```
//                                   M     M    A    RRRRRR   CCCCC  EEEEEEE L
//                                   MM   MM   A A   R     R C     C E       L
//                                   M M M M  A   A  R     R C       E       L
//                                   M  M  M A     A RRRRRR  C       EEEEE   L
//                                   M     M AAAAAAA R   R   C       E       L
//                                   M     M A     A R    R  C     C E       L
//                                   M     M A     A R     R  CCCCC  EEEEEEE LLLLLLL

//                                         J  OOOOO  H     H N     N  SSSSS   OOOOO  N     N
//                                         J O     O H     H NN    N S     S O     O NN    N
//                                         J O     O H     H N N   N S       O     O N N   N
//                                         J O     O HHHHHHH N  N  N  SSSSS  O     O N  N  N
//                               J     J O     O H     H N   N N       S O     O N   N N
//                               J     J O     O H     H N    NN S     S O     O N    NN
//                                JJJJJ   OOOOO  H     H N     N  SSSSS   OOOOO  N     N

//                                           FFFFFFF 7777777 7777777  SSSSS
//                                           F       7    7  7    7  S     S
//                                           F           7       7    S
//                                           FFFFF      7       7     SSSSS
//                                           F         7       7            S
//                                           F         7       7      S     S
//                                           F         7       7       SSSSS
```

```
//   Line    ISN *....*....|....1....|....2....|....3....|....4....|....5....|....6....|....7..*.|....8
//               C NASA (FORMERLY NACA) USED LARGE 'MAINFRAME' COMPUTERS IN FOR EXAMPLE
//               C THE MERCURY, GEMINI AND APOLLO PROJECTS.
//               C IN THE MOVIE 'HIDDEN FIGURES' YOU CAN SEE A IBM 7094 FROM 1962.
//               C LATER, NASA USED IBM 360'S IN THE APOLLO PROJECT.
//
//               C THIS PROGRAM IS AN EXAMPLE OF HOW A PROGRAM FROM THAT TIME MIGHT
//               C HAVE LOOKED LIKE.
//
//               C THIS IS VINTAGE FORTRAN CODE, AS WAS USED IN THE 1960'S. ALL TEXT
//               C IS UPPER-CASE ONLY AND VARIABLE NAMES ARE AT MOST SIX CHARACTERS.
//
//               C THIS PROGRAM USES THE EULER METHODE FOR INTEGRATION THAT WAS USED
//               C IN 'HIDDEN FIGURES'. IT IS MENTIONED WHEN KATHERINE WORKS ON THE
//               C FIRST MANNED FLIGHT IN THE MERCURY PROJECT:
//
//               C KATHERINE: 'THE PROBLEM IS WHEN THE CAPSULE MOVES FROM
//               C             AN ELLIPTICAL ORBIT TO A PARABOLIC ORBIT.
//               C             THERE IS NO MATHEMATICAL FORMULA FOR THAT.
//               C             ...
//               C HARRISON:  'MAYBE WE'RE THINKING ABOUT THIS ALL WRONG.'
//               C STAFFORD:  'HOW'S THAT?'
//               C HARRISON:  'MAYBE IT'S NOT NEW MATH AT ALL.'
//               C KATHERINE: 'MAYBE IT'S OLD MATH.
//               C             SOMETHING THAT LOOKS AT THE PROBLEM NUMERICALLY.
//               C             AND NOT THEORETICALLY. MATH IS ALWAYS DEPENDABLE.'
//               C HARRISON:  'FOR YOU IT IS.' (EXITS)
//               C KATHERINE: 'EULER'S METHOD.'
//               C STAFFORD:  'THAT'S ANCIENT.'
//               C KATHERINE: 'YES. BUT IT WORKS. IT WORKS NUMERICALLY.' (EXITS)
//
//               C THIS PROGRAM IS A SIMPLE DEMO FOR FINDING A STRATEGY TO SAFELY
//               C LAND A 'LUNAR EXCURSION MODULE' (LEM) ON THE MOON.
//
//               C A LEM STARTED ITS DESCENT FROM AN ALTITUDE OF 15 KILOMETER.
//               C IT WEIGHED 16 METRIC TONS AND HAD TO LAND WITH A SPEED NOT
//               C EXCEEDING 1 METER PER SECOND. OTHERWISE THE LEM WOULD GET
//               C DAMAGED, LEAVING ITS ASTRONAUTS MAROONED ON THE MOON.
//
//               C TO ASCEND, THE LEM HAD A SEPARATE ENGINE WITH ITS OWN FUEL.
//               C DURING DESCENT, FUEL COULD NOT BE EXHAUSTED SINCE A RESERVE
//               C WAS NEEDED FOR LAST-SECOND MANEUVERING TO FIND A GOOD LANDING SPOT.
//               C (THIS HAPPENED TO APOLLO 11).
//
//               C HERE THE PROBLEM IS SIMPLIFIED FOR THE SAKE OF DEMONSTRATION.
//               C THE STRATEGY IS THIS:
//               C DESCENT WITH IGNITED ENGINE TO REDUCE SPEED UNTIL SOME HEIGHT 'ROOF'.
//               C UP TO 'ROOF' WE ACCEPT A SAFE HIGH SPEED.
//               C THEN SPEED IS REDUCED.
```

```
//   Line    ISN *....*....|....1....|....2....|....3....|....4....|....5....|....6....|....7..*.|....8
//               C THE FINAL 300M WE REDUCE SPEED TO A LOW FINAL SPEED.
//               C IF DURING SPEED FUEL RUNS OUT WE ARE IN FREE FALL, AND THE SIMULATION
//               C GENERALLY FAILS SINCE THE LEM LIKELY CRASHES.
//
//               C THIS PROGRAM FINDS VALUES FOR 'ROOF' WHERE WE CAN LAND AT A SAFE
//               C SPEED OF AT MOST 1 METER PER SECOND.
//               C WE SOLVE IT HERE BY TAKING INCREASING VALUES FOR 'ROOF' AND
//               C SIMULATE THE LANDING - HOPING THE LEM LANDS SAFELY.
//
//               C ALTHOUGH THIS IS AN OVERSIMPLIFICATION OF THE ACTUAL LANDING
//               C PROCEDURES, RESULTS ARE COMPARABLE TO THOSE ACHIEVED IN ACTUAL
//               C LANDINGS ON THE MOON.
//
//             1       PROGRAM EULER
//
//               C EULER INTEGRATION IS DONE WITH DOUBLE PRECISION.
//             2       IMPLICIT REAL*8 (A-Z)
//
//               C EULER INTEGRATION STEP DETERMINES ACCURACY OF COMPUTATION.
//               C 1/20 SECOND APPEARS GOOD ENOUGH.
//             3       DELTAT = 0.01
//               C FUEL COMSUMPTION IN KG/S AT FULL THROTTLE (ESTIMATED FROM DOCUMENTS).
//             4       CONSUM = 8
//               C THRUST IS 45 KILO-NEWTON (FROM LEM DOCUMENTATION).
//             5       MAXTHR = 45000
//               C GRAVITATIONAL ACCELARATION ON THE MOON IN M/S/S.
//             6       G = 1.62
//               C DESCENT STARTS AT 15 KM.
//             7       ORBIT = 15000
//               C THE LEM HAD 8200 KG FUEL FOR DESCENT. SOME OF THAT WAS USED TO GO
//               C FROM HIGH ORBIT AT 110 K TO THE LOW ORBIT AT 15 KG.
//               C WE MAKE AN EDUCATED GUESS OF THE AMOUNT OF FUEL NEEDED TO GO FROM
//               C 110 KM DOWN TO 15 KG.
//             8       FULL = 8200 - 2000
//               C FROM 'LOW' ON, WE LAND CAREFULLY.
//             9       LOW = 300
//               C WE INCREASE 'ROOF' EVERY ITERATION, START LOW.
//            10       ROOF = ORBIT / 2
//               C DESCENT BEGINS HERE.
//            11     1 SPEED = 0
//            12       FASTST = 0
//            13       TIME = 0
//            14       HEIGHT = ORBIT
//               C FUEL IS FULL WHEN WE START DESCENDING.
//            15       FUEL = FULL
//               C WE RECORD PASSING 'ROOF' AND 'LOW' HEIGHTS.
//            16       ROFTIM = 0
//            17       LOWTIM = 0
```

```
//   Line    ISN *....*....|....1....|....2....|....3....|....4....|....5....|....6....|....7..*.|....8
//               C GAS IS ACTUAL THRUST.
//               C FOR THE LEM THAT WAS 10-60% OF FULL THROTTLE (FROM DOCUMENTATION).
//            18       GAS = 0.1
//               C LEM MASS IS 16 TON WITH FUEL.
//               C THE LEM LOOSES WEIGHT DURING DESCENT (BURNING FUEL).
//            19       EMPTY = 16000 - FULL
//
//               C ENGINE WILL BURN WHILE THERE IS FUEL.
//            20     2 IF (FUEL .GT. 0) GOTO 3
//               C OOPS! NO FUEL, FREE FALL, HELP!
//            21       SPEED = SPEED + DELTAT * G
//            22       GOTO 4
//
//               C WE DO HAVE FUEL.
//               COMPUTE ’SAFE’ SPEED DEPENDING ON HEIGHT.
//            23     3 SAFE = 1
//            24       IF (HEIGHT .GE. LOW) SAFE = 10
//            25       IF (HEIGHT .GE. ROOF) SAFE = 100
//               COMPUTE NEW STATE (EULER, TWICE).
//               CURRENT MASS OF THE LEM.
//            26       MASS = EMPTY + FUEL
//               COMPUTE THRUST FROM NEWTON’S LAW F=M*A.
//            27       THRUST = MAXTHR / MASS * GAS
//            28       SPEED = SPEED + DELTAT * (G - THRUST)
//            29       FUEL = FUEL - DELTAT * CONSUM * GAS
//            30       IF (FUEL .LT. 0) FUEL = 0
//
//            31     4 HEIGHT = HEIGHT - DELTAT * SPEED
//               C RECORD THINGS WE WANT TO KNOW LATER.
//            32       IF (HEIGHT .LT. ROOF .AND. ROFTIM .EQ. 0) ROFTIM = TIME
//            33       IF (HEIGHT .LT. LOW .AND. LOWTIM .EQ. 0) LOWTIM = TIME
//            34       IF (SPEED .GT. FASTST) FASTST = SPEED
//               CLOCK PROGRESSES A TINY BIT.
//            35       TIME = TIME + DELTAT
//
//               C IF THERE IS FUEL LEFT WE CORRECT THRUST.
//            36       IF (FUEL .EQ. 0) GO TO 5
//               C DO WE ASCEND? THEN REDUCE THRUST.
//            37       IF (SPEED .LE. 0) GAS = GAS - 0.05
//               C DO WE DESCEND TOO FAST? THEN INCREASE THRUST.
//            38       IF (SPEED .GE. SAFE) GAS = GAS + 0.05
//               C KEEP THRUST WITHIN LEM PARAMETERS (10-60%).
//            39       IF (GAS .GT. 0.6) GAS = 0.6
//            40       IF (GAS .LT. 0.1) GAS = 0.1
//               C WHILE NOT LANDED, PERFORM NEXT EULER ITERATION.
//            41     5 IF (HEIGHT .GT. 0) GO TO 2
//
//               C REPORT WHEN WE LANDED SAFELY.
```

```
//   ISN *....*....|....1....|....2....|....3....|....4....|....5....|....6....|....7..*.|....8
//    42       IF (SPEED .GT. SAFE .OR. TIME .GT. 1500) GO TO 7
//    43       WRITE (6, 6) ROOF, ROFTIM, LOWTIM, TIME,
//    43      .             FASTST, SPEED, FUEL, GAS * 100
//    44     6 FORMAT (X, '| ROOF     | TIME TO ROOF/LOW/LANDING |',
//    44      .           ' MAX SPEED   | FINAL SPEED | FUEL LEFT | GAS  |'/
//    44      .        X, '|', F6.0, ' M |',
//    44      .        X, 2(F6.1, '/'), F6.1, ' S   |',
//    44      .        X, F5.1, ' M/S   |', X, F5.1, ' M/S   |',
//    44      .        X, F6.0, ' KG |' ,
//    44      .        X, F3.0, '% |')
//
//       CONSIDER A HIGHER ALTITUDE IN THE NEXT ATTEMPT.
//    45     7 ROOF = ROOF + 25
//       CONTINUE WHILE 'ROOF' IS BELOW 'ORBIT' (QUITE LOGICAL, CAPTAIN).
//    46       IF (ROOF .LT. ORBIT) GO TO 1
//
//       C WE'RE DONE, NO ATTEMPTS LEFTS.
//    47       STOP
//    48       END
```