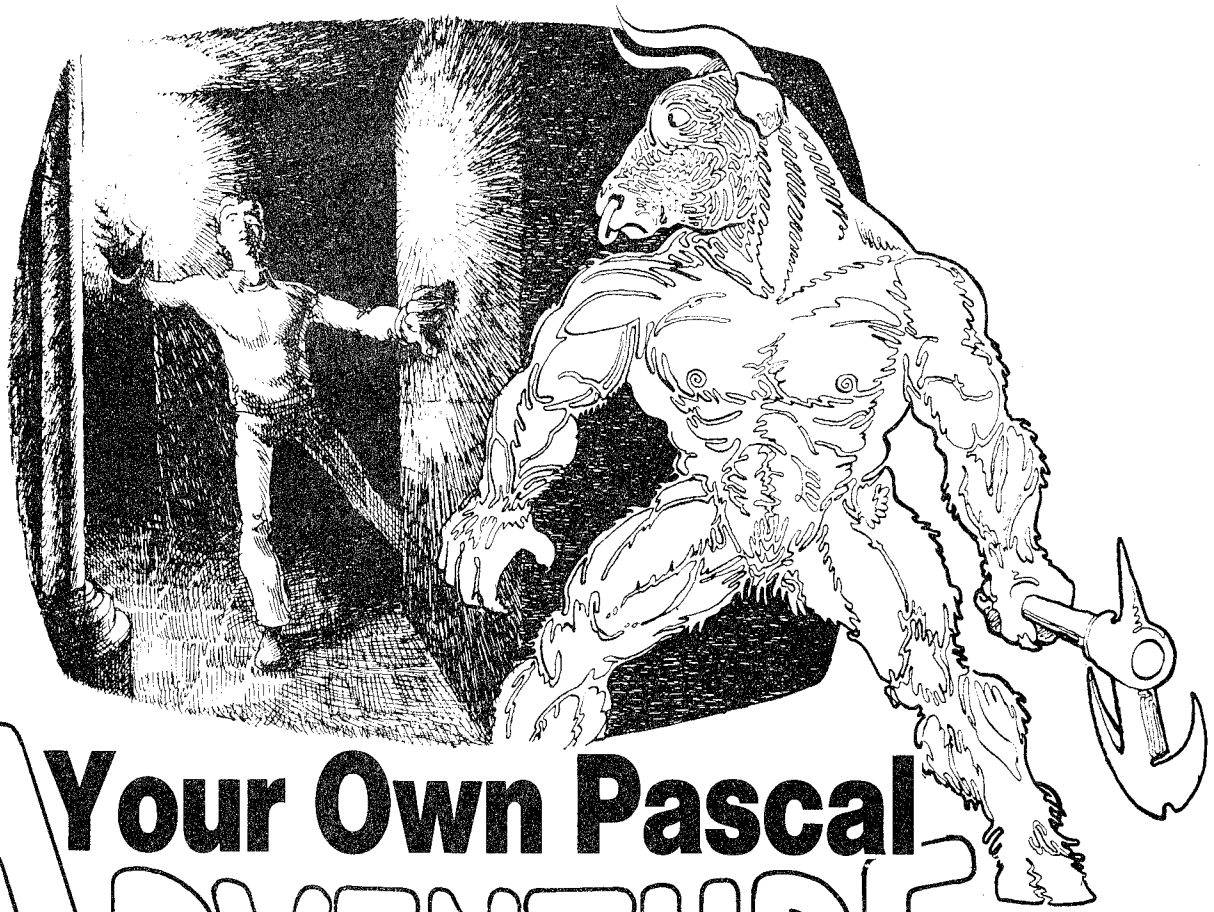

Your Own Pascal ADVENTURE

G-Pascal author Nick Gammon presents a 'skeleton' adventure for our Pascal programmers to develop. It's a fun game on its own, and illustrates how easy it can be to produce such a seemingly complex program using this language. You can play it as is, modify it fairly easily for most Pascals, and turn it into a massive adventure saga...

EVER SINCE Crowther, Woods and Palter wrote the 'original' Adventure game, computerised fantasy games have become very popular. Hardly a month goes by without a computer magazine somewhere printing another adventure-style game for its readers (usually in BASIC).

Probably quite a few of us have felt the desire to write our own adventure, but have been daunted by the seeming complexity of such games.

Writing a complex text-oriented game in Assembler code is tedious, and BASIC doesn't really lend itself to doing the job easily either, as well as being fairly slow.

Fortunately there is relief on the horizon — Pascal turns out to be a very good adventure-game medium, as this article illustrates.

Why Pascal?

Pascal makes the job easy because: Its long data names make the program self-documenting; the CASE statement is very useful for decision-making; and its strong structure makes it easy to debug and enhance.

The listing accompanying the article is a complete game, which will work if keyed in. However it is intended as a 'skeleton' adventure rather than the end product itself. Readers are encouraged to understand how the program works and then add their own plot, room and object descriptions to the game.

The particular Pascal implementation used is G-Pascal, a cheap, high-speed Pascal subset available for the Apple II.

Converting G-Pascal

With minor changes the game should run on other Pascals — you would probably have to rewrite the GETWORD procedure. If your Pascal has no ELSE clause on CASE statements you may need another couple of lines of code to check whether the CASE statements did what is intended.

Adventure games generally accept commands from the player in the form of two word sentences; for example, GET ROD or WALK NORTH (some of the more sophisticated adventure games, such as Zork, can interpret longer and more complicated sentences, such as TAKE EVERYTHING EXCEPT THE BOOK THEN GO NORTH.)

Our game expects commands to consist of a verb, followed in some cases by a noun. For the sake of brevity, movement

Reprinted with permission *Your Computer* magazine, August 1982.

commands such as GO NORTH may be abbreviated to the direction only: NORTH (for example).

Commands are accepted from the player by GETLINE which reads a line of text from the player. GETLINE calls GETWORD twice to decode the text into two words. If GETLINE finds more than two words it asks the player to try again. You could easily accept more than two words by changing the constant MAXWORDS.

GETWORD may seem a bit obscure, but it basically 'packs' the first three letters of the next word on the input line into a single integer (G-Pascal uses 3-byte integers so this is a neat way of storing a three-letter word).

A side-effect of this is that our game only regards the first three characters of the word as significant. GETWORD also keeps track of where the whole word starts and ends, so if you say TAKE ANTELOPE it can reply I SEE NO ANTELOPE HERE rather than I SEE NO ANT HERE.

Our adventure player can take one of five general categories of actions: Move (for example, GO NORTH); Take something (TAKE ROD); Drop something (DROP LAMP); Use something (WAVE RING); and Other (SCORE, QUIT, INVENTORY).

Make A Move...

The player's current location is stored in ROOM. His or her previous location is stored in OLDROOM. If ROOM and OLDROOM differ the program prints a description of the current location by using a CASE statement in DESCRIBEROOM.

```
287   BEGIN
288     WRITE ("YOU ARE ");
289     CASE ROOM OF
290
291 1:
292
293 WRITE ("AT A PLATEAU NEAR A CLIFF.",CR,
294 "A ROCKY PATH LEADS SOUTH.",CR);
295
296 2:
297
298 WRITE ("ON A ROCKY PATH LEADING",CR,
299 "NORTH AND CURVING TO THE EAST. THERE",CR,
300 "IS A SLIGHT BREEZE.",CR);
```

DESCRIBEROOM also prints a description of all objects in the same room as the player:

```
422
423   FOR I := 0 TO MAXOBJ DO
424     IF INROOM(I) THEN
425       WRITE (CR,"THERE IS A ",
426         DESCRIBEOBJECT(I),
427         " HERE!",CR)
```

When the player enters a movement command such as NORTH, SOUTH, UP, DOWN and so on the VERB procedure calls the appropriate movement procedure. The movement procedures handle all movement by a single CASE statement — invalid directions are caught by the ELSE clause of the CASE.

```
477 PROCEDURE SOUTH;
478 BEGIN
479   CASE ROOM OF
480     1:ROOM := 2;
481     4:ROOM := 5;
482     5:ROOM := 6;
483     9:ROOM := 7;
484     7:ROOM := 11;
485   16:ROOM := 15;
486   17:ROOM := 16;
487   19:IF CARRYING(STATUE) THEN
488         ROOM := 18
489       ELSE
490         FORCE
491   ELSE NOWAY END
492 END;
```

Movement can be conditional — in the above example the player can only move from room 19 to 18 if carrying the statue.

To add more rooms to our game we merely have to add their descriptions to DESCRIBEROOM, and make provision for getting to and from them in the movement procedures.

Manipulating Objects

A lot of the fun on adventure games is finding 'objects' (such as lamps and rods) and discovering a use for them.

An object can either be carried, lying around somewhere, or nonexistent.

In our game we use an array, OBJECT, to contain the current location of each object. For example OBJECT (LAMP) is the location of the lamp. The locations are: -1, being carried; 0, nowhere; room number, lying in that room.

We now define some handy boolean functions which tell us if a given object is being carried, is in the room, or is here (meaning carried or in the room).

```
38 ROOM, OLDROOM : INTEGER;
39
40 FUNCTION CARRYING (X);
41 BEGIN
42   CARRYING := OBJECT (X) = INHAND
43 END;
44
45 FUNCTION INROOM (X);
46 BEGIN
47   INROOM := OBJECT (X) = ROOM
48 END;
```

We can now pick up an object by setting its location to -1, or drop it by setting its location to the current room number, for example:

```
121 PROCEDURE PICKUP (X);
122 BEGIN
123   IF HOLDING >= MAXCARRY THEN
124     WRITE ("YOU CAN'T CARRY ANY MORE!",CR)
125   ELSE
126   BEGIN
127     HOLDING := HOLDING + 1;
128     OBJECT(X) := INHAND;
129     WRITE ("TAKEN.",CR)
130   END
```

At this point you might ask: "If I say TAKE LAMP, how does the word 'LAMP' become a subscript in the OBJECT array?"

Good question! This conversion is carried out by CONVERTOBJECT:

```
63 FUNCTION CONVERTOBJECT;
64 BEGIN
65   OBJ := 0;
66   IF WORD(2) = 0 THEN
67     WRITE (SAYWORD(1)," WHAT?",CR)
68   ELSE
```

```
69   CASE WORD(2) OF
70     "LAM":          OBJ := LAMP;
71     "BUN", "CRE": OBJ := BUN;
72     "ROD":          OBJ := ROD;
73     "RIN", "GOL": OBJ := RING;
74     "SIL", "STA": OBJ := STATUE;
75     "JEW", "CRO": OBJ := CROWN
76   ELSE
77     OBJ := 9999
78   END; (* CASE *)
79   CONVERTOBJECT := OBJ
```

CONVERTOBJECT uses a CASE statement to convert the name of any object (or its synonym/s) into a value which is stored in OBJ. (The actual numbers are stored as constants at the start of the program to avoid confusion and make the program more self-explanatory).

CONVERTOBJ also prints 'WHAT?' if the player has not supplied a noun, so if you just say TAKE it will reply TAKE WHAT?

CONVERTOBJECT is called by the boolean function GETOBJECT whose function is to check that the requested object is in the right place.

Since GETOBJECT returns true or false you can write (for example): IF GETOBJECT (CARRIED) THEN statement;

In this case the statement is executed if: a noun was supplied; it is a valid object name; and it is being carried.

If the statement is executed then the object number is in OBJ, otherwise the appropriate error message will already have been printed.

This greatly simplifies programming the rest of the game. For example, here is how we handle the verb EAT:

```
431
432 PROCEDURE EAT;
433 BEGIN
434   IF GETOBJECT(NEARBY) THEN
435     IF OBJ = BUN THEN
436     BEGIN
437       WRITE ("THANKS! YOU WERE RATHER HUNGRY!",CR);
438       DESTROY(BUN)
439     END
440     ELSE
441       CRAZY
442 END;
```

To Add More Objects...

Here's how to add more objects to the game:

- Increase MAXOBJ to the appropriate size.
- Allocate an internal name and sequential number to each object in the CONST declaration at the start of the program (see lines 22 to 29).
- Put the description of each object in DESCRIBEOBJECT.
- Put the external name to internal name conversion in CONVERTOBJECT.
- Allocate the object's initial room number (if any) in INITIALIZE.
- Decide what the use of the object will be and put it in an appropriate procedure. For example, a book might be read, or a key might open a lock.

Expanding The Program

To turn this game into a full-scale adventure you would need to: create more

rooms; create more objects; create more verbs (for example, READ, LOCK, OPEN, magic words and so on.)

It might be necessary to store room descriptions on disk as a text file (for example) if the game becomes too big to fit into memory.

You would probably want to keep the last five or so room descriptions in memory so that you don't keep accessing the disk as the player blunders backwards and forwards between a couple of rooms. Your imagination is the limit — enjoy creating your own scenarios!

```
(* ADVENTURE-STYLE GAME

   AUTHOR: NICK GAMMON *)

CONST

HOME=5;
CR=13;

FALSE=0;
TRUE=1;

INHAND = -1;
MAXOBJ=6;
MAXWORDS=2;
MAXCARRY=3;

CARRIED=1;
NOTCARRIED=2;
NEARBY=3;

(* OBJECTS *)

LAMP=1;
BUN=2;
ROD=3;
RING=4;
STATUE=5;
CROWN=6;

VAR

LINE : ARRAY (100) OF CHAR;
OBJECT : ARRAY (MAXOBJ) OF INTEGER;
WORD, STARTWORD, ENDWORD :
       ARRAY (MAXWORDS) OF INTEGER;
HOLDING, SCORE, TURNS, OBJ, PTR,
ROOM, OLDROOM : INTEGER;

FUNCTION CARRYING (X);
BEGIN
  CARRYING := OBJECT (X) = INHAND
END;

FUNCTION INROOM (X);
BEGIN
  INROOM := OBJECT (X) = ROOM
END;

FUNCTION HERE (X);
BEGIN
  HERE := CARRYING (X) OR INROOM (X)
END;

FUNCTION SAYWORD (X);
VAR I : INTEGER;
BEGIN
  FOR I := STARTWORD(X) TO ENDWORD(X) - 1 DO
    WRITE (LINE(I));
  SAYWORD := 0
END;

FUNCTION CONVERTOBJECT;
BEGIN
  OBJ := 0;
  IF WORD(2) = 0 THEN
    WRITE (SAYWORD(1)," WHAT?",CR)
  ELSE
  CASE WORD(2) OF
    "LAM":          OBJ := LAMP;
    "BUN", "CRE": OBJ := BUN;
    "ROD":          OBJ := ROD;
    "RIN", "GOL": OBJ := RING;
    "SIL", "STA": OBJ := STATUE;
    "JEW", "CRO": OBJ := CROWN
  ELSE
    OBJ := 9999
  END; (* CASE *)
  CONVERTOBJECT := OBJ
END;

FUNCTION DESCRIBEOBJECT (X);
BEGIN
  CASE X OF
    LAMP:   WRITE ("LAMP");
    BUN:    WRITE ("CREAM BUN");
    ROD:    WRITE ("ROD");
    RING:   WRITE ("GOLD RING");
    STATUE: WRITE ("SILVER STATUE");
    CROWN:  WRITE ("JEWELLED CROWN")
  END;
  DESCRIBEOBJECT := 0
END;

FUNCTION GETOBJECT (MUSTBE);
BEGIN
  GETOBJECT := FALSE;
  IF CONVERTOBJECT THEN
  BEGIN
    IF NOT HERE(OBJ) THEN
      WRITE ("I SEE NO ",SAYWORD(2),
             " HERE.",CR)
    ELSE
    CASE MUSTBE OF
      CARRIED:
        IF CARRYING(OBJ) THEN
          GETOBJECT := TRUE
        ELSE
          WRITE ("YOU'RE NOT CARRYING IT!",CR);
      NOTCARRIED:
        IF NOT CARRYING(OBJ) THEN
          GETOBJECT := TRUE
        ELSE
          WRITE ("YOU'RE ALREADY CARRYING IT!",CR)
    ELSE
      GETOBJECT := TRUE
    END
  END
END;

PROCEDURE PICKUP (X);
BEGIN
  IF HOLDING >= MAXCARRY THEN
    WRITE ("YOU CAN'T CARRY ANY MORE!",CR)
  ELSE
  BEGIN
    HOLDING := HOLDING + 1;
    OBJECT(X) := INHAND;
    WRITE ("TAKEN.",CR)
  END
END;

PROCEDURE DROPIT (X);
BEGIN
  HOLDING := HOLDING - 1;
  OBJECT(X) := ROOM;
  WRITE ("DROPPED.",CR);
  IF ROOM = 1 THEN
    IF (INROOM(RING))
    AND(INROOM(STATUE))
    AND(INROOM(CROWN)) THEN
      ROOM := 0;  (* FINISHED QUEST! *)
END;

PROCEDURE DESTROY (X);
BEGIN
  IF CARRYING(X) THEN
    HOLDING := HOLDING - 1;
  OBJECT(X) := 0
END;

PROCEDURE OK;
BEGIN
  WRITE (CR,CR,"OK.",CR)
END;

PROCEDURE TAKE;
VAR I : INTEGER;
BEGIN
  IF WORD(2) = "ALL" THEN
    FOR I := 0 TO MAXOBJ DO
      IF INROOM(I) THEN
      BEGIN
        WRITE (DESCRIBEOBJECT(I),": ");
```

```
165         PICKUP(I)
166       END ELSE
167   ELSE
168   IF GETOBJECT (NOTCARRIED) THEN
169     PICKUP (OBJ)
170 END;
171
172 PROCEDURE DROP;
173 VAR I : INTEGER;
174 BEGIN
175   IF WORD(2) = "ALL" THEN
176     FOR I := 0 TO MAXOBJ DO
177       IF CARRYING(I) THEN
178       BEGIN
179         WRITE (DESCRIBEOBJECT(I),": ");
180         DROPIT(I)
181       END ELSE
182   ELSE
183   IF GETOBJECT (CARRIED) THEN
184     DROPIT (OBJ)
185 END;
186
187 PROCEDURE INSPECT;
188 BEGIN
189   IF GETOBJECT (NEARBY) THEN
190     CASE OBJ OF
191      ROD, RING, STATUE
192          : WRITE ("MAGIC SEEMS TO ",
193            "EMANATE FROM THE ",SAYWORD(2)," ...",CR);
194       BUN: WRITE ("IT LOOKS TASTY!",CR)
195     ELSE
196       WRITE ("YOU SEE NOTHING SPECIAL",CR)
197     END
198 END;
199
200 PROCEDURE LOOK;
201 BEGIN
202   IF WORD(2) = 0 THEN
203     OLDROOM := 0
204   ELSE
205     INSPECT
206 END;
207
208 PROCEDURE CRAZY;
209 BEGIN
210   CASE TURNS MOD 4 OF
211    0:WRITE ("DON'T BE RIDICULOUS!",CR);
212    1:WRITE ("NICE TRY.",CR);
213    2:WRITE ("I WOULDN'T!",CR);
214    3:WRITE ("THAT'S A *VERY* SILLY IDEA!",CR)
215   END
216 END;
217
218 PROCEDURE DONTUNDERSTAND;
219 BEGIN
220   CASE TURNS MOD 4 OF
221    0:WRITE ("WHAT?",CR);
222    1:WRITE ("PARDON?",CR);
223    2:WRITE ("I DON'T UNDERSTAND THAT!",CR);
224    3:WRITE ("EH?",CR)
225   END
226 END;
227
228 PROCEDURE GETWORD (X);
229 VAR I : INTEGER;
230 BEGIN
231   WORD(X) := 0;
232   I := 0;
233   WHILE LINE(PTR) = " " DO
234     PTR := PTR + 1;
235   STARTWORD(X) := PTR;
236   WHILE (LINE(PTR) <> CR) AND
237         (LINE(PTR) <> " ") DO
238   BEGIN
239     IF WORD(X) AND $FF0000 = 0 THEN
240       WORD(X) := WORD(X) + (LINE(PTR) SHL I);
241     I := I + 8;
242     PTR := PTR + 1
243   END;
244   ENDWORD(X) := PTR
245 END;
246
247 PROCEDURE GETLINE;
248 VAR I : INTEGER;
249 BEGIN
250   WRITE (CR);
251   REPEAT
252     READ (LINE);
253     LINE(100) := CR;
254     PTR := 0;
255     FOR I := 1 TO MAXWORDS DO
256       GETWORD (I);
257     WHILE LINE(PTR) = " " DO
258       PTR := PTR + 1;
259     IF LINE(PTR) <> CR THEN
260     BEGIN
261       WORD(1) := 0;
262       WRITE (CR,"PLEASE USE NO MORE ",
263         "THAN ",MAXWORDS#," WORDS",CR,CR)
264     END
265   UNTIL WORD(1) <> 0
266 END;
267
268 PROCEDURE INSTRUCTIONS;
269 BEGIN
270   WRITE (
271 "YOUR QUEST IS TO EXPLORE THE CAVE OF",CR,
272 "THE EVIL UR-LORD, AND BRING BACK TO",CR,
273 "THE EDGE OF THE CLIFF THE FOLLOWING",CR,
274 "VALUABLES:",CR,CR,
275 "  1.  THE WHITE GOLD RING;",CR,
276 "  2.  THE SACRED SILVER STATUE;",CR,
277 "  3.  THE JEWELLED CROWN OF THE UR-LORD.",CR,CR,CR,
278 "BE CAREFUL ....",CR,CR,CR)
279 END;
280
281 PROCEDURE DESCRIBEROOM;
282 VAR I : INTEGER;
283 BEGIN
284   IF (ROOM > 4) AND (NOT HERE(LAMP)) THEN
285     WRITE ("IT'S TOO DARK TO SEE!",CR)
286   ELSE
287   BEGIN
288    WRITE ("YOU ARE ");
289    CASE ROOM OF
290
291 1:
292
293 WRITE ("AT A PLATEAU NEAR A CLIFF.",CR,
294 "A ROCKY PATH LEADS SOUTH.",CR);
295
296 2:
297
298 WRITE ("ON A ROCKY PATH LEADING",CR,
299 "NORTH AND CURVING TO THE EAST. THERE",CR,
300 "IS A SLIGHT BREEZE.",CR);
301
302 3:
303
304 WRITE ("AT THE ENTRANCE TO A DARK",CR,
305 "CAVE. A ROCKY PATH TO THE WEST",CR,
306 "CURVES NORTH.",CR);
307
308 4:
309
310 WRITE ("JUST INSIDE A DARK CAVE.",CR,
311 "LIGHT COMES FROM AN ENTRANCE TO THE",CR,
312 "WEST. THERE IS A DANK, MOULDY SMELL.",CR,
313 "A TUNNEL LEADS SOUTH.",CR);
314
315 5:
316
317 WRITE ("IN A LOW NORTH/SOUTH TUNNEL.",CR);
318
319 6:
320
321 WRITE ("IN AN OVAL CAVERN WITH AN",CR,
322 "EXIT TO THE NORTH. THERE IS A",CR,
323 "FORBIDDING STONE STAIRCASE LEADING",CR,
324 "DOWN.",CR);
325
326 7:
327
328 WRITE ("IN A HIGH, SQUARE CAVE WITH",CR,
329 "WALLS OF FROZEN ICE. THERE ARE",CR,
330 "PASSAGES IN MANY DIRECTIONS, AND",CR,
331 "A STAIRWAY LEADING UP.",CR);
332
333 8:
334
335 WRITE ("IN A TRIANGULAR SIDE-CHAMBER.",CR);
336
337 9:
338
339 WRITE ("IN A MUSTY-SMELLING ALCOVE.",CR);
340
341 10:
342
343 WRITE ("IN A EERIE CHAMBER - SMALL",CR,
344 "SQUEALING SOUNDS COME FROM THE",CR,
345 "WALLS.",CR);
346
347 11:
348
349 WRITE ("PASSING THROUGH AN ENORMOUS",CR,
350 "CAVE WITH A DOUBLE PILLAR OF GREEN",CR,
351 "STONE DOWN THE CENTRE. A BIG ARCH",CR,
352 "LEADS NORTH AND DANK TUNNELS",CR,
353 "LEADS FROM THE SOUTHEAST AND",CR,
354 "SOUTHWEST CORNERS.",CR);
355
356 12:
357
358 WRITE ("CROUCHED IN A MALODOUROUS",CR,
```

```
359 "TUNNEL. THE ONLY EXIT IS THE",CR,
360 "WAY YOU CAME.",CR);
361
362 13:
363
364 WRITE ("A ROOM WHICH APPEARS TO",CR,
365 "ONLY HAVE AN EXIT IN THE NORTHWEST",CR,
366 "CORNER. HOWEVER YOU GET THE",CR,
367 "IMPRESSION THAT OTHERS HAVE",CR,
368 "SOMEHOW TRAVELLED ONWARDS THROUGH",CR,
369 "THIS ROOM - THE EXACT METHOD THEY",CR,
370 "USED IS NOT APPARENT.",CR);
371
372 14:
373
374 WRITE ("A SECRET ROOM REACHED ONLY",CR,
375 "BY MAGIC MEANS. A HIGH PASSAGE",CR,
376 "EXITS TO THE NORTHEAST.",CR);
377
378 15:
379
380 WRITE ("A DEPRESSING OCTAGONAL ROOM.",CR,
381 "EERIE PASSAGES LEAD NORTH AND",CR,
382 "SOUTHWEST.",CR);
383
384 16:
385
386 WRITE ("AN ENORMOUS MISTY CAVERN -",CR,
387 "THE ROOF IS SO HIGH THAT MIST",CR,
388 "OBSCURES IT.",CR,
389 "PASSAGES LEAD NORTH AND SOUTH.",CR);
390
391 17:
392
393 WRITE ("A TINY BOX-SHAPED ROOM. A",CR,
394 "DOOR LEADS SOUTH AND STAIRS LEAD",CR,
395 "DOWN.",CR);
396
397 18:
398
399 WRITE ("A BIZZARE ROOM WITH A SMELL",CR,
400 "OF CHLORINE. A PATH LEADS NORTH AND",CR,
401 "STAIRS LEAD UP.",CR);
402
403 19:
404
405 WRITE ("A STEAMY CHAMBER WITH WARM",CR,
406 "WALLS. FOOTSTEPS IN THE DUST SEEM",CR,
407 "TO LEAD WEST, AND COME FROM THE",CR,
408 "SOUTH.",CR);
409
410 20:
411
412 WRITE ("A LARGE ROOM, LITTERED WITH",CR,
413 "ALABASTER SLABS. DOORS LEAD EAST",CR,
414 "AND WEST.",CR);
415
416 21:
417
418 WRITE ("THE THRONE ROOM OF THE EVIL",CR,
419 "UR-LORD! A LOW DOOR LEADS EAST.",CR)
420
421 END; (* OF CASE *)
422
423   FOR I := 0 TO MAXOBJ DO
424     IF INROOM(I) THEN
425       WRITE (CR,"THERE IS A ",
426         DESCRIBEOBJECT(I),
427         " HERE!",CR)
428   END;
429   OLDROOM := ROOM
430 END;
431
432 PROCEDURE EAT;
433 BEGIN
434   IF GETOBJECT(NEARBY) THEN
435     IF OBJ = BUN THEN
436     BEGIN
437       WRITE ("THANKS! YOU WERE RATHER HUNGRY!",CR);
438       DESTROY(BUN)
439     END
440     ELSE
441       CRAZY
442 END;
443
444 PROCEDURE FORCE;
445 BEGIN
446   WRITE (
447 "AN INVISIBLE FORCE PREVENTS YOU FROM",
448 CR,"PASSING.",CR)
449 END;
450
451 PROCEDURE NOWAY;
452 BEGIN
453    WRITE ("YOU CANNOT GO THAT WAY.",CR)
454 END;
455
456 PROCEDURE NORTH;
457 BEGIN
458  CASE ROOM OF
459    1:CRAZY;
460    2:ROOM := 1;
461    5:ROOM := 4;
462    6:ROOM := 5;
463    7:ROOM := 9;
464   11:ROOM := 7;
465   15:ROOM := 16;
466   16:IF CARRYING(RING) THEN
467        FORCE
468      ELSE
469        ROOM := 17;
470   18:IF CARRYING(STATUE) THEN
471        ROOM := 19
472      ELSE
473        FORCE
474  ELSE NOWAY END
475 END;
476
477 PROCEDURE SOUTH;
478 BEGIN
479  CASE ROOM OF
480    1:ROOM := 2;
481    4:ROOM := 5;
482    5:ROOM := 6;
483    9:ROOM := 7;
484    7:ROOM := 11;
485   16:ROOM := 15;
486   17:ROOM := 16;
487   19:IF CARRYING(STATUE) THEN
488        ROOM := 18
489      ELSE
490        FORCE
491  ELSE NOWAY END
492 END;
493
494 PROCEDURE EAST;
495 BEGIN
496  CASE ROOM OF
497    2:ROOM := 3;
498    3:ROOM := 4;
499    7:ROOM := 10;
500    8:ROOM := 7;
501   20:ROOM := 19;
502   21:ROOM := 20
503  ELSE NOWAY END
504 END;
505
506 PROCEDURE WEST;
507 BEGIN
508  CASE ROOM OF
509    1:CRAZY;
510    3:ROOM := 2;
511    4:ROOM := 3;
512   10:ROOM := 7;
513    7:ROOM := 8;
514   19:ROOM := 20;
515   20:ROOM := 21
516  ELSE NOWAY END
517 END;
518
519 PROCEDURE UP;
520 BEGIN
521  CASE ROOM OF
522    7:ROOM := 6;
523   18:ROOM := 17
524  ELSE NOWAY END
525 END;
526
527 PROCEDURE DOWN;
528 BEGIN
529  CASE ROOM OF
530    6:ROOM := 7;
531   17:ROOM := 18
532  ELSE NOWAY END
533 END;
534
535 PROCEDURE NORTHEAST;
536 BEGIN
537  CASE ROOM OF
538   12:ROOM := 11;
539   14:ROOM := 15
540  ELSE NOWAY END
541 END;
542
543 PROCEDURE NORTHWEST;
544 BEGIN
545  CASE ROOM OF
546   13:ROOM := 11
547  ELSE NOWAY END
548 END;
549
550 PROCEDURE SOUTHEAST;
551 BEGIN
552  CASE ROOM OF
```

```
553    11:ROOM := 13
554   ELSE NOWAY END
555 END;
556
557 PROCEDURE SOUTHWEST;
558 BEGIN
559   CASE ROOM OF
560    11:ROOM := 12;
561    15:ROOM := 14
562   ELSE NOWAY END
563 END;
564
565 PROCEDURE IN;
566 BEGIN
567   CASE ROOM OF
568     3:ROOM := 4
569   ELSE NOWAY END
570 END;
571
572 PROCEDURE OUT;
573 BEGIN
574   CASE ROOM OF
575     4:ROOM := 3
576   ELSE NOWAY END
577 END;
578
579 PROCEDURE WAVE;
580 BEGIN
581     IF GETOBJECT(CARRIED) THEN
582       IF OBJ = ROD THEN
583       BEGIN
584         CASE ROOM OF
585           13:ROOM := 14;
586           14:ROOM := 13
587         ELSE
588           WRITE ("NOTHING HAPPENS HERE.",CR)
589         END
590       END ELSE
591         WRITE ("WAVING THE ",SAYWORD(2),
592                " IS NOT VERY REWARDING!",CR);
593     IF ROOM <> OLDROOM THEN
594       WRITE ("THERE IS A BLINDING FLASH ",
595              "OF LIGHT,",CR,"A LOUD ",
596              "BURPING NOISE, AND YOU ",CR,
597              "SUDDENLY FIND ...",CR,CR);
598 END;
599
600 PROCEDURE GIVESCORE;
601 BEGIN
602     SCORE := 0;
603     IF OBJECT(STATUE) = 1 THEN
604       SCORE := SCORE + 100;
605     IF OBJECT(RING) = 1 THEN
606       SCORE := SCORE + 100;
607     IF OBJECT(CROWN) = 1 THEN
608       SCORE := SCORE + 100;
609     IF CARRYING(STATUE) THEN
610       SCORE := SCORE + 10;
611     IF CARRYING(RING) THEN
612       SCORE := SCORE + 10;
613     IF CARRYING(CROWN) THEN
614       SCORE := SCORE + 10;
615     WRITE (CR,"YOUR SCORE IS ",
616            SCORE#," POINTS, IN ",
617            TURNS#," TURNS.",CR)
618 END;
619
620 PROCEDURE QUIT;
621 VAR REPLY: CHAR;
622 BEGIN
623     GIVESCORE;
624     WRITE (CR,"DO YOU WANT TO QUIT NOW? Y/N ");
625     READ (REPLY);
626     IF REPLY <> "Y" THEN
627       OK
628     ELSE
629       ROOM := 0
630 END;
631
632 PROCEDURE INVENTORY;
633 VAR I, COUNT : INTEGER;
634 BEGIN
635     COUNT := 0;
636     FOR I := 0 TO MAXOBJ DO
637       IF CARRYING(I) THEN
638       BEGIN
639         IF COUNT = 0 THEN
640           WRITE ("YOU ARE CARRYING:",CR);
641         COUNT := COUNT + 1;
642         WRITE ("  ",DESCRIBEOBJECT(I),CR)
643       END;
644     IF COUNT = 0 THEN
645       WRITE ("YOU AREN'T CARRYING ANYTHING!",CR)
646 END;
647
648 PROCEDURE VERB;
649 VAR I : INTEGER;
650 BEGIN
651     WRITE (CR);
652     I := 1;
653     CASE WORD(1) OF
654       "GO", "RUN", "WAL", "MOV",
655       "CRA" :
656       BEGIN
657         IF WORD(2) = 0 THEN
658         BEGIN
659           WRITE (SAYWORD(1)," WHERE?",CR);
660           I := 0
661         END ELSE
662           I := 2
663       END
664     END; (* CASE *)
665     IF I <> 0 THEN
666     CASE WORD(I) OF
667      "QUI", "Q": QUIT;
668      "EAT": EAT;
669      "NOR", "N": NORTH;
670      "SOU", "S": SOUTH;
671      "EAS", "E": EAST;
672      "WES", "W": WEST;
673      "NE"      : NORTHEAST;
674      "NW"      : NORTHWEST;
675      "SE"      : SOUTHEAST;
676      "SW"      : SOUTHWEST;
677      "UP",  "U": UP;
678      "DOW", "D": DOWN;
679      "IN"      : IN;
680      "QUT"     : OUT;
681      "TAK", "T", "GET", "PIC":
682                  TAKE;
683      "DRO", "PUT", "PLA", "THR":
684                  DROP;
685      "SCO"     : GIVESCORE;
686      "INV", "I": INVENTORY;
687      "LOO", "L": LOOK;
688      "WAV"     : WAVE;
689      "INS"     : INSPECT
690     ELSE DONTUNDERSTAND END
691 END;
692
693 PROCEDURE INITIALIZE;
694 VAR I : INTEGER;
695 BEGIN
696     WRITE (HOME);
697     ROOM := 3;
698     OLDROOM := 0;
699     SCORE := 0;
700     HOLDING := 0;
701     TURNS := 0;
702     FOR I := 0 TO MAXOBJ DO
703       OBJECT(I) := 0;
704     OBJECT(LAMP) := 1;
705     OBJECT(BUN)  := 6;
706     OBJECT(ROD)  := 9;
707     OBJECT(RING) := 8;
708     OBJECT(STATUE) := 12;
709     OBJECT(CROWN) := 21;
710 END;
711
712 (* ------------------------------
713    PROGRAM STARTS HERE
714    ------------------------------ *)
715
716 BEGIN
717     INITIALIZE;
718     INSTRUCTIONS;
719     REPEAT
720       TURNS := TURNS + 1;
721       IF ROOM <> OLDROOM THEN
722         DESCRIBEROOM;
723       GETLINE;
724       VERB
725     UNTIL ROOM = 0;
726     WRITE (HOME);
727     GIVESCORE;
728     IF SCORE = 300 THEN WRITE
729      (CR,CR,"CONGRATULATIONS!!",CR,CR,
730       "YOU HAVE COMPLETED YOUR QUEST!",CR,CR)
731 END.
```