1.1	8.44	8.43	1.45	7.44	sy = sx, and
1.44	8.1	1.43	1.45	7.44	nx and ny have
8.1	1.44	1.43	8.45	7.44	a different #
8.44	1.1	8.43	8.45	7.44	of digits.
2.1	12.4	7.3	7.3	0.4	ABS(sx-sy)=10;
2.4	12.1	2.3	2.3	0.4	nx and ny have
12.1	2.4	12.5	13.3	10.4	the same # of
12.4	2.1	12.5	12.3	10.4	digits.
2.1	12.44	12.45	13.43	0.44	ABS(sx-sy)=10;
2.44	12.1	2.43	2.43	0.44	nx and ny have
12.1	2.44	2.43	7.43	10.44	a different #
12.44	2.1	12.45	12.43	10.44	of digits.

- -

By next month I'm sure that you will be able to cook up some great programs that utilize pseudo-signs, and while you're working on that, I'll be finalizing an article on further uses of the SST-RUN maneuver, including certain hex displays which are storable (here comes the HP-25 prompt-word library!), and other NNN's that act up in startling ways. If we both hurry, your programs and my article will be in next month's NOTES! If not, HP-25's the world over will breathe easier!

-- Joseph Horn (# 1537) R/S

TWIX

I first saw the game of Twix in V2N3P31, programmed for the HP 65 by Tony Karp (125). The 67 version is based on that program and has some additional features which there just wasn't room for in the 65's 100 steps. Since the current version takes up less than 112 steps, there is plenty of room to add your own touches, including word/phrase prompts recorded as data on side two. The word phrases can now be keyed up easily in batches of ten, or even 16, with the aid of the black box. (Aren't you glad you've got yours!)

To use the program in its present version, read it in (side one only) and press C. You will see the number 987654321. The object is to re-arrange the digits into the order 123456789 by a sequence of the following moves:

To interchange the two digits on the left, hit A. A will change 987654321 to 897654321.

To interchange the rightmost two digits, hit E. Hitting E will change 987654321 to 987654312.

To shift the entire number one place to the left, hit B. B changes 987654321 to 876543219.

To shift right, hit D. D changes 987654321 to 198765432. These are the only possible moves.

When I first tried the game, it took me almost 50 moves to obtain 123456789 (When you do, the number is shown with a -x- instruction, followed by the number of moves you took.) I found it useful to begin with simpler problems. For example, hit C and then key in the number 123456987 - in which only the last three digits are out of order. Can you change this to 123456789 in as few as 5 moves? Then move on to C, 123459876. Can you straighten it out in only 10 ? If any of you give up, just drop me a line, and I'll send you a copy of my solution to 987654321. I don't know if it's the best solution, but it is under 30 moves. Good Luck.

James Garon (2042) R/S

NUMBER THEORY

INTRODUCTION

Although this article is motivated by the Number Theory Pac in this month's HP-25 Library, it is also an independent introduction for all club members to a fascinating and entertaining branch of mathematics. The field is vast, and we can only touch on a few topics here, but they are ones with calculator applications.

BACKGROUND

Number theory is the branch of mathematics which has as its basis the study of the integers and the various relations among them. As in most areas of modern math, many of the important results in number theory are of a theoretical nature. There are few applications to physical situations and many results would not seem to be directly applicable to computers. As opposed to numerical analysis, number theory is not concerned with obtaining numerical answers (approximations) to mathematical and scientific problems.

The development of number theory may be compared to that of Greek geometry. While the origins of geometry may have been based on physical problems, the Greek development of the subject was based more on the desire to discover new relationships and properties of figures and to understand the nature of mathematical and philosophical reasoning. In number theory many new questions arise out of properties exhibited by certain numbers. The impetus for study is based on curiosity and the desire to understand relationships and properties of numbers.

As an example, 6=3+3, 12=5+7, 30=13+17, and 216=107+109. Is it true that every even number is the sum of two prime numbers? While this question is easy to understand, the answer has been a mystery for 200 years. The affirmative statement is known as Goldbach's Conjecture. The most celebrated number theory problem is known as Fermat's Last Theorem. If n is a positive integral exponent, $n \geq 3$, are there any integers a, b, and c such that $a^n + b^n = c^n$? The complete answer to this question is also unknown but any reader of this column who could give a solution would instantly become the world's most famous mathematician. How many prime numbers are there? Given an integer n, is there a formula which gives the number of primes which are less than n? How dense are the primes in the sequence of integers?

Number theory contains many surprising results. For example, any arithmetic progression of the form $a_n = a + nd$ contains an infinite number of primes, provided the greatest common divisor of a & d is 1. Number theory is known for the difficulty of its problems, however easy those problems may be to state. The subject has influenced the development of higher mathematics and it is surprising that there are few areas of mathematics that are not related to number theory. So while the subject of higher arithmetic may seem trivial, techniques from advanced mathematical analysis are required to solve some of its problems.

Computers have been used with minor success in providing counterexamples to assertions such as those above. The above are typical of the kinds of questions asked in number theory and these in turn give rise to investigations in the more theoretical aspects of the subject. For further reading see Number Theory and Its History by O. Ore, or see History of the Theory of Numbers by L. E. Dickson.

Now let's turn to a few specific topics.

DIVISION ALGORITHM

One of the most fundamental results known about integers is the Division Algorithm:

If a & b are integers with b>0 then there exist integers q & r such that $0 \leq r < b$, and

$$\frac{a}{b} = q + \frac{r}{b} \quad \text{or} \quad a = bq + r$$

All this says is that when you divide an integer a by a positive divisor b you get a quotient q and a remainder r, a fact that all of us were painfully made aware of in grade school. Each time you carry out long division on paper you apply this result for each digit you write in the quotient. Your final answer also represents this result, in which case q may be taken as the entire quotient and r is the final remainder. Think of all the multiples of b as being represented by equally spaced dots on a number line. Then since a is also on the number line, a must lie between two dots, i.e., between two multiples of b. Since the dots are b units apart, the remainder r can always be chosen so that $0 \leq r < b$. On the HP-25, q = INT(a/b) and r = a - bq. This last equation really forms the basis for Integral Digit Slicing. See V4N3P14.

The rule for integer base conversions is also founded on the Division Algorithm. The general procedure for converting an integer from base 10 to base b is to perform successive divisions by b. The resulting remainders will be the digits of the original integer in base b. An example will best illustrate this use of the Division Algorithm. The digits in octal code are always 0-7 and these represent coefficients on powers of 8. For example,

$$3820_{10} = 7 \cdot 8^3 + 3 \cdot 8^2 + 5 \cdot 8^1 + 4 \cdot 8^0$$

Thus,

$$\frac{3820}{8} = \frac{7 \cdot 8^3}{8} + \frac{3 \cdot 8^2}{8} + \frac{5 \cdot 8^1}{8} + \frac{4 \cdot 8^0}{8}$$

$$= (7 \cdot 8^2 + 3 \cdot 8^1 + 5 \cdot 8^0) + \frac{4}{8}$$

(in the form $\frac{a}{b} = q + \frac{r}{b}$)

the virtual elimination of special-case matrices and higher accuracy of the new version should more than compensate. Speaking of accuracy, it is woth noting that if the run is stopped before entering the last phase (routine 8) the stored values are the cofactor array. Cofactors generally have digit lengths comparable to those of the input data and will have experienced no roundoff error. Routine 8 divides each by the determinant to produce the matrix inverse, so the inverse values frequently do have roundoff error. Consequently, if further computations are to be performed using the inverse values, such as multiplication by a column matrix for simultaneous equation solution; maximum accuracy will result if division by the determinant is delayed until the end. To do this replace LBL 8 at step 102 with R/S, then when the run stops copy the displayed determinant value and use the stored array of cofactors as the inverse was to be used. Divide the final answer(s) by the determinant.

I note there is a one-card 5x5 Determinant and Inverse program in the High-Level Math booklet of User Library programs offered by HP, so apparently someone else has written one. I am curious to know what algorithm was used, but not sufficiently curious to pay ten dollars for the booklet. If anyone can enlighten me it will be appreciated.

Hal Brown (362)

R/S

SORTING ON HP-67/97

The June 1977 issue of HP Key Notes described a Bubble Sort Routine (#00619D) for HP 67/97 that sorts as many as 21 data registers using 69 programs steps. I have written programs for the Bubble Sort using only 21 steps for as many as 22 data registers. The HP program can sort in either ascending or descending mode, presumably selecting by keyboard entry. My program can do either but one step in the program has to be changed when using only 21 steps. Keyboard selection of the mode can, of course, be done with 17 additional program steps and about 1/3 longer run time for a maximum disorder sort.

I have also enclosed a Shell Sort routine which has 45 steps but which is much faster than the Bubble Sort. For the full 22 registers at maximum disorder, for example, the Shell Sort runs in less than half the time of the Bubble Sort (or 3 minutes faster!). When the number of registers to be sorted is small -- 6 or less -- the Bubble Sort is a few seconds faster.

I use the Sort routines in a "Racing Results" program for boat racing where the competitors' elapsed times are converted to corrected times. Then, the competitors are sorted into order of corrected finish by the Shell Sort.

This program has greatly speeded up our Sailing Club's time in determining handicap race results as well as reducing errors. Now, if I only had an HP 97 sort that we could post printed results ...!

I hope these programs -- or routines -- will be helpful to the Club Members; and I am sure someone will make my programs more efficient!

A. Babcock Jr., (2154)

R/S

H P - 2 9 C   R O U T I N E S

Memory Register Review

A program for reviewing each storage register in sequence, used to see how many registers have been used or if any data in the indirect registers should be recorded before shutting the machine off. The method is not elegant, and the program stops on "Error" when "30" is addressed by register "i". Register number is shown as integer and data to two decimal places, with a double pause to view data.

Initialize by storing "0" in R_0. (RCL 0 first to avoid destroying desired data in this register).

LBL number and step number are of course arbitrary.

01	G LBL 9	15 13 09	06	F FIX 2	14 11 02
02	RCL 0	24 00	07	F PAUSE	14 74
03	F FIX 0	14 11 00	08	F PAUSE	14 74
04	F PAUSE	14 74	09	G ISZ	15 24
05	RCL i	24 22	10	GSB 9	12 09

Large Factorials

For computing n! for n=70 and larger, this gives the mantissa in the X-register and the power of ten in Y-register, the latter being viewed by X⇄Y. Accuracy is limited by use of logs, but gives 6-7 significant figures up to 500! Not valid for n=0.

To initialize, clear R_1 and put n in R_0.

01	G LBL 6	15 13 06	07	RCL 1	24 01
02	RCL 0	24 00	08	F INT	14 62
03	F LOG	14 43	09	F LSTX	14 73
04	STO + 1	23 51 01	10	G FRAC	15 62
05	G DSZ	15 23	11	G 10^x	15 43
06	GTO 6	13 06	12	G RTN	15 12

Test: FIX6 10!=3.628800 (Y=6)

Program takes 0.85 sec/iteration. For very large factorials (above 500!), Stirling's second-term approximation is more accurate and much faster:

$$\log n! = 1/2(\log 2\pi) + (n+1/2)\log n + [-n + (1/12n)]\log e$$

G. T. Delahunty (2175)

R/S

67 TWIX - AN OLD PPC GAME

```
67 TWIX
001  31 25 13  LBL C
002     31 43  CL REG
003        09  9
004     35 62  1/x
005        43  EEX
006        01  1
007        00  0
008        71  X
009        01  1
010        51  -
011        01  1
012        02  2
013        03  3
014        04  4
015        05  5
016        06  6
017        07  7
018        08  8
019        09  9
020     33 00  STO 0
021        51  -
022     35 22  RTN
023  31 25 11  LBL A
024     31 34  ISZ
025        43  EEX
026        07  7
027        81  ./.
028        41  ENTER
029     31 83  INT
030        01  1
031        00  0
032        81  ./.
033     33 02  STO 2
034     32 83  FRAC
035        43  EEX
036        02  2
037        71  X
038     34 02  RCL 2
039     31 83  INT
040        61  +
041     35 52  X EXCH Y
042     32 83  FRAC
043        61  +
044        43  EEX
045        07  7
046        71  X
047     22 00  GTO 0
048  31 25 12  LBL B
049     31 34  ISZ
050        43  EEX
051        08  8
052        81  ./.
053        41  ENTER
054     31 83  INT
055        43  EEX
056        09  9
057        81  ./.
058        61  +
059     32 83  FRAC
060        43  EEX
061        09  9
062        71  X
063     22 00  GTO 0
064  31 25 14  LBL D
065     31 34  isz
066        01  1
067        00  0
068        81  ./.
069        41  INT
070     32 83  FRAC
071        43  EEX
072        09  9
073        71  X
074        61  +
075     31 83  INT
076     22 00  GTO 0
077  31 25 15  LBL E
078     31 34  ISZ
079        43  EEX
080        02  2
081        81  ./.
082        41  ENTER
083     32 83  FRAC
084        01  1
085        00  0
086        71  X
087     33 02  STO 2
088     32 83  FRAC
089        43  EEX
090        02  2
091        71  X
092     34 02  RCL 2
093     31 83  INT
094        61  +
095     35 52  X EXCH Y
096     31 83  INT
097        43  EEX
098        02  2
099        71  X
100        61  +
101  31 25 00  LBL 0
102     34 00  RCL 0
103     35 52  x EXCH y
104     32 61  X ≠ Y ?
105     35 22  RTN
106     31 84  - X -
107     35 34  RC I
108        84  R/S
```

Set display mode to FIX, DSP 0 before recording.

James Garon (2042)

R/S

T I P S

QUICK NOTE ON MAG CARD HOLDERS- There are at least two types of card holders available from Hewlett-Packard. Every 67/97 owner is familar with the pink plastic type which came with the Standard Pac. This type is also available with the other Application Pacs and the Blank Cards, and are less likely to present any problems for the storage of Mag Cards with rub-on lettering. The other type is a grayish-tan color and are what you recieve when you order the Mag Card Holders alone. Fixatives, rub-on lettering, and even "SHARPIE" pen lettering stick or transfer to the Clear Plastic of this grayish-tan type of card holder and are suitable only for the storage of program cards produced by Hewlett Packard.

M. WALLOW (1431)

R/S