281	24186	68455
479	21,88606	86282	39289	15224	50621	43684	73558	16223	54663	31169
487	22,06807	64907	13910	88194	22605	52057	84778	65428	54603	35392
491	22,15851	98061	60338	38059	09777	20375	92804	67492	52406	85534
499	22,33830	79036	88676	66083	14514	38894	28915	44207	59131	94738
503	22,42766	14920	05803	85824	20380	30739	45348	29021	44088	27432
509	22,56102	83453	56955	38183	30744	81148	62079	88421	02168	35106
521	22,82542	44210	26654	81124	37240	69568	69442	74086	40511	93263
523	22,86919	32520	58543	04667	61743	01097	85735	60465	44830	84915
541	23,25940	66992	26014	43654	62457	46405	15603	83939	11546	38314
547	23,38803	11270	52999	54189	25027	80757	03505	20416	58457	42212
557	23,60084	74424	11893	18908	16622	99081	86485	00421	40471	93634
563	23,72762	10354	09344	21420	03688	93288	43502	64087	36717	33967
569	23,85372	08837	53125	69654	57132	38553	00043	29282	84299	93399
571	23,89560	62906	97041	02702	45989	60419	27321	82342	09318	77688
577	24,02082	42989	28627	73237	24515	45390	10979	50595	86976	74959
587	24,22808	28791	71434	76436	94679	24865	88441	24893	74747	95490
593	24,35159	13237	71840	92759	67845	86989	89507	37446	36336	30740
599	24,47447	65010	40834	31567	81446	15826	75340	16748	05297	85821
601	24,51530	13442	62525	70758	38757	48357	13970	00932	55046	50731
607	24,63736	99895	09838	10142	79629	55091	13828	24176	82193	24112
613	24,75883	68062	79894	37614	78109	22693	00584	95259	31596	13190
617	24,83948	46967	48441	20235	36737	33526	54085	20252	91366	81389
619	24,87971	06092	49457	26487	76618	06588	73587	45775	48694	67569
631	25,11971	33741	60940	05539	77455	24820	58449	54471	28285	96756
641	25,31797	78023	44325	46177	72857	46124	29583	43255	33875	74778
643	25,35744	46662	11932	79893	78700	05815	48188	80781	71881	90476
647	25,43610	46839	53808	73819	38412	74330	54433	41475	49915	33717
653	25,55386	46783	61275	22291	69779	94754	22047	66212	32181	99973
659	25,67099	53059	86871	68869	05699	24711	00884	02737	12573	26317

5	70	0 : 74	29	10	74	29 : 00	19
6	45	5 : 20	147	1	74	31 : 00	4
7	74	30 : 00	16	2	74	19 : 00	21
8	74	30 : 00	1	3	74	17 : 00	22
9	74	30 : 00	2	4	74	18 : 00	7

15	74	20 : 00	8	60	73	139 : 40	86
6	74	27 : 00	25	1	30	147 : 52	131
7	74	17 : 46	27	2	57	0 : 07	144
8	51	0 : 02	0	3	41	112 : 00	0
9	21	144 : 20	148	4	00	142 / 30	150
20	73	142 : 40	31	65	42	125 : 02	0
1	47	21 : 74	18	6	05	142 : 45	67
2	26	144 : 74	28	7	22	144 : 30	151
3	74	6 : 30	147	8	04	151 : 24	144
4	74	28 : 52	137	9	00	142 / 22	150
25	54	38 : 43	36	70	30	148 : 24	147
6	20	147 : 22	144	1	40	61 : 04	110
7	44	24 : 74	16	2	20	140 : 30	139
8	74	18 : 74	28	3	04	146 : 46	76
9	74	6 : 74	28	4	00	139 / 30	7
30	74	16 : 44	30	75	20	139 / 74	0
1	26	146 : 30	132	6	22	146 : 22	141
2	20	141 : 30	147	7	30	141 : 04	144
3	50	2 : 26	139	8	42	39 : 30	141
4	05	110 : 45	71	9	00	142 / 42	1
35	22	139 : 40	34	80	30	140 : 54	3
6	30	146 : 07	135	1	20	140 : 50	2
7	20	145 : 36	147	2	04	140 : 44	33
8	50	36 : 44	43	3	21	147 : 02	0
9	30	140 : 04	148	4	27	150 : 22	148
40	42	40 : 74	14	85	22	148 : 40	54
1	44	78 : 30	147	6	26	144 : 02	0
2	24	147 : 30	148	7	20	140 : 30	142
3	54	2 : 42	47	8	20	141 : 07	140
4	20	148 : 02	147	9	41	91 : 42	93
45	04	147 : 07	148	90	00	139 / 40	1
6	45	47 : 22	147	1	22	140 / 30	149
7	20	148 : 32	140	2	55	1 : 40	94
8	05	133 : 45	41	3	22	140 / 30	149
9	73	142 : 40	31	4	00	141 / 52	150
50	30	148 : 52	137	95	57	0 : 24	144
1	57	0 : 10	147	6	02	0 : 07	141
2	52	134 : 57	0	7	40	88 : 02	138
3	20	148 : 26	150	8	05	143 : 46	98
4	02	148 : 05	147	9	22	143 / 30	149
55	41	83 : 30	145	100	21	139 / 30	7
6	05	146 : 20	145	1	20	139 / 74	0
7	04	144 : 20	149	2	30	143 : 05	145
8	07	136 : 20	146	3	45	97 : 30	149
9	26	142 : 26	143	4	74	29 : 24	145

105	74	30 : 26	139	2	42	60 : 30	142
6	74	29 : 32	139	3	05	145 : 41	60
7	74	28 : 05	146	4	73	140 : 44	97
8	00	140 / 45	8190				
9	44	106 : 40	111	125	02	0 : 27	142
				6	73	139 : 40	86
110		+ 250000000000		7	30	147 : 04	144
1	73	140 : 44	97	8	50	38 : 56	131
2	20	139 : 30	130	9	20	147 : 40	64
3	10	139 : 04	148				
4	45	115 : 05	148	130			- 9
				1			+ 10
115	40	117 : 12	139	2			- 12
6	41	113 : 10	139	3			+ 18
7	21	147 : 30	139	4			+ 20
8	05	130 : 20	139				
9	22	142 / 21	150	135			+ 59
				6			+ 68
120	30	142 : 05	138	7			+ 100
1	42	111 : 30	139	8			+ 3945

#### ЦИТИРОВАННЫЕ РАБОТЫ

[1] Dziś i jutro naszej elektroniki, Odra 3.4(1963), 13-22.

[2] E. Viglesio, *Extractione graduale de radice quadrato*, Wiadom. Mat. 27(1926), 53-58.