```
 1 # Features of my program:
 2 # For 13 user inputs, 23 output design parameters are
    displayed.
 3 # Also tells whether the given output is satisfactory
    considering wear and beam strenght.
 4 # Gives suggestions of various spur gears on the e-
   market which has precise module value got in the sum.
 5 def main():
 6     import math
 7     import matplotlib.pyplot as plt
 8     in_out = 0
 9     print("Enter 0 if value of the parameter is not
   known to you")
10     print("Feed me values!!")
11     print()
12     print("Enter any two of the following three:\n")
13     zp = int(input("Enter no of teeth of pinion (18
   to 40): \n"))
14     zg = int(input("Enter no of teeth of gear (18 to
   40): \n"))
15     i = int(input("velocity ratio/speed reduction(if
   known): \n"))
16     print()
17     print("Next values...")
18     kw = float(input("power(compulsory): \n"))
19     cs_op = int(input("is the service factor given?(1
   /0): "))
20     if cs_op==1:
21         Cs = float(input("Enter the service factor: "
   ))
22     else:
23         Cs = float(input("Enter the starting torque
   : "))
24         Cs = Cs/100
25     print()
26     m_op = int(input("is the module given?(1/0): "))
27     if m_op==1:
28         m = float(input("Enter module: "))
29     else:
30         pass
31     print()
```

```
32      b_op = int(input("Is face width given?(1/0): "))
33      if b_op==1:
34          b = float(input("Enter face width: "))
35      else:
36          pass
37      print("ANY ONE OF THE RPM IS REQUIRED FOR
    CALCULATION")
38      np = int(input("RPM of pinion(if known): "))
39      ng = int(input("RPM of gear(if known): "))
40      print()
41      fs_op = int(input("is factor of safety given?(1/0
    ): "))
42      if fs_op==1:
43          fs = float(input("factor of safety: "))
44      else:
45          pass
46      print()
47      mat = int(input("Are the material of the pinion
    and gear same(1) or different(0) : "))
48      if mat == 1:
49          Sut = float(input("ultimate tensile strenght
    : "))
50      else:
51          Sut_p = float(input("ultimate tensile
    strenght of pinion(if known): "))
52          Sut_g = float(input("ultimate tensile
    strenght of gear(if known): "))
53      print()
54      Ep = int(input("Modulus of Elasticity of pinion(
    if known): \n"))
55      Eg = int(input("Modulus of Elasticity of Gear(if
    known): \n"))
56      print(
    "**********************************************
    ******************************************")
57      print("Some More technical information required:
    \n")
58      print("What type of gear do you require?\n")
59      # purpose
60      type_p = int(input("(1.Ordinary or commercially
    available ; 2.Accuratly cut ; 3.Precision Gear) = \n"
```

```
60 ))
61     print()
62     print("What type of teeth your gear requires?\n"
   )
63     # for value of k (1: 0.107 ; 2: 0.111 ; 3:0.115
   ) and also for use of Y
64     type_k = int(input("(1. 14.5' Full Depth ; 2. 20
   ' Full Depth ; 3. 20' Stud Teeth) = \n"))
65     print(
66         "************************* Here Comes THE
   OUTPUT for the INFORMATION YOU PROVIDED
    ****************************")
67     rel_14F = {18: 0.270, 19: 0.277, 20: 0.283, 21:
   0.289, 22: 0.292, 23: 0.296, 24: 0.302, 25: 0.305,
   26: 0.308,
68                27: 0.311, 28: 0.314, 29: 0.316, 30:
   0.318, 32: 0.322, 33: 0.324, 35: 0.327, 37: 0.330,
   39: 0.335,
69                40: 0.336, 45: 0.340, 50: 0.346, 55:
   0.352, 60: 0.355, 65: 0.358, 70: 0.360, 75: 0.361,
   80: 0.363,
70                90: 0.366, 100: 0.368}
71     rel_20F = {18: 0.308, 19: 0.314, 20: 0.320, 21:
   0.326, 22: 0.330, 23: 0.333, 24: 0.337, 25: 0.340,
   26: 0.344,
72                27: 0.348, 28: 0.352, 29: 0.355, 30:
   0.358, 32: 0.364, 33: 0.367, 35: 0.373, 37: 0.380,
   39: 0.386,
73                40: 0.389, 45: 0.399, 50: 0.408, 55:
   0.415, 60: 0.421, 65: 0.425, 70: 0.429, 75: 0.433,
   80: 0.436,
74                90: 0.442, 100: 0.446}
75     rel_20S = {18: 0.377, 19: 0.386, 20: 0.393, 21:
   0.399, 22: 0.404, 23: 0.408, 24: 0.411, 25: 0.416,
   26: 0.421,
76                27: 0.426, 28: 0.430, 29: 0.434, 30:
   0.437, 32: 0.443, 33: 0.445, 35: 0.449, 37: 0.454,
   39: 0.457,
77                40: 0.459, 45: 0.468, 50: 0.474, 55:
   0.480, 60: 0.484, 65: 0.488, 70: 0.493, 75: 0.496,
   80: 0.499,
```

```
 78                 90: 0.503, 100: 0.506}
 79     b_m = 10
 80     rel_zp_zg = [32,33,35,37,39,40]
 81     if zp == 0:
 82         zp = zg / i
 83     if zg == 0:
 84         zg = i * zp
 85     if mat == 1:
 86         print("For Same Material of pinion and gear
    , The pinion is weaker. So we will work on design of
     pinion")
 87         if i != 0:
 88             i = i
 89         else:
 90             i = zg / zp
 91         print("The Velocity ratio is:{0}".format(i))
 92         if ng != 0 or np != 0:
 93             if np == 0:
 94                 np = ng * i
 95             else:
 96                 np = np
 97                 ng = np / i
 98         print("the RPM of pinion: {0}rpm".format(np
    ))
 99         print("the RPM of Gear: {0}rpm".format(ng))
100         Mt = (60 * (10 ** 6) * (kw * (10 ** 3))) / (
    2 * math.pi * np)
101         Mt = Mt * (10 ** -3)
102         print("Torque: {0}N-mm".format(Mt))
103         Q = (2 * zg) / (zg + zp)
104         print("the Ratio Factor: {0}".format(Q))
105         if type_p == 1:
106             v = 5
107             Cv = 0.375
108         elif type_p == 2:
109             v = 15
110             Cv = 0.285
111         elif type_p == 3:
112             v = 25
113             Cv = 0.528
114         else:
```

```
115             Cv = 0.375
116         Sut_3 = Sut / 3
117         print("Permissible Bending Stress: {0}N/mm^2
    ".format(Sut_3))
118         if type_k == 1:
119             Y = rel_14F[zp]
120         elif type_k == 3:
121             Y = rel_20S[zp]
122         else:
123             Y = rel_20F[zp]
124         print("Lewis factor (Y): {0}".format(Y))
125         if m_op==0:
126             m = (60 * math.pow(10, 6) / math.pi
    ) * ((kw * 1000 * Cs * fs) / (zp * np * Cv * (b_m
    ) * (Sut_3) * Y))
127             m = (m) ** 0.33
128             m = math.ceil(m * (math.pow(10, -1)))
129             print("The Calculated module is: {0}mm".
    format(m))
130         else:
131             print()
132         dp = m * zp
133         dg = m * zg
134         print("the Pitch Circle Diameters for pinion
     is {0}mm and for Gear is {1}mm".format(dp, dg))
135         if b_op==0:
136             b = 10 * m
137             print("the Face width: {0}mm".format(b))
138         else:
139             print()
140         Pt = (2 * Mt) / dp
141         print("the Tangential Load: {0}N".format(Pt
    ))
142         A_v = (math.pi * dp * np) / (60 * (10 ** 3))
143         print("the Velocity: {0}m/s".format(A_v))
144         if A_v < 10:
145             A_Cv = 3 / (3 + A_v)
146         elif A_v < 20:
147             A_Cv = 6 / (6 + A_v)
148         elif A_v > 20:
149             A_Cv = 5.6 / (5.6 + ((A_v) ** (1. / 2.
```

```
149 )))
150         else:
151             A_Cv = 3 / (3 + A_v)
152         print("the Velocity Factor: {0}".format(A_Cv
    ))
153         Peff = (Cs / A_Cv) * Pt
154         print("Effective load between 2 teeth: {0}N"
    .format(Peff))
155         Sb = m * b * Sut_3 * Y
156         print("Beam Strenght on Gear tooth: {0}N".
    format(Sb))
157         if fs_op==1:
158             fs = fs
159         else:
160             fs = Sb/Peff
161         # fs = A_fs
162         print("The required Factor of Safety is: {0
    }".format(fs))
163         a = (m * (zp + zg) / 2)
164         print("The Center distance: {0}mm".format(a
    ))
165         ha = m
166         hf = 1.25 * m
167         print("the addendum is {0} and the deddendum
     is {1}".format(ha, hf))
168         tt = 1.5708 * m
169         print("the tooth thickness: {0}mm".format(tt
    ))
170         c = 0.25 * m
171         print("the Clearance is: {0}mm".format(c))
172         if type_k == 1:
173             k = 0.107
174         elif type_k == 3:
175             k = 0.115
176         else:
177             k = 0.111
178         if Ep != 0 and Eg != 0:
179             C = k / ((1 / Ep) + (1 / Eg))
180             print("the Deformation Factor is: {0}".
    format(C))
181         Sw = Peff * fs
```

```
182         print("the Wear Strenght of the gear tooth
    : {0}N".format(Sw))
183         K = Pt / (b * Q * dp)
184         print("The Load-Stress Factor: {0}".format(K
    ))
185         # BHN = ((10000*Peff*fs)/(b*Q*dp*0.16))
186         # BHN = ((BHN)**0.5)
187         BHN = math.pow(((Peff*fs*math.pow(10,4))/(0.
    16*b*Q*dp)),0.5)
188         print("the Brinell Hardness Number: {0}".
    format(BHN))
189
190         # line 1 points
191         if zp>=18 and zp<=30:
192             x1 = [zp, zp + 2, zp + 4]
193             y1 = [rel_14F[zp], rel_14F[zp + 2],
    rel_14F[zp + 4]]
194             # plotting the line 1 points
195             plt.plot(x1, y1, label="For 14.5' Full
    Depth")
196
197             # line 2 points
198             x2 = [zp, zp + 2, zp + 4]
199             y2 = [rel_20F[zp], rel_20F[zp + 2],
    rel_20F[zp + 4]]
200             # plotting the line 2 points
201             plt.plot(x2, y2, label="For 20' Full
    Depth")
202
203             x3 = [zp, zp + 2, zp + 4]
204             y3 = [rel_20S[zp], rel_20S[zp + 2],
    rel_20S[zp + 4]]
205             # plotting the line 2 points
206             plt.plot(x3, y3, label="For 20' Stub
    Depth")
207
208             # naming the x axis
209             plt.xlabel('No of teeth of pinion')
210             # naming the y axis
211             plt.ylabel('Form factor')
212             # giving a title to my graph
```

```
213             plt.title('Zp v/s Y for different depths
    ')
214
215             # show a legend on the plot
216             plt.legend()
217
218             # function to show the plot
219             plt.show()
220         elif zp in rel_zp_zg:
221             x1 = [zp, zp + 2, zp + 4]
222             y1 = [rel_14F[zp], rel_14F[zp + 2],
    rel_14F[zp + 4]]
223             # plotting the line 1 points
224             plt.plot(x1, y1, label="For 14.5' Full
    Depth")
225
226             # line 2 points
227             x2 = [zp, zp + 2, zp + 4]
228             y2 = [rel_20F[zp], rel_20F[zp + 2],
    rel_20F[zp + 4]]
229             # plotting the line 2 points
230             plt.plot(x2, y2, label="For 20' Full
    Depth")
231
232             x3 = [zp, zp + 2, zp + 4]
233             y3 = [rel_20S[zp], rel_20S[zp + 2],
    rel_20S[zp + 4]]
234             # plotting the line 2 points
235             plt.plot(x3, y3, label="For 20' Stub
    Depth")
236
237             # naming the x axis
238             plt.xlabel('No of teeth of pinion')
239             # naming the y axis
240             plt.ylabel('Form factor')
241             # giving a title to my graph
242             plt.title('Zp v/s Y for different depths
    ')
243
244             # show a legend on the plot
245             plt.legend()
```

```
246
247             # function to show the plot
248             plt.show()
249         elif zp>40 and zp<=100:
250             x1 = [zp, zp + 5, zp + 10]
251             y1 = [rel_14F[zp], rel_14F[zp + 5],
    rel_14F[zp + 10]]
252             # plotting the line 1 points
253             plt.plot(x1, y1, label="For 14.5' Full
    Depth")
254
255             # line 2 points
256             x2 = [zp, zp + 5, zp + 10]
257             y2 = [rel_20F[zp], rel_20F[zp + 5],
    rel_20F[zp + 10]]
258             # plotting the line 2 points
259             plt.plot(x2, y2, label="For 20' Full
    Depth")
260
261             x3 = [zp, zp + 5, zp + 10]
262             y3 = [rel_20S[zp], rel_20S[zp + 5],
    rel_20S[zp + 10]]
263             # plotting the line 2 points
264             plt.plot(x3, y3, label="For 20' Stub
    Depth")
265             # naming the x axis
266             plt.xlabel('No of teeth of pinion')
267             # naming the y axis
268             plt.ylabel('Form factor')
269             # giving a title to my graph
270             plt.title('Zp v/s Y for different depths
    ')
271
272             # show a legend on the plot
273             plt.legend()
274
275             # function to show the plot
276             plt.show()
277         else:
278             print()
279
```

```
280         if m>0.5 and m<=4:
281             print("Check out this link for Spur gear
     of the correct module range: https://www.indiamart.
    com/proddetail/ground-spur-gears-15026091597.html?
    pos=2&pla=n")
282         elif m>5 and m<=10:
283             print("Check out this link for Spur gear
     of the correct module range: https://www.indiamart.
    com/proddetail/textile-spur-gears-23886979612.html?
    pos=5&kwd=spur%20gear%20module&tags=A||||8249.014|
    Price|product")
284         elif m>4 and m<=18:
285             print("Check out this link for Spur gear
     of the correct module range: https://www.indiamart.
    com/proddetail/spur-gears-4117283991.html?pos=3&kwd=
    spur%20gear%20module&tags=A||||8132.985||product")
286         if Sb > Peff and Sw > Peff:
287             print("The output displayed is
    satisfactory to the input given as Sb > Peff and Sw
     > Peff")
288             print()
289             print(
    "**************************************************
    **************************************************
    ********************")
290             print()
291             main()
292
293         elif Sb > Peff or Sw > Peff:
294             if Sb > Peff:
295                 print("The Design is satisfactry in
    terms of beam strenght")
296             else:
297                 print("The Design is satisfactory in
     terms of wear strenght")
298             print()
299             print(
300
    "**************************************************
    **************************************************
    ********************")
```

```
301             print()
302             main()
303 
304         else:
305             print("The output displayed is not
    satisfactory to the input given as Sb < Peff and Sw
     < Peff")
306             print()
307             print(
308 
    "**************************************************
    **************************************************
    ********************")
309             print()
310             main()
311 
312     else:
313         if type_k == 1:
314             p1 = (Sut_p / 3) * rel_14F[zp]
315             g1 = (Sut_g / 3) * rel_14F[zg]
316         elif type_k == 3:
317             p1 = (Sut_p / 3) * rel_20S[zp]
318             g1 = (Sut_g / 3) * rel_20S[zg]
319         else:
320             p1 = (Sut_p / 3) * rel_20F[zp]
321             g1 = (Sut_g / 3) * rel_20F[zg]
322         if p1 < g1:
323             print(
324                 "In different Material, the sigma*Y
    for pinion {0} is smaller than sigma*Y for Gears {1
    }. So we will focus on Design of Pinion".format(
325                     p1, g1))
326             if i != 0:
327                 i = i
328             else:
329                 i = zg / zp
330             print("The Velocity ratio is:{0}".format
    (i))
331             if ng != 0 or np != 0:
332                 if np == 0:
333                     np = ng * i
```

```
334                 else:
335                     np = np
336             print("the RPM of pinion: {0}rpm".format
    (np))
337             Mt = (60 * (10 ** 6) * (kw * (10 ** 3
    ))) / (2 * math.pi * np)
338             Mt = Mt * (10 ** -3)
339             print("Torque: {0}N-mm".format(Mt))
340             Q = (2 * zg) / (zg + zp)
341             print("the Ratio Factor: {0}".format(Q))
342             if type_p == 1:
343                 v = 5
344                 Cv = 0.375
345             elif type_p == 2:
346                 v = 15
347                 Cv = 0.285
348             elif type_p == 3:
349                 v = 25
350                 Cv = 0.528
351             else:
352                 Cv = 0.375
353             Sut_3 = Sut_p / 3
354             print("Permissible Bending Stress: {0}N/
    mm^2".format(Sut_3))
355             if type_k == 1:
356                 Y = rel_14F[zp]
357             elif type_k == 3:
358                 Y = rel_20S[zp]
359             else:
360                 Y = rel_20F[zp]
361             print("Lewis factor(Y): {0}".format(Y))
362             if m_op == 0:
363                 m = (60 * math.pow(10, 6) / math.pi
    ) * ((kw * 1000 * Cs * fs) / (zp * np * Cv * (b_m
    ) * (Sut_3) * Y))
364                 m = (m) ** 0.33
365                 m = math.ceil(m * (math.pow(10, -1
    )))
366                 print("The Calculated module is: {0}
    mm".format(m))
367             else:
```

```
368                 print()
369             dp = m * zp
370             dg = m * zg
371             print("the Pitch Circle Diameters for
    pinion is {0}mm and for Gear is {1}mm".format(dp, dg
    ))
372             if b_op == 0:
373                 b = 10 * m
374                 print("the Face width: {0}mm".format
    (b))
375             else:
376                 print()
377             Pt = (2 * Mt) / dp
378             print("the Tangential Load: {0}N".format
    (Pt))
379             A_v = (math.pi * dp * np) / (60 * (10
     ** 3))
380             print("the Velocity: {0}m/s".format(A_v
    ))
381             if A_v < 10:
382                 A_Cv = 3 / (3 + A_v)
383             elif A_v < 20:
384                 A_Cv = 6 / (6 + A_v)
385             elif A_v > 20:
386                 A_Cv = 5.6 / (5.6 + ((A_v) ** (1. /
    2.)))
387             else:
388                 A_Cv = 3 / (3 + A_v)
389             print("the Velocity Factor: {0}".format(
    A_Cv))
390             Peff = (Cs / A_Cv) * Pt
391             print("Effective load between 2 teeth: {
    0}N".format(Peff))
392             Sb = m * b * Sut_3 * Y
393             print("Beam Strenght on Gear tooth: {0}N
    ".format(Sb))
394             if fs_op == 1:
395                 fs = fs
396             else:
397                 fs = Sb / Peff
398             # fs = A_fs
```

```
399             print("The required Factor of Safety is
    : {0}".format(fs))
400             a = (m * (zp + zg) / 2)
401             print("The Center distance: {0}mm".
    format(a))
402             ha = m
403             hf = 1.25 * m
404             print("the addendum is {0} and the
    deddendum is {1}".format(ha, hf))
405             tt = 1.5708 * m
406             print("the tooth thickness: {0}mm".
    format(tt))
407             c = 0.25 * m
408             print("the Clearance is: {0}mm".format(c
    ))
409             if type_k == 1:
410                 k = 0.107
411             elif type_k == 3:
412                 k = 0.115
413             else:
414                 k = 0.111
415             if Ep != 0 and Eg != 0:
416                 C = k / ((1 / Ep) + (1 / Eg))
417                 print("the Deformation Factor is: {0
    }".format(C))
418             Sw = Peff * fs
419             print("the Wear Strenght of the gear
    tooth: {0}N".format(Sw))
420             K = Pt / (b * Q * dp)
421             print("The Load-Stress Factor: {0}".
    format(K))
422             BHN = ((10000 * Peff * fs) / (b * Q * dp
     * 0.16))
423             BHN = 2 * ((BHN) ** 0.5)
424             print("the Brinell Hardness Number: {0}"
    .format(BHN))
425             # line 1 points
426             if zp >= 18 and zp <= 30:
427                 x1 = [zp, zp + 2, zp + 4]
428                 y1 = [rel_14F[zp], rel_14F[zp + 2],
    rel_14F[zp + 4]]
```

```
429                 # plotting the line 1 points
430                 plt.plot(x1, y1, label="For 14.5'
    Full Depth")
431
432                 # line 2 points
433                 x2 = [zp, zp + 2, zp + 4]
434                 y2 = [rel_20F[zp], rel_20F[zp + 2],
    rel_20F[zp + 4]]
435                 # plotting the line 2 points
436                 plt.plot(x2, y2, label="For 20' Full
     Depth")
437
438                 x3 = [zp, zp + 2, zp + 4]
439                 y3 = [rel_20S[zp], rel_20S[zp + 2],
    rel_20S[zp + 4]]
440                 # plotting the line 2 points
441                 plt.plot(x3, y3, label="For 20' Stub
     Depth")
442
443                 # naming the x axis
444                 plt.xlabel('No of teeth of pinion')
445                 # naming the y axis
446                 plt.ylabel('Form factor')
447                 # giving a title to my graph
448                 plt.title('Zp v/s Y for different
    depths')
449
450                 # show a legend on the plot
451                 plt.legend()
452
453                 # function to show the plot
454                 plt.show()
455             elif zp in rel_zp_zg:
456                 x1 = [zp, zp + 2, zp + 4]
457                 y1 = [rel_14F[zp], rel_14F[zp + 2],
    rel_14F[zp + 4]]
458                 # plotting the line 1 points
459                 plt.plot(x1, y1, label="For 14.5'
    Full Depth")
460
461                 # line 2 points
```

```
462                 x2 = [zp, zp + 2, zp + 4]
463                 y2 = [rel_20F[zp], rel_20F[zp + 2],
    rel_20F[zp + 4]]
464                 # plotting the line 2 points
465                 plt.plot(x2, y2, label="For 20' Full
     Depth")
466
467                 x3 = [zp, zp + 2, zp + 4]
468                 y3 = [rel_20S[zp], rel_20S[zp + 2],
    rel_20S[zp + 4]]
469                 # plotting the line 2 points
470                 plt.plot(x3, y3, label="For 20' Stub
     Depth")
471
472                 # naming the x axis
473                 plt.xlabel('No of teeth of pinion')
474                 # naming the y axis
475                 plt.ylabel('Form factor')
476                 # giving a title to my graph
477                 plt.title('Zp v/s Y for different
    depths')
478
479                 # show a legend on the plot
480                 plt.legend()
481
482                 # function to show the plot
483                 plt.show()
484             elif zp > 40 and zp <= 100:
485                 x1 = [zp, zp + 5, zp + 10]
486                 y1 = [rel_14F[zp], rel_14F[zp + 5],
    rel_14F[zp + 10]]
487                 # plotting the line 1 points
488                 plt.plot(x1, y1, label="For 14.5'
    Full Depth")
489
490                 # line 2 points
491                 x2 = [zp, zp + 5, zp + 10]
492                 y2 = [rel_20F[zp], rel_20F[zp + 5],
    rel_20F[zp + 10]]
493                 # plotting the line 2 points
494                 plt.plot(x2, y2, label="For 20' Full
```

```
494  Depth")
495
496                 x3 = [zp, zp + 5, zp + 10]
497                 y3 = [rel_20S[zp], rel_20S[zp + 5],
    rel_20S[zp + 10]]
498                 # plotting the line 2 points
499                 plt.plot(x3, y3, label="For 20' Stub
     Depth")
500                 # naming the x axis
501                 plt.xlabel('No of teeth of pinion')
502                 # naming the y axis
503                 plt.ylabel('Form factor')
504                 # giving a title to my graph
505                 plt.title('Zp v/s Y for different
    depths')
506
507                 # show a legend on the plot
508                 plt.legend()
509
510                 # function to show the plot
511                 plt.show()
512             else:
513                 print()
514             if m > 0.5 and m <= 4:
515                 print(
516                     "Check out this link for Spur
    gear of the correct module range: https://www.
    indiamart.com/proddetail/ground-spur-gears-
    15026091597.html?pos=2&pla=n")
517             elif m > 5 and m <= 10:
518                 print(
519                     "Check out this link for Spur
    gear of the correct module range: https://www.
    indiamart.com/proddetail/textile-spur-gears-
    23886979612.html?pos=5&kwd=spur%20gear%20module&tags
    =A||||8249.014|Price|product")
520             elif m > 4 and m <= 18:
521                 print(
522                     "Check out this link for Spur
    gear of the correct module range: https://www.
    indiamart.com/proddetail/spur-gears-4117283991.html?
```

```
pos=3&kwd=spur%20gear%20module&tags=A||||8132.985||product")
            if Sb > Peff and Sw > Peff:
                print("The output displayed is satisfactory to the input given as Sb > Peff and Sw > Peff")
                print()
                print(
                "**************************************************
                **************************************************
                ********************")
                print()
                main()

            elif Sb > Peff or Sw > Peff:
                if Sb > Peff:
                    print("The Design is satisfactry in terms of beam strenght")
                else:
                    print("The Design is satisfactory in terms of wear strenght")
                print()
                print(
                "**************************************************
                **************************************************
                ********************")
                print()
                main()

            else:
                print("The output displayed is not satisfactory to the input given as Sb < Peff and Sw < Peff")
                print()
                print(
                "**************************************************
                **************************************************
                ********************")
```

```
547                 print()
548                 main()
549         else:
550             print(
551                 "In different Material, the sigma*Y
    for Gear {0} is smaller than sigma*Y for Pinion {1
    }. So we will focus on Design of Gear".format(
552                     g1, p1))
553             if i != 0:
554                 i = i
555             else:
556                 i = zg / zp
557             print("The Velocity ratio is:{0}".format
    (i))
558             if ng != 0 or np != 0:
559                 if ng == 0:
560                     ng = np / i
561                 else:
562                     ng = ng
563             print("the RPM of Gear: {0}rpm".format(
    ng))
564             Mt = (60 * (10 ** 6) * (kw * (10 ** 3
    ))) / (2 * math.pi * ng)
565             Mt = Mt * (10 ** -3)
566             print("Torque: {0}N-mm".format(Mt))
567             Q = (2 * zg) / (zg + zp)
568             print("the Ratio Factor: {0}".format(Q))
569             if type_p == 1:
570                 v = 5
571                 Cv = 0.375
572             elif type_p == 2:
573                 v = 15
574                 Cv = 0.285
575             elif type_p == 3:
576                 v = 25
577                 Cv = 0.528
578             else:
579                 Cv = 0.375
580             Sut_3 = Sut_g / 3
581             print("Permissible Bending Stress: {0}N/
    mm^2".format(Sut_3))
```

```
582             if type_k == 1:
583                 Y = rel_14F[zg]
584             elif type_k == 3:
585                 Y = rel_20S[zg]
586             else:
587                 Y = rel_20F[zg]
588             print("Lewis factor(Y): {0}".format(Y))
589             if m_op == 0:
590                 m = (60 * math.pow(10, 6) / math.pi
    ) * ((kw * 1000 * Cs * fs) / (zg * ng * Cv * (b_m
    ) * (Sut_3) * Y))
591                 m = (m) ** 0.33
592                 m = math.ceil(m * (math.pow(10, -1
    )))
593                 print("The Calculated module is: {0}
    mm".format(m))
594             else:
595                 print()
596             dp = m * zp
597             dg = m * zg
598             print("the Pitch Circle Diameters for
    pinion is {0}mm and for Gear is {1}mm".format(dp, dg
    ))
599             if b_op == 0:
600                 b = 10 * m
601                 print("the Face width: {0}mm".format
    (b))
602             else:
603                 print()
604             Pt = (2 * Mt) / dg
605             print("the Tangential Load: {0}N".format
    (Pt))
606             A_v = (math.pi * dp * np) / (60 * (10
     ** 3))
607             print("the Velocity: {0}m/s".format(A_v
    ))
608             if A_v < 10:
609                 A_Cv = 3 / (3 + A_v)
610             elif A_v < 20:
611                 A_Cv = 6 / (6 + A_v)
612             elif A_v > 20:
```

```
613                 A_Cv = 5.6 / (5.6 + ((A_v) ** (1. /
    2.)))
614             else:
615                 A_Cv = 3 / (3 + A_v)
616             print("the Velocity Factor: {0}".format(
    A_Cv))
617             Peff = (Cs / A_Cv) * Pt
618             print("Effective load between 2 teeth: {
    0}N".format(Peff))
619             Sb = m * b * Sut_3 * Y
620             print("Beam Strenght on Gear tooth: {0}N
    ".format(Sb))
621             if fs_op == 1:
622                 fs = fs
623             else:
624                 fs = Sb / Peff
625             # fs = A_fs
626             print("The required Factor of Safety is
    : {0}".format(fs))
627             a = (m * (zp + zg) / 2)
628             print("The Center distance: {0}mm".
    format(a))
629             ha = m
630             hf = 1.25 * m
631             print("the addendum is {0} and the
    deddendum is {1}".format(ha, hf))
632             tt = 1.5708 * m
633             print("the tooth thickness: {0}mm".
    format(tt))
634             c = 0.25 * m
635             print("the Clearance is: {0}mm".format(c
    ))
636             if type_k == 1:
637                 k = 0.107
638             elif type_k == 3:
639                 k = 0.115
640             else:
641                 k = 0.111
642             if Ep != 0 and Eg != 0:
643                 C = k / ((1 / Ep) + (1 / Eg))
644                 print("the Deformation Factor is: {0
```

```
644 }".format(C))
645             Sw = Peff*fs
646             print("the Wear Strenght of the gear
    tooth: {0}N".format(Sw))
647             K = Pt / (b * Q * dp)
648             print("The Load-Stress Factor: {0}".
    format(K))
649             BHN = ((10000 * Peff * fs) / (b * Q * dp
     * 0.16))
650             BHN = 2 * ((BHN) ** 0.5)
651             print("the Brinell Hardness Number: {0}"
    .format(BHN))
652             # line 1 points
653             if zg >= 18 and zg <= 30:
654                 x1 = [zg, zg + 2, zg + 4]
655                 y1 = [rel_14F[zg], rel_14F[zg + 2],
    rel_14F[zg + 4]]
656                 # plotting the line 1 points
657                 plt.plot(x1, y1, label="For 14.5'
    Full Depth")
658
659                 # line 2 points
660                 x2 = [zg, zg + 2, zg + 4]
661                 y2 = [rel_20F[zg], rel_20F[zg + 2],
    rel_20F[zg + 4]]
662                 # plotting the line 2 points
663                 plt.plot(x2, y2, label="For 20' Full
     Depth")
664
665                 x3 = [zg, zg + 2, zg + 4]
666                 y3 = [rel_20S[zg], rel_20S[zg + 2],
    rel_20S[zg + 4]]
667                 # plotting the line 2 points
668                 plt.plot(x3, y3, label="For 20' Stub
     Depth")
669
670                 # naming the x axis
671                 plt.xlabel('No of teeth of pinion')
672                 # naming the y axis
673                 plt.ylabel('Form factor')
674                 # giving a title to my graph
```

```
675                 plt.title('Zg v/s Y for different
    depths')
676
677                 # show a legend on the plot
678                 plt.legend()
679
680                 # function to show the plot
681                 plt.show()
682             elif zg in rel_zp_zg:
683                 x1 = [zg, zg + 2, zg + 4]
684                 y1 = [rel_14F[zg], rel_14F[zg + 2],
    rel_14F[zg + 4]]
685                 # plotting the line 1 points
686                 plt.plot(x1, y1, label="For 14.5'
    Full Depth")
687
688                 # line 2 points
689                 x2 = [zg, zg + 2, zg + 4]
690                 y2 = [rel_20F[zg], rel_20F[zg + 2],
    rel_20F[zg + 4]]
691                 # plotting the line 2 points
692                 plt.plot(x2, y2, label="For 20' Full
     Depth")
693
694                 x3 = [zg, zg + 2, zg + 4]
695                 y3 = [rel_20S[zg], rel_20S[zg + 2],
    rel_20S[zg + 4]]
696                 # plotting the line 2 points
697                 plt.plot(x3, y3, label="For 20' Stub
     Depth")
698
699                 # naming the x axis
700                 plt.xlabel('No of teeth of pinion')
701                 # naming the y axis
702                 plt.ylabel('Form factor')
703                 # giving a title to my graph
704                 plt.title('Zg v/s Y for different
    depths')
705
706                 # show a legend on the plot
707                 plt.legend()
```

```
708
709                 # function to show the plot
710                 plt.show()
711             elif zg > 40 and zg <= 100:
712                 x1 = [zg, zg + 5, zg + 10]
713                 y1 = [rel_14F[zg], rel_14F[zg + 5],
    rel_14F[zg + 10]]
714                 # plotting the line 1 points
715                 plt.plot(x1, y1, label="For 14.5'
    Full Depth")
716
717                 # line 2 points
718                 x2 = [zg, zg + 5, zg + 10]
719                 y2 = [rel_20F[zg], rel_20F[zg + 5],
    rel_20F[zg + 10]]
720                 # plotting the line 2 points
721                 plt.plot(x2, y2, label="For 20' Full
     Depth")
722
723                 x3 = [zg, zg + 5, zg + 10]
724                 y3 = [rel_20S[zg], rel_20S[zg + 5],
    rel_20S[zg + 10]]
725                 # plotting the line 2 points
726                 plt.plot(x3, y3, label="For 20' Stub
     Depth")
727                 # naming the x axis
728                 plt.xlabel('No of teeth of pinion')
729                 # naming the y axis
730                 plt.ylabel('Form factor')
731                 # giving a title to my graph
732                 plt.title('Zg v/s Y for different
    depths')
733
734                 # show a legend on the plot
735                 plt.legend()
736
737                 # function to show the plot
738                 plt.show()
739             else:
740                 print()
741             if m > 0.5 and m <= 4:
```

```
742                 print(
743                     "Check out this link for Spur
    gear of the correct module range: https://www.
    indiamart.com/proddetail/ground-spur-gears-
    15026091597.html?pos=2&pla=n")
744             elif m > 5 and m <= 10:
745                 print(
746                     "Check out this link for Spur
    gear of the correct module range: https://www.
    indiamart.com/proddetail/textile-spur-gears-
    23886979612.html?pos=5&kwd=spur%20gear%20module&tags
    =A||||8249.014|Price|product")
747             elif m > 4 and m <= 18:
748                 print(
749                     "Check out this link for Spur
    gear of the correct module range: https://www.
    indiamart.com/proddetail/spur-gears-4117283991.html?
    pos=3&kwd=spur%20gear%20module&tags=A||||8132.985||
    product")
750             if Sb > Peff and Sw > Peff:
751                 print("The output displayed is
    satisfactory to the input given as Sb > Peff and Sw
     > Peff")
752                 print()
753                 print(
754
    "**************************************************
    **************************************************
    ********************")
755                 print()
756                 main()
757
758             elif Sb > Peff or Sw > Peff:
759                 if Sb > Peff:
760                     print("The Design is satisfactry
     in terms of beam strenght")
761                 else:
762                     print("The Design is
    satisfactory in terms of wear strenght")
763                 print()
764                 print(
```

```
765
    "**************************************************
    **************************************************
    ********************")
766                 print()
767                 main()
768
769             else:
770                 print("The output displayed is not
    satisfactory to the input given as Sb < Peff and Sw
     < Peff")
771                 print()
772                 print(
773
    "**************************************************
    **************************************************
    ********************")
774                 print()
775                 main()
776 main()
777
778
779
780
```