```
#Import math module
import math

#Define main function
def main():

    #Defined Variables
    team_number = 26
    body_weight = 1015.34
    outer_dia = 35.0
    canal_diameter = 19.00
    femoral_head_offset = 51.0
    modulus_bone = 20.0 #(17-20 GPa) citationC;, S. E. R. (n.d.). Biomechanics of immature human cortical bone: A systematic review. Journal of the mechanical behavior of biomedical materials. Retrieved December 5, 2022, from https://pubmed.ncbi.nlm.nih.gov/34736022/
    ult_ten_strength = 1280.0 #(780-1280 MPa) material: Cobalt-Chromium_molybdenum alloy
    modulus_implant= 250.0 #(210 to 250 GPa) material: Cobalt-Chromium_molybdenum alloy
    stem_dia= 0.85 * canal_diameter


    #Subprogram One
    def min_stem_dia():
       min_stem_dia=0
       canal_d_range=[]

       for i in range (1,1900):
          canal_d_range.append(float(i)/100.0) #input all possible diamters up to canal diameter

       for d in canal_d_range: #run through all the diameters to see which produces a tensile stress that equals the ultimate tensile stress
          force = 3.25 * body_weight
          area =(math.pi/4) * ((d)**2)
          moment = force * femoral_head_offset
          y= 0.5 * d
          i = (math.pi/64) * (d**4)
          axial_stress = force/area
          bending_stress = (moment * y)/i
          app_ten_stress = int(axial_stress + bending_stress)

          if app_ten_stress == ult_ten_strength: #if the applied stress equals ultimate strength, dia is at its minimum
             min_stem_dia = d
```

```
        print('Body weight:\t\t\t', round(body_weight,1),'N\nCanal diameter:\t\t\t',
round(canal_diameter,1),'mm\nUltimate tensile strength:\t', round(ult_ten_strength,1),
'MPa\nMininum stem diameter:\t\t', round(min_stem_dia,1),'mm\nApplied tensile stress:\t\t',
format(app_ten_stress,'.1f'),'N/mm^2\n')



    #Subprogram Two
    def convert_to_int(dataset): #converts data to integers
      data = []
      for row in dataset:
        row_list = []
      for col in row:
        value = float(col)
        row_list.append(value)
      data.append(row_list)
      return data

    def failed_cycles_adj_stress():
      force=10*body_weight
      area=(math.pi/4)*(stem_dia**2)

      max_stress = force/area
      min_stress = -force/area

      amplitude_stress = (max_stress - min_stress)/2

      file = open('SN Data - Sample Metal.txt','r')
      failed_cycles = []
      for line in file:
        cycles = line.split()
        failed_cycles.append(cycles)

      failed_cycles = convert_to_int(failed_cycles)

      for i in range(len(failed_cycles)):
        kn = 9.25 + (math.log(failed_cycles[i][1],10)**((0.65*team_number)/40))
        adjusted_stress_amplitude = kn * amplitude_stress

        if adjusted_stress_amplitude > failed_cycles[i][0]: #if the adjusted stess amplitude is
greater than the stress amplitude S, it corresponds to failure
          print('Number of cycles before failure:',failed_cycles[i][1], '\nAdjusted stress
amplitude:',adjusted_stress_amplitude, 'N/mm^2\n')
```

```
    #Subprogam Three
    def years_before_risk():
        area = (math.pi/4) * ((outer_dia - canal_diameter) ** 2)
        force = 28 * body_weight
        comp_strength = force/area
        stress_reduc = comp_strength * (((4 * modulus_bone)/(modulus_bone +
modulus_implant)) ** (1/3))
        E_ratio = math.sqrt(modulus_implant/modulus_bone)

        for i in range(0,20): #hip implants usualy last for 10-20 years, i represents the years from
post-surgery to 20 years
            stress_fail = ((0.0012 * (i ** 2))-(3.725 * i * E_ratio) + 186.42)
            if stress_fail>stress_reduc:
                print('Stress failure:',stress_fail, 'N/mm^2')
                print('Number of years before femoral fracture:', i, '\n')


    #Home menu
    while True:
        user_input = input("Home menu to test hip implant materials. Please type one of the
following letters to proceed: \na) Minimum stem diameter\nb) Number of cycles before implant
failure and its adjusted stress amplitude\nc) Years post-surgery before implant fracture\nd) Exit
program\n")
        if user_input.upper() == 'A':
            min_stem_dia()
        elif user_input.upper() == 'B':
            failed_cycles_adj_stress()
        elif user_input.upper() == 'C':
            years_before_risk()
        elif user_input.upper() == 'D':
            print('\nEnd program.')
            break
        else:
            print("\nPlease select an option from the menu.\n")
```