EXPONENTIATION OF POWER SERIES

1. Purpose

Given a formal power series with constant coefficient zero,

$$Q = a_1x + a_2x^2 + \dots ,$$

to calculate the coefficients c_n of the series

$$e^Q = 1 + c_1x + c_2x^2 + \dots .$$

2. Method

To use the differential equation satisfied by the series $S := e^Q$ is not feasible due to the lack of storage facilities. However, from the fact that

$$e^Q = 1 + \sum_{h=1}^{\infty} \frac{1}{h!} Q^h$$

we have, using the notation of the two preceding programs,

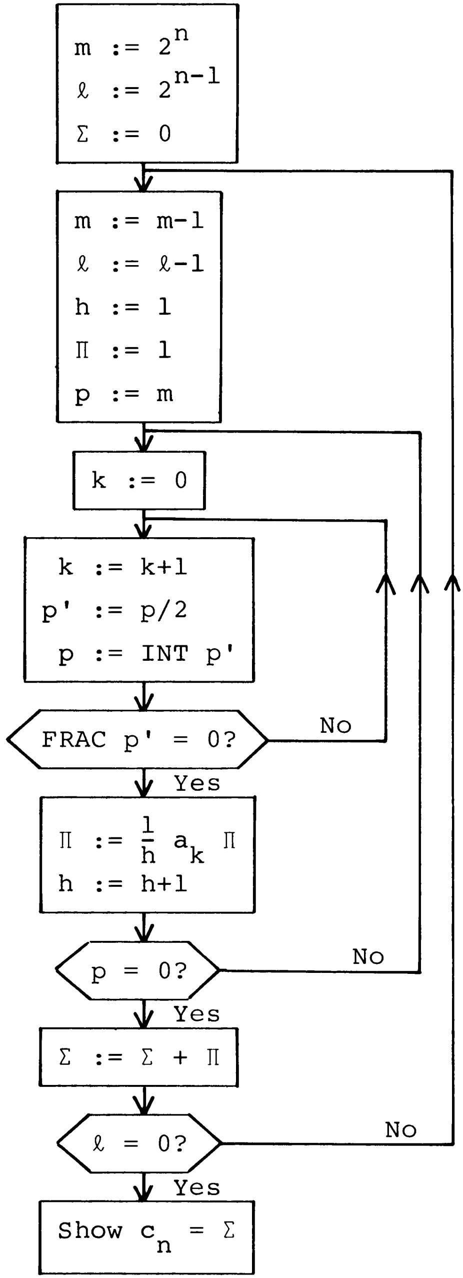
$$c_n = \frac{1}{h!} a_{k_1} a_{k_2} \cdots a_{k_h} ,$$

where the sum comprises all systems of indices $(k_1, k_2, \ldots, k_h)$ satisfying the conditions (2) of the program "Reciprocal Power Series". As before, these systems are coded in terms of the binary representation of the integers m satisfying $2^{n-1} \leqq m < 2^n$, and we thus obtain

$$c_n = \sum_{m=2^{n-1}}^{2^n-1} \frac{1}{h!} \Pi_m ,$$

where Π_m has the same meaning as in the two preceding programs.

3. Flow Diagram



4. Storage and Program

R_0	R_1	R_2	R_3	R_4	R_5	R_6	R_7
	2^n	2^{n-1}	k	h	p	Π	$\boxed{0}$
	m	ℓ					Σ

00		25	
→ 01	1	26	
02	STO 6	27	
03	STO 4	28	
04	STO-1	29	
05	STO-2	30	
06	RCL 1	31	
07	STO 5	32	
→ 08	CLX	33	
09	STO 3	34	
→ 10	1	35	
11	STO+3	36	STO*6
12	RCL 5	37	RCL 4
13	2	38	STO÷6
14	÷	39	1
15	INT	40	STO+4
16	STO 5	41	RCL 5
17	LAST x	42	x ≠ 0
18	FRAC	43	GTO 08
19	x = 0	44	RCL 6
20	GTO 10	45	STO+7
21		46	RCL 2
22		47	x ≠ 0
23		48	GTO 01
24		49	RCL 7

Locations 21 ÷ 35 are to be used for the program to compute a_k. This program should assume k in R_3; at the end of program execution, a_k must be in X register. (If it is more convenient to compute $1/a_k$, this may be done; instruction 36 must then be changed to STO÷6.) Register R_0 is available for auxiliary storage. Unused locations are to be filled with NOP instructions.

5. Operating Instructions

Load the program, including the program to compute a_k. Move the operating switch to RUN. Select the mode of displaying numbers. If coefficient c_n is desired, load

$$
\begin{array}{lll}
2^n & \text{into} & R_1 \\
2^{n-1} & \text{into} & R_2 \\
0 & \text{into} & R_7
\end{array}
$$

(The powers of 2 should be loaded as integers, without using the y^x instruction.) Press

PRGM

R/S

to start computation. The calculator will stop by displaying c_n. (Computing time increases exponentially with n.) Caution: For large n, if some a_k are negative, cancellation may cause c_n to be contaminated by rounding error.

6. Examples and Timing

$\boxed{1}$ $$Q = \mathrm{Log}(1 - x) = - \sum_{k=1}^{\infty} \frac{1}{k} x^k .$$

We should obtain

$$e^Q = 1 - x ,$$

that is, $c_1 = -1$, $c_k = 0$ for $k > 1$. The program to compute a_k is as follows:

```
21    RCL 3
22     CHS
23    GTO 36
24  ⎫
 :  ⎬  NOP
35  ⎭
36    STO÷6
```

Results:

n	c_n	Computing time
1	-1.000000000	3 sec
2	0.000000000	6 sec
3	0.000000000	15 sec
4	$1 * 10^{-10}$	38 sec
5	0.000000000	90 sec

6	0.000000000	3.0 min
7	$1 * 10^{-10}$	8.1 min

2 The Bell numbers b_n are defined by the expansion

$$e^{e^x-1} = \sum_{n=0}^{\infty} \frac{b_n}{n!} x^n .$$

For

$$Q = e^x - 1 = \sum_{k=1}^{\infty} \frac{1}{k!} x^k$$

we may calculate a_k by the program

```
21   RCL 3        28   RCL 3
22     1          29   STO÷6
23     -          30   GTO 37
24   x = 0        31 ⎫
25   GTO 28        : ⎬  NOP
26   STO*3         : ⎪
27   GTO 22       36 ⎭
```

Results:

n	c_n	$b_n=n!c_n$	Computing time
1	1.000000000	1	3 sec
2	1.000000000	2	8 sec
3	0.833333333	5	18 sec

4	0.625000000	15	46 sec
5	0.433333333	52	110 sec
6	0.281944445	203	260 sec
7	0.174007996	877	10 min
8	0.102678571	4140	23 min
9	0.058275463	21147	51 min