LISTENING IN TO THE PASCAL

by W. NIJENHUIS *).

681.14

The work of an electronic computer in solving arithmetical problems consists in many cases of running through a succession of repeated cycles, and of cycles within cycles. This will perhaps have become clear from the foregoing articles in this number.

The idea has occurred to many builders of electronic computers to *listen in* to these cycles by making audible through a loudspeaker the passing of numbers through the registers. To this end the loudspeaker is connected to one of the flip-flop circuits in a register of the arithmetical unit. The listener will hear the voltage variations in that flip-flop as the numbers pass. Sometimes the pattern of the voltage variations is so arbitrary and changes so fast that only a hissing noise is heard. Often, however, the cycles mentioned produce a recognizable regularity in the sound, which may even result in a musical tone.

In this way every programme, or part of a programme, produces a characteristic sound by which it can be recognized. The sounds thus offer a means of checking the operation of a computer: the programmer who has become familiar with these characteristic sounds while testing his programme, can later often tell by ear whether the computation is proceeding normally. Use is made of the same facility with the PASCAL, for which purpose the loudspeaker is connected to the last digit (the least significant one) of the S register [1]).

By way of illustration we have brought together on the attached gramophone record some fragments of the sounds which the PASCAL produces in performing the calculations discussed in four of the articles in this issue. The reader will find these fragments in four tracks on side 1 of the record, separated by visible margins; acoustically they are identified by an introductory morse signal. The first fragment consists of three sections, each announced by a morse sign. The fragments are taken from the following calculations:

Clover-leaf cyclotron [2])	Fourier analysis:	— •
	Smoothing:	— • •
	Calibration:	— • • •
Corrugated cardboard trim-losses [3]):		— —
Potential fields and electron trajectories [4]):		— — —
Chessboard puzzle [5]):		— — — —

*) Philips Research Laboratory, Eindhoven.

In the following we shall examine each fragment in turn and comment on the sound pertaining to the various computations. Side 2 of the record contains other sounds produced by the PASCAL, making it possible to go deeper into their relation with the computing operations.

Side 1, first sound fragment (— •)

The sound of the first section of the first fragment, which relates to the Fourier analysis of measurement data of the clover-leaf cyclotron [2]), can be roughly represented phonetically as:

groom tik t toe-doe-de-doe-da-de-dee-doo-da

followed by a few times "tik-tik", and the whole thing is repeated almost identically a number of times.

During the first "groom" 60 numbers are fed in from the punched tape; these give the results of measurements of the azimuthal variation of the magnetic field at a given radius in the air gap of the cyclotron, for a particular choice of parameters (correction currents, etc.). In the subsequent interval one or two lines are printed, concerning the choice of parameters. During the short "tik" noise the measurements are reduced with the aid of a 7th degree curve (see — • • •) to magnetic induction values, and the deviations from the average field are computed. The composite sound now following is produced by the Fourier analysis proper. It comprises the following operations: the Fourier coefficients of the order 0 (constant term), 3, 6 etc., are successively computed; the sum of the Fourier series obtained up to a certain order, calculated at each of the 60 field points, is subtracted from the field value measured at each point; an autocorrelation is computed for the 60 differences; if this turns out to be too large, the procedure is repeated with the Fourier series to the next higher order. As a rule, working to 6 orders (i.e. using the constant term and 12 Fourier terms)

[1]) See Philips tech. Rev. **23**, 1-18, 1961/62 (No. 1), especially p. 4.
[2]) N. F. Verster and H. L. Hagedoorn, Philips tech. Rev. **24**, 106-120, 1962/63 (No. 4/5).
[3]) H. W. van den Meerendonk and J. H. Schouten, Philips tech. Rev. **24**, 121-129, 1962/63 (No. 4/5).
[4]) C. Weber, Philips tech. Rev. **24**, 130-143, 1962/63 (No. 4/5).
[5]) A. J. Dekkers and A. J. W. Duijvestijn, Philips tech. Rev. **24**, 157-163, 1962/63 (No. 4/5).

the autocorrelation of the residue is small enough; the answers are then prepared for printing and are printed in a few lines (the series of sounds "tik tik tik . . .").

In the following sequence of sounds beginning with "groom" the same computation is carried out for the series of measurements relating to the next radius value, and so on.

— • •

The "smoothing" process produces a sound roughly like:

te-te-te-te-te-feet,

and this is repeated a number of times. The process consists each time in deriving a "smooth" table $\bar{y}_i$ from the originally computed table y_i of one of the Fourier coefficients, for 32 circles i of increasing diameter, such that there is a "smooth" transition between the values of that Fourier coefficient for successive circles. The machine computes $\bar{y}_i$ so that

$$\sum_i [\delta^4(\bar{y}_i)]^2 + K(\bar{y}_i - y_i)^2$$

is minimized with a certain selected value of the factor K; $\delta^4(\bar{y}_i)$ represents here the fourth difference of $\bar{y}_i$. First of all K is assigned a large value; $\bar{y}_i$ is then still close to y_i, and the autocorrelation of the differences $\bar{y} - y$ is small. A smaller K makes $\bar{y}$ "smoother" as the 4th differences now have more influence. Each "te" in the sound corresponds to a step in K. The process is stopped when the autocorrelation exceeds a predetermined value, because this indicates that "information" as well as "noise" is also being smoothed away. The process is then repeated for the next Fourier coefficient, and so on.

— • • •

The third section of the first fragment is very short, lasting less than one second. This part is the sound of the complete computation — by the least squares method — of the 8 coefficients of the 7th-degree polynomial, the graph of which is the line of best fit drawn through 22 calibrated points.

The three sections of this sound fragment strikingly illustrate the computing speed of the PASCAL.

Side 1, second sound fragment (— —)

The second fragment, which relates to the problem of cutting-losses, is in marked contrast to the first. Since the process of "linear programming"" used for solving this problem was discussed only in broad lines in the relevant article [3]), we shall not give an explanation of the sounds produced. It can only be noted that each cycle in the sound corresponds to one complete iteration cycle.

Side 1, third sound fragment (— — —)

As mentioned in the relevant article [4]) the calculations of potential fields and electron trajectories were carried out with an IBM 650. Subsequently, however, the calculations have been programmed for the PASCAL, and the sound reproduced in the third fragment relates to a potential field calculation using this machine, for given boundary conditions (given electrode configuration and voltages on the electrodes). The sound somewhat resembles the clucking of a hen; phonetically it can be represented as a frequent repetition of the group (u being pronounced as in the French "la lune")

kree-lu-dlu-dlu-dlu- . . .

Each of these groups corresponds to an iterative cycle over the entire potential field. For some boundary conditions as many as a hundred such cycles are necessary for convergence.

During each "lu", the machine computes the potential in a row of network points on a line parallel to the axis of the electrode system. The pitch of "lu" is determined by the nature of the computing cycle per network point, and its duration by the length of the relevant row of points. The successive "lu" sounds correspond to the successive rows of points at increasing distance from the axis; the individual notes differ in duration since the length of the outermost rows of points, depending on the electrode configuration, differs from that of the rows close to the axis. The first "kree" corresponds to the calculation for all network points on the axis itself; since a different formula underlies this calculation [5]) the pitch of "kree" differs from that of "lu".

The values of all network-point potentials (e.g. 3000 points) are stored in the drum memory of the PASCAL. When a network-point calculation is in progress, use must be made of the (provisional) potential values at points of three successive rows. These "working data" are transferred for this purpose to the ferrite core store. When the computation is completed for all points on a row, the working data of the row nearest to the axis are returned to the drum and the data for a further row of points, more distant from the axis, are extracted from the drum. This transport process causes the "d" sounds in the "lu-dlu-dlu- . . ." series. At the end of an iteration cycle over the entire network, all the working data last used are stored and the data of the row of points on the axis and of the next two rows of points are extracted from the drum. This longer transport accounts for the interval between each group of "kree-lu-dlu . . ." sounds.

Side 1, fourth sound fragment (— — — —)

The fourth fragment, relating to the solving of a chessboard puzzle [5]), is again quite different in character. First, we hear the sound of the programme being fed in (on punched tape). After about 12 seconds the sound of the actual computing begins. This has entirely the character of noise and continues until the first solution of the puzzle has been found. The PASCAL then types the solution on the typewriter, during which time nothing is heard as nothing is being done in the *S* register. This period of silence, which in reality lasts about 16 seconds, has been shortened on the record to about 5 seconds. After this the machine is again heard searching for the next solution, which takes about 1 second. When this has been typed out (interval again shortened to 5 seconds) the work of computing the third solutions begins, which takes more than 20 seconds, after which the fragment is broken off. The puzzle has eleven different solutions, which takes the machine altogether about 8 minutes to find and type out. During this time the machine has tried out several million possibilities.

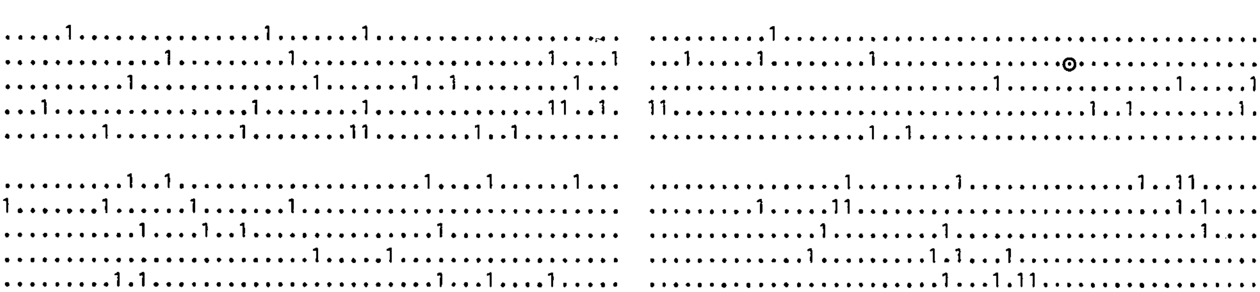
We now turn to side 2 of the gramophone record. The first fragment on this side relates to the search for prime numbers, and for this case we shall give a more detailed explanation of the sounds of computation. The second fragment is musical in character, and the last fragment is a logical complement to it.

Side 2, first sound fragment

The programme for finding prime numbers served for a time as an example to demonstrate the PASCAL in operation. The odd numbers *G*, starting from a predetermined arbitrary number, are tried one by one in the machine to see whether they can be divided by the odd numbers $p \geqq 3$. By using slightly different programmes, the results can be presented by the machine in different ways. The most straightforward method is to let the machine print out each prime number found but not the other numbers. This was the procedure adopted for the present recording. Another, very useful method is that of the "prime number pattern": for each number *G* tried the machine prints a point if *G* is found to be divisible, and another character, e.g. the figure 1, if *G* is found to be indivisible, i.e. is a prime number. In this way, beginning for example with the number 34 359 738 000, the pattern in *fig. 1* was produced for the thousand succeeding odd numbers; the pattern shows very clearly the arbitrary distribution of the prime numbers. A third method of presenting results may be mentioned, as it will prove useful in the following explanations: in this method the machine resolves all consecutive numbers (which may include the even numbers) completely into their factors and prints out the results. A table is then obtained as shown in *fig. 2* for 62 numbers from the pattern in fig. 1, beginning at the place marked with a circle (the number $2^{35} + 1$). This is precisely the series of numbers covered by the machine during the recording of the present fragment. The prime numbers in this series can immediately be recognized (marked by a star in fig. 2).

Let us now consider the computing sound. In the search for large prime numbers the sound begins as a hissing noise, from which a siren-like wailing develops, ending in a short interval of silence while the prime number found is being printed. Sometimes the wail ends in a hissing noise again, indicating that, after some computation, the number tried has proved

34359738000



34359740000

Fig. 1. Prime-number pattern for the 1000 odd numbers between 34 359 738 000 and 34 359 740 000. The number $2^{35} + 1$, with which the machine begins computing in the recorded fragment, is marked with a circle. — Note the "pairs" of prime numbers occurring here and there in the pattern. Experience has shown (although there is as yet no proof) that such pairs continue to occur no matter how far the search is pursued.

```
    34359738369=3.11.43.281.86171
    34359738371=7.4908534053
    34359738373=59.582368447
    34359738375=3.5.5.5.811.112979
    34359738377=17.19.106376899
    34359738379=97.103.149.23081
    34359738381=3.3.3.3.1427.297263
    34359738383=163.883.238727
    34359738385=5.7.29.33851959
    34359738387=3.13.23.38305171
a   34359738389=20249.1696861
    34359738391=11.3123612581
    34359738393=3.347.1699.19427
    34359738395=5.6871947679
    34359738397=673.51054589
    34359738399=3.3.7.7.77913239
b   34359738401=37511.915991
    34359738403=53.3319.195329
    34359738405=3.5.2290649227
c   34359738407=132949.258443
    34359738409=157.3109.70393
    34359738411=3.17.2221.303341
    34359738413=7.11.13.34325413
    34359738415=5.19.361681457
    34359738417=3.3.3817748713
    34359738419=467.73575457
d  *34359738421=34359738421
    34359738423=3.37.309547193
    34359738425=5.5.1374389537
    34359738427=7.42461.115601
    34359738429=3.31.181.277.7369
    34359738431=419.1583.51803
    34359738433=23.1493901671
    34359738435=3.3.3.5.11.23137871
e   34359738437=29077.1181681
    34359738439=13.67.39448609
    34359738441=3.7.41.39906781
    34359738443=29.1184818567
    34359738445=5.17.3343.120919
    34359738447=3.10529.1087781
    34359738449=1171.29342219
f  *34359738451=34359738451
    34359738453=3.3.19.89.2257687
    34359738455=5.7.43.47.485753
    34359738457=11.107.139.210019
    34359738459=3.11453246153
    34359738461=61.563274401
    34359738463=1571.21871253
    34359738465=3.5.13.173.937.1087
g  *34359738467=34359738467
    34359738469=7.193.25432819
    34359738471=3.3.3817748719
h  *34359738473=34359738473
    34359738475=5.5.15107.90977
    34359738477=3.73.1543.101681
    34359738479=11.17.23.167.47837
    34359738481=617.5003.11131
    34359738483=3.7.439.3727057
    34359738485=5.27541.249517
    34359738487=151.1531.148627
    34359738489=3.3.3.241.5280427
    34359738491=13.19.19.31.59.4003
   *
```

Fig. 2. Table of the printed-out prime factors found for the 62 odd numbers from $2^{35} + 1$ to $2^{35} + 123$.

to be divisible and that the machine has started to examine a following number (and possible further following numbers). This happens in our fragment before the first prime number (*d*) has been found: the machine has then already examined the "difficult" numbers *a*, *b* and *c* (see fig. 2), its efforts each time only being successful with large divisors *p*; the work on the number *c* is particularly audible. Also heard in the fragment is the computation of the prime numbers *f*, *g*, *h*, between which the only rather difficult case is the number *e*.

To explain the strange wail, rising and falling in pitch, we must consider what happens during the programme in the *S* register of the PASCAL, especially as far as it concerns the last digit, to which the loudspeaker is connected. The *S* register is used during the division operations: at the beginning of each division the machine sets the dividend in *S*, and at the end of the division *S* contains the quotient. The programme comprises the steps shown in *fig. 3* [6]).

The PASCAL takes almost exactly 180 μs to complete the thickly drawn cycle. Plainly, then, the investigation of most of the numbers *G* (which prove to be divisible by 3, 5, 7 or some other relatively

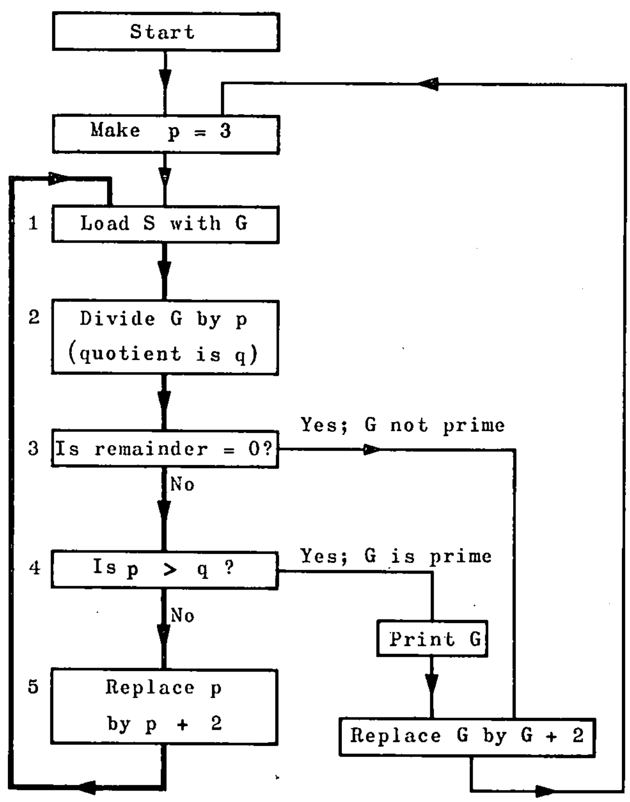


Fig. 3. Computing programme used in the search for prime numbers.

[6]) This programme is certainly not the most economical for the purpose, since *G* is divided by *all* odd numbers *p*, whereas of course the result of a division by, say, 9 or 15 is already established if it is found that the number does not divide by 3 and 5.

small number) takes only a few milliseconds. Only if G is indivisible or contains only very large prime factors is it necessary to continue the divisions up to large divisors p (up to $p \approx \sqrt{G}$ if G is indivisible); the thickly drawn cycle is then repeated numerous times. This is the stage in which the siren wail is heard. All other numbers together contribute only to a hissing sound at the beginning of each such stage.

Let us now consider what happens with the last digit of S during the above-mentioned cycle in the steps 1-5 of the diagram.

1) Since G is always odd, the last digit of S becomes a 1. This remains for about 10 µs.
2) The quotient q is built up in S. During this process numerous ones and noughts pass the last location of S in a fairly irregular pattern. This takes about 75 µs.
3) 4) 5) The quotient q in S remains unchanged, and so too therefore does the content of the last digit of S. This lasts 95 µs.

The voltage variations in the loudspeaker during a single thick cycle thus appear as shown in *fig. 4a* or *b*, depending on whether the quotient q is even or odd. Everything depends now on the alternation between even and odd q.

On a first glance one would expect this alternation to be entirely irregular. The only recognizable periodicity in the loudspeaker voltage is then the fundamental period of the cycle, of duration 180 µs, and all that is strictly repeated in this is the presence of the voltage 1 during step (*1*), which lasts 10 µs. The next following group of pulses in step (*2*) is chaotic, and so is the alternation in the succeeding state, lasting during steps (*3*), (*4*), (*5*). All that can therefore be heard is a note of $10^6/180 \approx 5500$ c/s. rich in overtones and not particularly striking.

If the machine is working on a large number G, however, a certain regularity gradually enters into the alternation of odd and even quotients q. To see how this comes about, we consider the graph of the relation $G = pq +$ remainder: the graph is a stepped curve (*fig. 5*) which follows the hyperbola $PQ = G$ (P and Q are continuous variables instead of the discrete variables p and q) and in which each step arises from an increase of $+2$ in p. We first consider

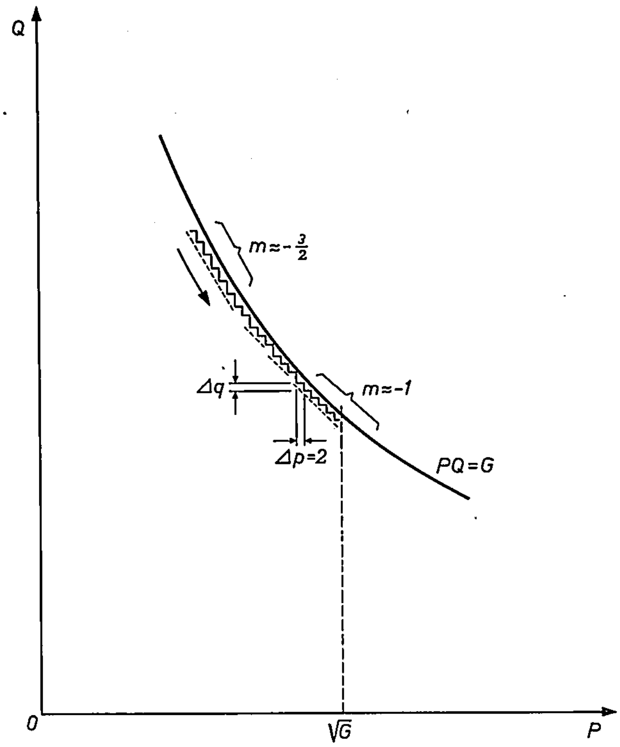


Fig. 5. Hyperbola $PQ = G$ and the stepped curve $G = pq +$ remainder.

the portion of the graph where the hyperbola has the slope $m = -\frac{3}{2}$. If G is large, then each step with $\Delta p = 2$ in this portion has a height $\Delta q = -3$. Here, then, q alternates *in every step* between odd and even, and these alternations give the loudspeaker voltage a period of 2×180 µs (the shortest possible in these alternations) and must therefore be audible as a tone of about 2800 c/s. The same apparently applies to the portions of the curve where the slope is $m = -\frac{5}{2}$, $m = -\frac{7}{2}$ etc., and each of the "summits" in the siren sound means that the machine is then working on such a portion of the hyperbola.

The fact that the sound in between falls and rises again in pitch can be explained in an analogous way. In the portions where the slope m of the hyperbola is equal to a negative *integer* the quotient for each increase $\Delta p = 2$ takes an *even* step $\Delta q = 2|m|$, and thus does *not* alternate between odd and even. A rough estimate shows that the uninterrupted number of repetitions of this even step Δq is of the order of magnitude

$$t_m = k \sqrt[4]{G/|m|^3}, \qquad (1)$$

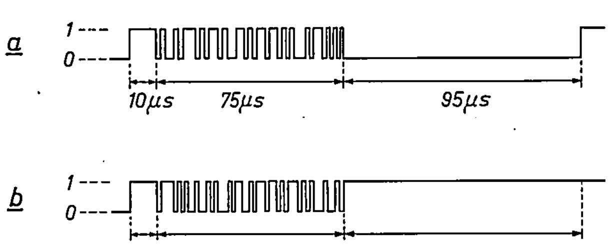


Fig. 4. Voltage variations on the loudspeaker during the thickly drawn cycle in fig. 3.
a) Quotient q even, *b*) q odd.

where k is a factor roughly of the magnitude of $\frac{1}{4}$ or $\frac{1}{5}$. After this series of even steps there will be *one* odd step Δq — that is to say a voltage discontinuity in the loudspeaker — and then again a whole series of even steps, the *series* being roughly t_m in length; this will be repeated over and over again (provided G is very large). The voltage discontinuities near the portion of the hyperbola considered, where m is an integer, thus cause the fairly long period $2t_m \times 180$ μs, and the sound here has a much lower pitch than before. Gradually the series with even step height Δq become shorter the closer we approach the next portion of the hyperbola where m is half of an odd integer, and longer again the nearer the next portion approaches where m is an integer. Accordingly, the period of the voltage discontinuities grows shorter, and longer again. It is this that produces the "wailing" effect.

For $m = \ldots -3, -2, -1$ and $G \approx 2^{35}$ (our recording was made in the region of this G) we find from (1) the respective periods 40×360, 50×360 and 90×360 μs, i.e. roughly the frequencies 70, 55 and 30 c/s. These should be successively the pitches of the sound in the last three "troughs" before the prime number is found, because from $m = -1$ onwards we have $p > q$ ($p > \sqrt{G}$) and the machine need compute no further (see step (4) in fig. 3) This is difficult to test quantitatively as it is not easy to measure the pitch of the computing sounds reliably, presumably owing to the overtones, whose amount is large and continuously changing and also owing to the continuously sliding pitch. Qualitatively, however, the phenomena are sufficiently explained.

Side 2, second sound fragment

From the foregoing it will be clear that the computer can be made to produce a melody by giving it a suitably designed programme for "computation". This is in fact a favourite way of letting visitors know that the machine is entering into the spirit of official opening ceremonies at computing establishments and on similar occassions. The programme by which the PASCAL "sings" is illustrated in *fig. 6*. Every time the machine receives the instruction "Invert S", all ones in the (arbitrary) contents of the S register are replaced by noughts, and vice versa. The result is a voltage alternation on the digit to which the loudspeaker is connected. This is periodically repeated at intervals of t microseconds, depending on the "wait" instructi[illegible], in the PASCAL, is a combination of a [illegible]g" and a repeat instruction — see p. 14 of the article [1]) quoted above). Thus, a tone is produced having a frequency of $10^6/2t$. The cycle is repeated n times, and therefore the tone lasts $n \times t$ microseconds. A list of the successive waiting times t to be observed and the number of repetitions n represents the melody to be "sung" and is stored in the machine's memory.

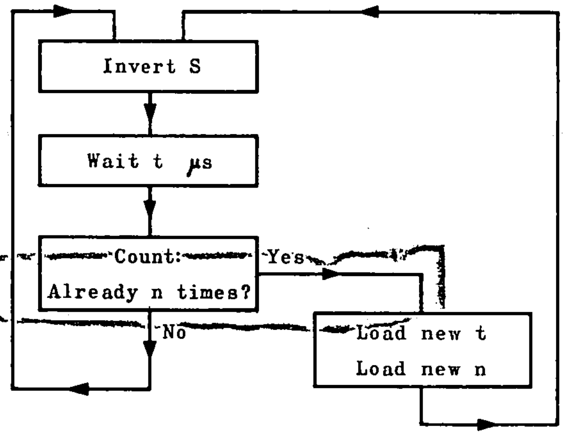


Fig. 6. Programme for "singing" a melody. The melody is stored in the computer's memory in the form of a list of values of t (pitch) and n (duration of tone).

In this way we have programmed a minuet of Mozart, which is heard as the second fragment on side 2 of the record. The "phrasing" of the melody is produced by introducing suitable pauses. During each tone the voltage variations on the loudspeaker are purely periodic, but because they have a square wave form the tones from the PASCAL have a nasal timbre.

Side 2, last sound fragment

The programmes supplied to the PASCAL always have the form of a series of characters punched into a tape, having the meaning of numbers, which according to a certain code represent operations, or of addresses in the memory, or arithmetical numbers proper. Further to what we have just said about the "singing" programme, when supplying melody information to the machine, we can now take the curious step of not using the list of values of t and n that correspond to the notes of the Mozart minuet (or any other piece of music) but of using an arbitrary piece of programme tape, one character on which being always interpreted by the machine as t and the next as n, and so on right along the tape. We may then expect the machine to produce musical tones that together form a kind of "stochastic music" (tonal combinations that are purely random and thus not predictable by any laws of music), a subject which has been a talking point among composers and music theoreticians in recent years. We achieved this by using as "music information" our test programme for the ferrite core store. The result is the last fragment on our gramophone record. Although the music is found to be not entirely "stochastic" — there was apparently already too much regularity in the test programme — the effect is perhaps strange enough to prompt reflection on the nature of what we call a "melody".

Summary. By connecting a loudspeaker to one of the flip-flops in the arithmetical unit of an electronic computer, the passage of numbers through the register concerned can be made audible. In many computing programmes the machine repeats one cycle over and over again; this will often produce in the loudspeaker distinguishable sounds, which in some cases can even be used to check the operation of the computer. A gramophone record attached to the article presents on side 1 some fragments of the sounds produced by the PASCAL in performing computations discussed in four of the articles published in the same issue. On side 2 the computer is heard searching for very large prime numbers, and for this case a closer analysis of the sound is given in the article. The record finally demonstrates the computer "singing" a minuet of Mozart and interpreting an arbitrary programme as a melody; explanatory comments to these sounds are also provided in the article.