

APPENDIX 1  
ANSWERS TO PROBLEMS

Chapter 1

1. : GIFT ." BOOKENDS " ;  
: GIVER ." STEPHANIE " ;  
: THANKS ." DEAR " GIVER ." , THANKS FOR THE "  
GIFT ." . " ;
2. : TEN-LESS -10 + ; or  
: TEN-LESS 10 - ;
3. When THANKS was compiled, the definition included a reference to the first version of GIFT (the only version of GIFT at that time). Thus THANKS will always execute the same version of GIFT.

Chapter 2

1. DUP DUP: (1 2 -- 1 2 2 2)  
2DUP: (1 2 -- 1 2 1 2)
2. SWAP 2SWAP SWAP
3. : 3DUP DUP 2OVER ROT ;
4. : 2-4 OVER + \* + ;
5. : 2-5 2DUP - ROT ROT + / ;
6. : CONVICTED-OF 0 ; : HOMICIDE 20 + ;  
: WILL-SERVE ." YEARS " ; : ARSON 10 + ;  
: BOOKMAKING 2 + ;  
: TAX-EVASION 5 + ;
7. : EGG.CARTONS 12 /MOD ." CARTONS AND "  
." LEFTOVERS " ;

Chapter 4

1. 1 0= NOT . 1 ok  
0 0= NOT . 0 ok  
200 0= NOT . 1 ok
2. Don't ask.

3. (assuming the legal age is 18 or over:)

```
: CARD 17 > IF ." ALCOHOLIC BEVERAGES PERMITTED "
      ELSE ." UNDER AGE " THEN ;
```

```
4. : SIGN.TEST DUP 0= IF ." ZERO " ELSE
      DUP 0< IF ." NEGATIVE " ELSE
          ." POSITIVE " THEN THEN DROP ;
      (or anything else that works)
```

```
5. : STARS ?DUP IF STARS THEN ;
```

```
6. : <ROT ROT ROT ;
      :-WITHIN <ROT OVER> NOT <ROT > AND ;
      Or here's a more efficient version, using tricks introduced in
      the next chapter:
      : WITHIN >R 1- OVER < SWAP R> < AND ;
```

```
7. : GUESS ( answer guess -- answer or -- )
      2DUP = IF ." CORRECT! " 2DROP ELSE
      2DUP < IF ." TOO HIGH " ELSE ." TOO LOW "
          THEN DROP THEN ;
```

```
8. : SPELLER DUP ABS 4 > IF ."OUT OF RANGE " ELSE
      DUP 0< IF ." NEGATIVE " ABS THEN
      DUP 0= IF ." ZERO " ELSE
      DUP 1 = IF ." ONE " ELSE
      DUP 2 = IF ." TWO " ELSE
      DUP 3 = IF ." THREE " ELSE
          ." FOUR "
      THEN THEN THEN THEN THEN DROP ;
```

```
9. assuming <ROT and WITHIN are still loaded:
      : 3DUP DUP 2OVER ROT ;
      : TRAP ( answer low-try hi-try -- answer or -- )
          3DUP OVER = <ROT = AND IF ." YOU GOT IT! " DROP ELSE
          3DUP SWAP 1 + SWAP WITHIN IF ." BETWEEN "
          ELSE ." NOT BETWEEN " THEN THEN 2DROP ;
```

### Chapter 5

- |                                |                          |
|--------------------------------|--------------------------|
| 1. */ MINUS                    | 4. : F>C 32 - 10 18 */ ; |
| 2. MAX MAX MAX .               | : C>F 18 10 */ 32 + ;    |
|                                | : K>C 273 - ;            |
|                                | : C>K 273 + ;            |
| 3. a) 0 32 - 10 18 */ . -17 ok | : F>K F>C C>K ;          |
| b) 212 32 - 10 18 */ . 100 ok  | : K>F K>C C>F ;          |
| c) -32 32 - 10 18 */ . -35 ok  |                          |
| d) 16 18 10 */ 32 + . 60 ok    |                          |
| e) 233 273 - . -40 ok          |                          |

## 186 LIST

```

0 ( ANSWERS, CHAP. 6)          EMPTY
1 ( PROBLEMS 1 - 6)
2 : STARS  0 DO ." *" LOOP ;
3 : BOX    0 DO CR DUP STARS LOOP DROP ;
4 : \STARS ( #-of-lines) 0 DO CR I SPACES 10 STARS LOOP ;
5 : /STARS ( #-of-lines) 1 SWAP DO CR I SPACES 10 STARS
6         -1 +LOOP ;
7 : ( USING BEGIN & UNTIL FOR /STARS :)
8 : A/STARS ( #-of-lines) BEGIN CR DUP SPACES 10 STARS
9         1- DUP 0= UNTIL DROP ;
10
11 ( DIAMONDS DEFINED IN TWO STAGES:)
12 : TRIANGLE DO CR 9 I - SPACES
13         I 2* 1+ STARS DUP +LOOP DROP ;
14 : DIAMONDS  0 DO 1 10 0 TRIANGLE
15         -1 0 9 TRIANGLE LOOP ;

```

## 187 LIST

```

0 ( ANSWERS, CHAP. 6, CONT'D)  EMPTY
1
2 ( PROB. 7)
3 : R%  10 */ 5 + 10 / ;
4 : DOUBLED ( AMT INT -- )
5         OVER 2* ROT ROT SWAP 21 1 DO
6         CR ." YEAR " I 2 U.R 3 SPACES
7         2DUP R% + DUP ." BAL " .
8         DUP 2OVER DROP > IF
9         CR CR ." MORE THAN DOUBLED IN " I . ." YEARS " LEAVE
10        THEN LOOP 2DROP DROP ;
11
12 ( PROB. 8)
13 : **  1- ?DUP IF
14     OVER ROT ROT  0 DO OVER * LOOP SWAP DROP THEN ;
15

```

## 188 LIST

```

0 ( ANSWERS, CHAP. 7)          EMPTY
1 ( PROB. 1)
2 : N-MAX  0 BEGIN 1+ DUP 0< UNTIL 1- . ;
3 ( Keeps incrementing the number on the stack by one until
4  it looks negative, which means the limit has been passed.
5  The final 1- sets it back to what it was just before it
6  surpassed the limit.)
7 ( PROB. 2 -- Assume that HUMOROUS and SENSITIVE are
8  both true. The "anded" result is "1". Now assume
9  that ART-LOVING and MUSIC-LOVING are also both true.
10  If we "+" their flags instead of "OR"ing them, we get "2."
11  But 0001 [one]
12  ANded with 0010 [two]
13  gives 0000, which is false.)
14
15

```

## 189 LIST

```

0 ( ANSWERS, CHAP. 7 -- CONT'D)   EMPTY
1 ( PROB. 3)
2 : BEEP ." BEEP " 7 EMIT ;
3 : DELAY 20000 0 DO LOOP ;
4 : 3BELLS BEEP DELAY BEEP DELAY BEEP ;
5
6 ( PROB. 4-a)
7 : F>C -320 M+ 10 18 M*/ ;
8 : C>F 18 10 M*/ 320 M+ ;
9 : K>C -2732 M+ ;
10 : C>K 2732 M+ ;
11 : F>K F>C C>K ;
12 : K>F K>C C>F ;
13 ( PROB. 4-b)
14 : .DEG SWAP OVER DABS
15 <# # 46 HOLD #5 SIGN #> TYPE SPACE ;

```

## 190 LIST

```

0 ( ANSWERS, CHAP. 7 -- CONT'D)
1 ( PROB. 5)
2 : DPOLY ( x -- dU)
3     DUP 7 M* 20 M+ ROT 1 M*/ 5 M+ ;
4 : ?DMAX 0 BEGIN 1+ DUP DPOLY 0 0 D< UNTIL 1- . ;
5 ( ?DMAX /ret/ 17513 ok -- this takes a while)
6
7
8 ( PROB. 6)
9 : BINARY 2 BASE ! ;
10 : 3-BASES
11     17 0 DO CR ." DECIMAL" DECIMAL I 4 U.R 8 SPACES
12     ." HEX " HEX I 3 U.R 8 SPACES
13     ." BINARY" BINARY I 8 U.R 8 SPACES
14     LOOP DECIMAL ;
15

```

## 191 LIST

```

0 ( ANSWERS, CHAP. 7 -- CONT'D)
1 ( PROB. 7 -- It tells you that double-length routines are
2   loaded. Two dots are interpreted as a double-length zero.)
3
4 ( PROB. 8)
5 : .PH# <# # # # 45 HOLD # # #
6     OVER IF 47 HOLD #S THEN #> TYPE SPACE ;
7 ( OVER supplies IF with the low-order cell of the
8   number being converted. This cell contains zero only
9   when conversion has completely "used up" the number.)
10
11
12
13
14
15

```

## 192 LIST

```

0 ( ANSWERS, CHAP. 8)           EMPTY
1 ( PROB. 1-a)
2 VARIABLE PIES 0 PIES !
3 : BAKE-PIE 1 PIES +! ;
4 : EAT-PIE PIES @ IF -1 PIES +! ." THANK YOU "
5           ELSE ." WHAT PIE? " THEN ;
6 ( PROB 1-b)
7 VARIABLE FROZEN-PIES 0 FROZEN-PIES !
8 : FREEZE-PIES PIES @ FROZEN-PIES +! 0 PIES ! ;
9 ( PROB. 2)
10 : .BASE BASE @ DUP DECIMAL . BASE ! ;
11 ( PROB. 3)
12 VARIABLE PLACES 0 PLACES !
13 : M. SWAP OVER DABS <#
14     PLACES @ ?DUP IF 0 DO # LOOP 46 HOLD THEN
15     #S SIGN #> TYPE SPACE ;

```

## 193 LIST

```

0 ( ANSWERS, CHAP. 8 -- CONT'D)   EMPTY
1
2 ( Prob. 4)
3 VARIABLE #PENCILS 6 ALLOT
4 0 CONSTANT RED           2 CONSTANT BLUE
5 4 CONSTANT GREEN         6 CONSTANT ORANGE
6
7 # PENCILS #PENCILS + ;
8
9 23 RED PENCILS !
10 15 BLUE PENCILS !
11 12 GREEN PENCILS !
12 0 ORANGE PENCILS !
13
14 ( To test, we can enter
15     BLUE PENCILS ? 15 ok )

```

## 194 LIST

```

0 ( ANSWERS, CHAP. 8, CONT'D)   EMPTY
1
2 ( PROB. 5)
3 CREATE 'SAMPLES 20 ALLOT ( 10 CELLS)
4 : STARS ?DUP IF 0 DO 42 EMIT LOOP THEN ;
5 : SAMPLES ( index# -- adr ) 2* 'SAMPLES + ;
6 : INIT-SAMPLES ( -- )
7     11 0 DO I 7 MOD I SAMPLES ! LOOP ;
8
9 : PLOT ( -- )
10     11 0 DO CR I 2 U.R SPACE I SAMPLES @ STARS LOOP CR ;
11
12 INIT-SAMPLES
13
14
15

```

```

194 LOAD
PLOT
0
1 *
2 **
3 ***
4 ****
5 *****
6 *****
7
8 *
9 **
10 ***

```

195 LIST

```

0 ( ANSWERS, CHAP. 8)          EMPTY
1 ( PROB. 6)
2 VARIABLE BOARD 7 ALLOT
3 : CLEAR BOARD 10 0 FILL ; CLEAR
4 : SQR BOARD + ;
5 : BAR ." | " ;
6 : DASHES CR 9 0 DO ." -" LOOP CR ;
7 : BOX SQR C@ DUP 0= IF 2 SPACES ELSE
8     DUP 1= IF ." X " ELSE
9     ." 0 " THEN THEN DROP ;
10 : DISPLAY CR 9 0 DO
11     I IF I 3 MOD 0= IF DASHES ELSE BAR THEN THEN
12     I .BOX LOOP CR QUIT ;
13 : PLAY 1- 0 MAX 8 MIN SQR C! ;
14 : X! 1 SWAP PLAY DISPLAY ;
15 : O! -1 SWAP PLAY DISPLAY ;

```

196 LIST

```

0 ( ANSWERS, CH. 9)          EMPTY
1 ( PROB. 1)
2 : COUNTS ' ROT ROT 0 DO OVER EXECUTE LOOP SWAP DROP ;
3
4 ( PROB. 2)
5 ( You can find out by entering
6 - EMPTY -HERE )
7
8 ( PROB. 3)
9 ( You can find out by entering
10 PAD HERE - )
11
12 ( PROB. 4)
13 ( a. No difference. A VARIABLE returns its own pfa.
14 b. A user variable returns the address of a cell in the user
15 table. The dictionary entry, which finds, is elsewhere.)

```

197 LIST

```

0 ( ANSWERS, CHAP. 9, CONT'D)
1 ( PROB. 5, SOLUTION #1)
2 VARIABLE 'TO-DO 10 ALLOT ( 6 CELLS)
3 : TO-DO ( index -- adr) 1- 2* 'TO-DO + ;
4
5 : GREET ." HELLO, I SPEAK FORTH. " ;
6 : SEQUENCE 11 1 DO I . LOOP ;
7 : TILE 10 5 BOX ; ( see answers, Ch. 6)
8 : NOTHING ;
9
10 ' GREET 1 TO-DO ! SEQUENCE. 2 TO-DO !
11 ' TILE 3 TO-DO ! NOTHING 4 TO-DO !
12 ' NOTHING 5 TO-DO ! NOTHING 6 TO-DO !
13
14 : DO-SOMETHING ( index -- ) TO-DO @ EXECUTE ;
15

```

## 198 LIST

```

0 ( ANSWERS, CHAP. 9, CONT'D)
1 ( PROB. 5, SOLUTION #2)
2 VARIABLE 'TO-DO 10 ALLOT ( 6 CELLS)
3 : TO-DO ( index -- adr) 1- 2* 'TO-DO + ;
4
5 : GREET ." HELLO, I SPEAK FORTH. " ;
6 : SEQUENCE 11 1 DO I . LOOP ;
7 : TILE 10 5 BOX ; ( see answers, Ch. 6)
8 : NOTHING ;
9
10 : INIT"TO-DO" ( -- ) 7 1 DO ['] NOTHING I TO-DO ! LOOP
11 ['] GREET 1 TO-DO ! ['] SEQUENCE 2 TO-DO !
12 ['] TILE 3 TO-DO ! ;
13 INIT"TO-DO"
14
15 : DO-SOMETHING ( index -- ) TO-DO @ EXECUTE ;

```

## 199 LIST

```

0 ( ANSWERS, CHAP. 10)          EMPTY
1
2 ( PROB. 1)
3 : CHANGE ( c1 c2 -- ) ( changes c1 to c2)
4   SWAP 228 BLOCK 1024 OVER + SWAP DO
5     2DUP I C@ = IF I C! ELSE DROP THEN
6     LOOP 2DROP ;
7
8 ( PROB. 2)
9 101 LOAD ( RANDOM NUMBERS)
10 : FORTUNE CR 16 CHOOSE 64 * ( block#) BLOCK +
11     64 -TRAILING TYPE SPACE ;
12 ( You'll have to invent your own "fortunes". Edit them
13 into an available block, one per line. Then edit the
14 block number into line 11 above, where indicated.)
15

```

## 200 LIST

```

0 ( ANSWERS, CHAP. 10, CONT'D)
1 ( PROB. 3)
2 : ANIMALS ." RAT OX TIGER RABBITDRAGONSSNAKE HORSE RAM M
3 ONKEYCOCK DOG BOAR " ;
4 : .ANIMAL ( u -- )
5     6 * ['] ANIMALS 3 + + 6 -TRAILING TYPE ;
6 ( .ANIMAL takes an argument from 0 to 11.)
7
8 : (JUNESHEE) ( yr -- )
9     1900 - 12 MOD
10     ." YOU WERE BORN IN THE YEAR OF THE " .ANIMAL
11     46 EMIT ( dot) CR ;
12
13 : JUNESHEE CR
14     ." IN WHAT YEAR WERE YOU BORN? "
15     S0 @ 4 EXPECT 0 >IN ! 1 WORD NUMBER CR (JUNESHEE) ;

```

## 201 LIST

```

0 ( ANSWERS, CHAP. 10, CONT'D)          EMPTY
1 ( PROB. 4)
2
3_: NAME      64 * 202 BLOCK +          24 -TRAILING TYPE ;
4 : HAIR      64 * 202 BLOCK + 24 +    20 -TRAILING TYPE ;
5 : EYES      64 * 202 BLOCK + 44 +    20 -TRAILING TYPE ;
6
7 : LETTER    CR CR          DUP DUP
8   ." DEAR " NAME ." ," CR
9 CR ." YOU'RE THE ONLY ONE FOR ME. LET ME RUN MY FINGERS "
10 CR ." THROUGH YOUR NICE " HAIR ." HAIR. LET ME LOOK INTO "
11 CR ." YOUR DEEP " EYES ." EYES. " ;
12
13 : LETTERS  4 0 DO I LETTER LOOP ;
14
15

```

## 202 LIST

```

0 LATICIA          BLACK          BROWN
1 ALICE            BLONDE         BLUE
2 STACEY           BROWN          HAZEL
3 BARBARA          BROWN          GREEN
4
5
6
7
8
9
10
11
12
13
14
15

```

## 203 LIST

```

0 ( ANSWERS, CHAP. 10, CONT'D)          EMPTY
1 ( PROB. 5)
2 VARIABLE #START 222 #START ! ( file begins at block 222)
3 : ELEMENT ( index -- adr)
4   2* 1024 /MOD #START @ + BLOCK + UPDATE ;
5 ( Test virtual array:)
6 : INIT-ARRAY 500 0 DO I I ELEMENT ! LOOP ;
7 : .ARRAY 0 DO CR I . SPACE I ELEMENT ? LOOP ;
8
9 ( Now make the virtual array into a file:)
10 : AVAILABLE ( -- adr) #START @ BLOCK UPDATE ;
11 0 AVAILABLE !
12 ( Redefine ELEMENT to skip over AVAILABLE:)
13 : ELEMENT ( index -- adr)
14   1+ 2* 1024 /MOD #START @ + BLOCK + UPDATE ;
15

```



## 204 LIST

```

0 ( ANSWERS, CHAP. 10, CONT'D)
1 ( PROB. 5, CONT'D)
2
3 : PUT ( value -- ) AVAILABLE @ ELEMENT ! 1 AVAILABLE +! ;
4
5 : SHOW ( -- ) AVAILABLE @ 0 DO CR I .
6   I ELEMENT ? LOOP ;
7
8 : ENTER ( value1 value2 -- ) SWAP PUT PUT ;
9
10 : TABLE AVAILABLE @ ?DUP IF
11   CR 0 DO I 8 MOD 0= IF CR THEN
12   I ELEMENT @ 8 U.R LOOP CR
13   THEN ;
14
15

```

## 205 LIST

```

0 ( ANSWERS, CHAP. 11) EMPTY
1 ( PROB. 1)
2 : LOADED-BY CREATE , DOES> @ LOAD ;
3
4 ( PROB. 2)
5 : BASED. CREATE , DOES> @ BASE @ SWAP BASE !
6   SWAP . BASE ! ;
7
8 ( PROB. 3)
9 : PLURAL ( adr -- ) CREATE ,
10   DOES> @ SWAP ?DUP IF 0 DO DUP EXECUTE LOOP THEN DROP ;
11 ' CR PLURAL CRS
12 5 CRS
13 : BEEP 7 EMIT 20000 0 DO LOOP ; ' BEEP PLURAL BEEPS
14 4 BEEPS
15

```

## 206 LIST

```

0 ( ANSWERS, CHAP. 11, CONT'D)
1
2 ( PROB. 4)
3 : TURNE [COMPILE] DO ; IMMEDIATE
4 : RETURNE [COMPILE] LOOP ; IMMEDIATE
5 : TRY 10 0 TURNE I . RETURNE ;
6
7 ( PROB. 5)
8 : ASCII 32 WORD 1+ C@ [COMPILE] LITERAL ; IMMEDIATE
9 : STAR ASCII * EMIT ;
10
11 ( PROB. 6)
12 : LOOPS >IN @ SWAP 0 DO DUP >IN ! INTERPRET LOOP DROP ;
13 10 LOOPS CR 30 SPACES STAR
14
15

```