I don't offer this program for any serious use; it's for educational purposes only. But it's nice to know that your 'little' HP-25 can do bubble sorts too, crude though they may be.

INSTRUCTIONS:

1) Store data randomly in Ro to R7.
2) Key R/S.
3) Recall data in ascending order from Ro to R7.

Jim Davidson (547) [R/S]

TAU ZERO

The object in developing this program, which has been named after a science fiction novel by Poul Anderson, was to create an educational toy: "toy" because the prescription for program use is a little less restrictive than that for a game, and "educational" because one result of using this program should be an expanded understanding of the real world around us. To be specific, it is pretty well accepted now that the "classical" equations for the motion of an object under constant acceleration are only valid for velocities that are small compared to the velocity of light ($c \sim 3x10^8$ m/s$\sim 6.7x10^8$ mi/hr). In this program, a "practical" example for which the usual equations are in error is illustrated. Thus it is hoped that a feeling for the way in which space and time are really tied together (at least locally) can be conveyed.

The program itself provides the operator with a deep space vehicle (DSV) capable of sustained accelerations of up to one earth gravity. This is about as much acceleration as one would like to put up with for long periods of time, regardless of the propulsion technology used. The DSV has on board a fully automatic operating system, and a computer which consults the traveler once every earth year. At this time, the computer comments on the success that the traveler has had since the previous consultation in traversing "interesting" regions in the accessible universe. It then displays maps of this universe, as it would be seen from the earth, for both the previous year and then the current year, with the DSV shown thereon. Next, it displays forecasted maps in the same way (without the DSV drawn in) for the next four earth years. A display showing the elapsed time on earth, the elapsed time on board the DSV, and the total time spent in "interesting" regions of the universe, all in years, is then shown. Finally, an input pause is provided, requesting that an acceleration for the next year be specified. If the traveler does not respond, the computer assumes that no acceleration is desired, and the DSV is allowed to "coast" for the next earth year.

The story, however, is not all sweetness and light. The maps that the computer displays point out both interesting areas (8) and dangerous areas (9). Although the dangerous areas are always interesting, a trip through one carries a 50% chance of destruction. Thus in a game where one is trying to remain as young as possible during 100 earth years of travel, the path may be frought with risk as well as with boredom. Similarly, if one attempts to spend as much time as possible, during a 10 year trip for the traveler, in "interesting" regions, a bad judgement might turn a curious traveler into a dead one.

The details of operation are rather simple:

(1) Load program card and initial data card (two sides each).
(2) (Optional) Enter your own seed: $0 < s < 1$ [A]

The journey can be started (or restarted) using either step (3) or step (4):

(3) Begin running with a display: [E]

Output
"_laST_YEar."
$(-\bar{1})$*
"__TOdaY___."
(0)
"daYS_TO_BE."
$(\bar{1})$
(2)
(3)
(4)
"_E_Y_S_."
IEEEYYYSSS.
"aCCElEraTE."
(comment)**

*Numbers in parenthesis denote maps of the universe, as seen from earth, at intervals of one earth year. The number denotes the position in the sequence relative to 0: the date of the earth map which portrays the location of our ship during the current consultation.
**Comments: danger warning (ClOSE Call), area of interest (9rEaT BOSS), etc.

(4) Begin running with input accel.: $-1 \leq a \leq 1$ [C]

(5) During a run, the "aCCElEraTE" cue acts as an input pause, and the cycle then repeats itself. If no input is made during the aCCElEraTE pause, then an input value of 0 is assumed. If input $|a| > 1$, default is $|a|=1$.

To reset all of the starting conditions, a data card prepared as follows must be used:

#0	4.5	#5	"ClOSE Call."	#10	777.7777777	#15	778.8747778
#1	0.0	#6	" dESTrOYEd."	#11	" laST YEar."	#16	797.7877798
#2	0.0	#7	"aCCElEraTE."	#12	" TOdaY ."	#17	797.7878789
#3	0.0	#8	"SOrrY BOSS."	#13	"daYS TO BE."	#18	778.7997879
#4	0.0	#9	"GrEaT BOSS."	#14	777.7737999	#19	787.7989787

#A 0.9302032 #B 997.0 #C 1000.0 #D " E Y S ." #E&I 0.0

Lastly, the maps of the accessible universe (chosen for convenience to be represented by nine discrete sectors arranged in the form of a loop, and displayed as a rolled out, linear array) use the following symbols:

blank	uninteresting region (UR)	⌿	DSV + UR
8	interesting region (IR)	4	DSV + IR
9	dangerous region (DR)	5	DSV + DR

Note: It is very helpful for this program, and for other routines for which the lateral position of a symbol is important, to place a small white marker, cut for example from stickon label material, directly beneath the central LED (just left of the W in W/PRGM) on the frosted strip just below the display window. This provides a very helpful positional reference in comparing one display with another.

For those who might use them elsewhere, the equations used are the exact special-relativistic equations for constant proper acceleration in one spatial dimension. Although I don't know of a simple way to write the case for three spatial dimensions, I think that the equations used here are worth summarizing for reference:

Classical Equations

$$v = a\,\Delta t + v_o; \qquad \Delta x = \tfrac{1}{2}a(\Delta t)^2 + v_o\Delta t = \Delta(v^2)/2a.$$

Special Relativistic Equations

$$n = (a/c)\,\Delta T + n_o = \text{arcsinh}((a/c)\,\Delta t + \sinh(n_o));$$

$$\Delta T \doteq (c/a)\,\Delta n; \qquad \Delta x = (c^2/a)\,\Delta(\cosh n); \quad \Delta t = (c/a)\,\Delta(\sinh n).$$

Identities with other Special Relativistic variables

$$v = c\tanh(n); \quad u = c\sinh(n); \quad dt/dT = \cosh(n) = \frac{1}{\sqrt{1-(\frac{v}{c})^2}}.$$

Variable Definitions

v = velocity of the DSV with respect to the "rest" frame.
x = position coordinate of DSV in the "rest" frame.
t = time elapsed in the "rest" frame (e.g. on earth).
a = acceleration of DSV as seen by the traveler.
T = time elapsed as far as the traveler is concerned.
n = velocity "angle" of DSV with respect to "rest" frame.
u = x component of DSV 4-velocity.
c = speed of light.

The above delta notation is defined by $\Delta z = z - z_o$, where the subscript $_o$ denotes initial value.

Keystrokes for Black Box Pseudoalphanumeric Data Construction

ClOSE_Call.	_dESTrOYEd.	aCCElEraTE.	SOrrY_BOSS.	9rEaT_BOSS.
RtoP	RtoP	RtoP	RtoP	RtoP
0	GTO0	STOx0	XneY?	XneY?
STO+1	REG	STO+7	&+	&+
*LBLB	STO-0	LBLa	*LBL8	*LBL8
DSZ	XleY?	STOx1	STO-4	STO+7
0	GTOE	GSBB	SIN	GSBe
COS	RCL&+	TAN	&+	RtoP

_laST_YEar.	TOdaY___.	daYS_TO_BE.	_E_Y_S_.
RtoP	RtoP	RtoP	RtoP
STO-0	*LBL0	STOx0	*LBL0
*LBLc	GTO.(A,D)	*LBL8	*LBL5
*LBL4	STO+4	RCL0	GTO.4(A,D)
XleY?	PtoR	DSZ(i)	GTO.(A,D)
CHS	*LBL7	STO+4	*LBLE
RCL&+	RCL&+	PtoR	RCL&+

Power Consumption for the HP-67

An addition to V3N7P18 for the HP-67. Turn on is 143 ma. All LED's lit is 198 ma. Running varies a lot with the program from 140 to 165 ma. Low pack comes on at 3.55 Volts. The display can be read correctly down to 3.0 Volts. When the display goes out, all but one digit blanks out quickly. The remaining digit , usualy the tenth, goes out at 2.78 Volts. The low pack LED finaly goes out at about 1.5 Volts! The 67 programs run fine down to 2.7 Volts, where the clock stops. So remember that if your low pack lite comes on and you cannot make it to the adapter/charger till the last digit just goes out, you can still plug in and save loosing program and/or data. Plugging the adapter/charger into the 67, after it is first plugged in to house current, will not hurt the 67 or even stop a running program...

Bob Edelen (339)

(All flags clear) TAU ZERO Program Steps (Deg Mode, Fix 2)

```
* LBLC 312513
Xgt0? 3181
SF2 355102
ABS 3564
1 01
XgtY? 3281
XexY 3552
F?2 357102
CHS 42
STOI 3533
RCL1 3401
SF2 355102
GSB1 312201
STOE 3315
+ 61
ENT 41
x² 3254
1 01
+ 61
x½ 3154
+ 61
ln 3152
ENT 41
ENT 41
GSB1 312201
RCL1 3401
GSB1 312201
- 51
RCLI 3534
Xeq0? 3151
GTO6 2206
/ 81
STO+0 336100
Rdn 3553
RCL1 3401
- 51
STO+1 336101
RCLI 3534
/ 81
* LBL9 312509
STO+2 336102
STOE 3315
1 01
STO+4 336104
RCL0 3400
Xlt0? 3171
SF2 355102
9 09
F?2 357102
STO+0 336100
XleY? 3271
STO-0 335100
PexS 3142
RCL5 3405
STO4 3304
RCL6 3406
STO5 3305
RCL7 3407
STO6 3306
RCL8 3408
STO7 3307
RCL9 3409
STO8 3308
0 00
STO9 3309
9 09
STOI 3533
* LBL8 312508
RCLA 3411
RCLB 3412
x 71
FRAC 3283
STOA 3311
x² 3254
3 03
x 71
INT 3183
1 01
0 00
STO/9 338109
x 71
STO+9 336109
DSZ 3133
GTO8 2208
RCL0 3400
STO+9 336109
PexS 3142
RCL0 3400
INT 3183
7 07
- 51
STOI 3533
10^x 3253
4 04
x 71
PexS 3142
STO-5 335105
RCL5 3405
PexS 3142
ISZ 3134
SPACE 3584
RCLI 3534
10^x 3253
/ 81
FRAC 3283
1 01
0 00
x 71
STOI 3533
GTO(i) 2224
* LBL5 312505
RCL5 3405
PRTX 3184
RCLA 3411
. 83
5 05
XleY? 3271
GTO2 2202
* LBL4 312504
RCLE 3415
STO+3 336103
RCL9 3409
PRTX 3184
GTOE 2215
* LBL3 312503
RCL8 3408
PRTX 3184
* LBLE 312515
PexS 3142
RCL1 3401
PRTX 3184
RCL4 3404
GSB0 312200
RCL2 3402
PRTX 3184
RCL5 3405
GSB0 312200
RCL3 3403
PRTX 3184
RCL6 3406
GSB0 312200
RCL7 3407
GSB0 312200
RCL8 3408
GSB0 312200
RCL9 3409
GSB0 312200
PexS 3142
RCLD 3414
PAUSE 3572
RCLC 3413
RCL4 3404
INT 3183
+ 61
RCLC 3413
x 71
RCL2 3402
INT 3183
+ 61
RCLC 3413
x 71
RCL3 3403
INT 3183
+ 61
PRTX 3184
0 00
RCL7 3407
CF3 356103
PAUSE 3572
F?3 357103
XexY 3552
XexY 3552
GTOC 2213
* LBL0 312500
0 00
Rdn 3553
GSB7 312207
GSB7 312207
GSB7 312207
GSB7 312207
GSB7 312207
GSB7 312207
GSB7 312207
GSB7 312207
GSB7 312207
GSB7 312207
RTN 3522
* LBL7 312507
STOI 3533
STOI 3533
STOI 3533
STOI 3533
STOI 3533
STOI 3533
STOI 3533
STOI 3533
STOI 3533
STOI 3533
RTN 3522
* LBL1 312501
e^x 3252
ENT 41
1/x 3562
F?2 357102
CHS 42
+ 61
2 02
/ 81
RTN 3522
* LBL2 312502
RCL6 3406
R/S 84
* LBL6 312506
LSTX 3582
1/x 3562
RCLE 3415
XexY 3552
x 71
STO+0 336100
LSTX 3582
GTO9 2209
* LBLA 312511
STOA 3311
RTN 3522
```

P. Fraundorf (1025) [R/S]