```
 1 def beam_input_generator(element_type, element_number,
   beam_size):
 2     """
 3     Beam model input file generator for FEMSS4 project
 4     Alasdair Gray, S1208454
 5     November 2016
 6     This function takes 3 input parameters:
 7     element type: either 'B21', 'B22' or 'B23'
 8     element number: The number of elements along the beam
 9     beam size: either 'thin' or 'thick'
10     The function then creates a subfolder in the directory it
   is being run from and writes an appropriate input file
11     with the same name to the subfolder, the function outputs
   the filename as a string.
12     """
13     # First, define the element properties based on the chosen
   element type
14     if element_type in ['B21', 'B23']:
15         node_per_el = 2
16         el_def = '1, 2'
17     else:
18         node_per_el = 3
19         el_def = '1, 2, 3'
20     # Define properties based on beam size
21     if beam_size == 'thick':
22         depth = 1.0
23         load = -100000
24     elif beam_size == 'thin':
25         depth = 0.2
26         load = -10000
27     # Calculate the number of nodes required along the beam
28     total_nodes = str(element_number*(node_per_el-1)+1)
29     import inspect, os
30     filename = element_type + '_' + beam_size + '_' + str(
   element_number) + 'EL'
31     Parent = os.path.dirname(os.path.realpath(__file__)) # Get
   the location the function is being run in
32     Child = Parent + '/' + filename # Create parameter specific
    subfolder name
33     if not os.path.exists(Child): # Create subfolder if it
   doesn't already exist
34         os.makedirs(Child)
35     os.chdir(Child)
36     output = open(filename+'.inp', 'w') # Define the inp file
   to write to
37     template = """
38     ************************************************************
   ********
39     ** FEMSS4 Project
```

```
39                                                                **
40      ** Alasdair Gray, S1208454
                                                  **
41      ** Beam model input file generated by my wonderful python
    script **
42      ************************************************************
    ********
43      *HEADING
44      5m LONG SIMPLY SUPPORTED %(beam_size)s BEAM WITH %(load)s
    kN/m DISTRIBUTED LOAD, %(element_number)s ELEMENT
45      %(element_type)s MESH
46      ******************************
47      *NODE
48      ** DEFINE LOCATION OF END NODES
49      1, 0.0, 0.0
50      %(total_nodes)s, 5.0, 0.0
51      ******************************
52      *NGEN, NSET=NALL
53      ** GENERATE NODES ALONG LENGTH AND DEFINE NODE SET 'NALL'
54      1, %(total_nodes)s, 1
55      ******************************
56      **DEFINE THE FIRST 'MASTER ELEMENT' NUMBER 1 AS A %(
    element_type)s
57      *ELEMENT, TYPE=%(element_type)s
58      1, %(el_def)s
59      ******************************
60      ** GENERATE THE REST OF THE ELEMENTS BY COPYING ELEMENT 1,
    MAKE %(element_number)s ELEMENTS, ELEMENT NUMBER
61      ** INCREMENT OF 1 AND NODE NUMBER INCREMENT OF %(node_shift
    )s, 1 ROW, ASSIGN TO ELSET 'EALL'
62      *ELGEN, ELSET=EALL
63      1,%(element_number)s,%(node_shift)s,1,1,1,1
64      ******************************
65      ** DEFINE MATERIAL CALLED 'STEEL'
66      *MATERIAL,NAME=STEEL
67      ******************************
68      ** DEFINE ELASTIC MATERIAL PROPERTIES, E = 200*10^9 AND
    POISSON'S RATIO 0
69      *ELASTIC
70      200E9, 0.3
71      ******************************
72      ** DEFINE THE BEAM SECTION, ASSIGN IT TO THE ELEMENT SET '
    EALL', ASSIGN IT WITH THE 'STEEL' MATERIAL AND DEFINE A
73      ** 0.2 M THICK BY (depth)s M DEEP BEAM
74      *BEAM SECTION, ELSET = EALL, MATERIAL = STEEL, SECTION =
    RECT
75      0.2, %(depth)s
76      ******************************
77      **ASK FOR HISTORY AND MODEL DATA TO BE INCLUDED IN RESULTS
```

```
 77 FILE
 78     *PREPRINT, ECHO=YES, MODEL=YES, HISTORY=YES
 79     ******************************
 80     ** BEGIN LINEAR STEP
 81     *STEP, PERTURBATION
 82     ******************************
 83     ** DEFINE STEP AS STATIC
 84     *STATIC
 85     ******************************
 86     ** DEFINE FIXED BOUNDARY AT NODE 1 LOCKED IN X AND Y
    DIRECTION AND AT NODE %(total_nodes)s LOCKED IN Y DIRECTION
 87     *BOUNDARY
 88     1, 1, 2
 89     %(total_nodes)s, 2, 2
 90     ******************************
 91     ** DEFINE DISTRIBUTED LOAD OF %(load)s KN/M IN Y DIRECTION
     ON ELSET 'EALL'
 92     *DLOAD
 93     EALL, PY, %(load)s
 94     ******************************
 95     ** ASK ABAQUS TO PRINT COORDINATE, STRESS AND STRAIN FOR
    EACH ELEMENT (AT INTEGRATION POINTS INSIDE CELLS)
 96     *EL PRINT
 97     COORD
 98     S
 99     E
100     ******************************
101     ** ASK ABAQUS TO PRINT STRESS VALUES AT NODES
102     *EL PRINT, POSITION=AVERAGED AT NODES
103     S
104     ******************************
105     ** ASK ABAQUS TO PRINT FORCES AND DISPLACEMENTS AT NODES
106     *NODE PRINT
107     U
108     RF
109     ******************************
110     ** END THE STEP
111     *END STEP """
112     # Now define the parameter based strings which will be
    inserted into the template string
113     context = {
114     "total_nodes":total_nodes,
115     "element_type":element_type,
116     "el_def":el_def,
117     "element_number": str(element_number),
118     "node_shift" : str(node_per_el-1),
119     "depth" : str(depth),
120     "load" : str(load),
121     "beam_size" : beam_size
```

```
122     }
123     # Fill the template blanks with the parameter based
    strings and write to the input file
124     output.write(template % context)
125     output.close() # Close the input file
126     return filename # Return the parameter specific filename
    to the caller
127
128 if __name__ == "__main__":
129     beam_input_generator('B22', 2, 'thin')
130
```