MECHANICAL SHAFT DESIGN AND DRAWING PROGRAM

COMPONENTS**: Shaft, Key**

PROGRAMMING LANGUAGE: **Python**

CODE:

```
import numpy as np
import math
from tkinter import *

print("In this program, you can find out the diameter of the shaft for required load conditions")

power, speed, Torque = 0, 0, 0
length_of_shaft = float(input("Enter the length of the shaft in mm:"))
fos = float(input("Enter factor of safety for the shaft:"))
print("Enter various shaft loads")
Max_BM = float(input("Enter the maximum bending moment in N.mm:"))

Choice_cal_torque \
    = input("Enter true to calculate the torque using power and speed or false to directly enter the amount of torque:")
if Choice_cal_torque == "true":
    power,speed = input("Enter power to be transmitted by shaft in Watt and corresponding speed in rpm:").split()
    Torque = float((int(power) * 60000) / (2 * np.pi * int(speed)))

else:
    Torque = float(input("Enter the amount torque on shaft:"))
print("Torque in the shaft is:"+str(Torque)+"N.mm")
print("Enter the shaft material properties")
ultimate_tensile_stress=int(input("Enter ultimate tensile stress of the material in Mpa:"))
ultimate_shear_stress=int(input("Enter ultimate shear stress of the material in Mpa:"))
allowable_shear_stress=(ultimate_shear_stress/fos)
allowable_tensile_stress=(ultimate_tensile_stress/fos)
```

```
val=str(input("To get values for Kb and Kt, press GL for gradual loading, MSL for minor shock load and HSL for heavy shock load:"))

Kb_value={"GL":1.5,"MSL":1.75,"HSL":2.5}

Kt_value={"GL":1,"MSL":1.25,"HSL":2.25}

Kb=float(Kb_value.get(val))

Kt=float(Kt_value.get(val))


equivalent_twisting_moment = math.sqrt(((Kb*Max_BM)**2)+((Kt*Torque)**2))

equivalent_bending_moment = 0.5 * (Kb*Max_BM +
math.sqrt(((Kb*Max_BM)**2)+((Kt*Torque)**2)))

print("Equivalent bending moment is"+str(equivalent_bending_moment)+"N.mm"+"\nEquivalent twisting moment is"+str(equivalent_twisting_moment)+"N.mm")


dt_cube=(16*equivalent_twisting_moment)/(3.14*allowable_shear_stress)

db_cube=(32*equivalent_bending_moment)/(3.14*allowable_tensile_stress)

if dt_cube<db_cube:

    db=int(math.pow(db_cube,(1/3)))

    print("Diameter of the shaft for bending is:" + str(db) + "mm")

    diameter=db

else:

    dt=int(math.pow(dt_cube,(1/3)))

    print("Diameter of the shaft for twisting is:"+str(dt)+"mm")

    diameter=dt

R_20_series=[10,11.2,12.5,14,16,18,20,22.4,25,28,31.5,35.5,40,45,50,56,63,71,80,90,100]

for i in range(len(R_20_series)):

    diameter_of_shaft=R_20_series[i]

    if diameter <= diameter_of_shaft:

        print("Diameter of the shaft rounded off to R20 series is:"+str(diameter_of_shaft)+"mm")

        break


# calculation to find out dimensions of Key


print("To find out dimensions of rectangular sunk key")
```

```
#key_width,key_thickness= input("Enter the width and height of the key from data book for corresponding shaft diameter in mm:").split()

key_series=[6,8,10,12,17,22,30,38,44,50,58,65,75,85,95,110,130,150,170,200,230,260,290,330,380,440]

for k in range(len(key_series)):

    if diameter_of_shaft<=key_series[k]:

        choice=k

        break

key_width_list=[2,3,4,5,6,8,10,12,14,16,18,20,22,25,28,32,36,40,45,50,56,63,70,80,90,100]

key_thickness_list=[2,3,4,5,6,7,8,8,9,10,11,12,14,14,16,18,20,22,25,28,32,32,36,40,45,50]

key_width=key_width_list[k]

key_thickness=key_thickness_list[k]

print("Key_width is:"+str(key_width)+"mm"+ " and key_thickness is:"+str(key_thickness)+"mm")


print("Enter the key material properties")

key_crushing_stress=int(input("Enter crushing stress of the material in Mpa:"))

key_shear_stress=int(input("Enter ultimate shear stress of the material in Mpa:"))


key_fos = float(input("Enter factor of safety for the shaft:"))

key_allowable_shear_stress=(key_shear_stress/key_fos)

key_allowable_crushing_stress=(key_crushing_stress/key_fos)


key_torsional_shear_strength=(np.pi*key_allowable_shear_stress*(diameter_of_shaft**3))/(16)

key_length_shear=(key_torsional_shear_strength*2)/(int(key_width)*key_allowable_shear_stress*diameter_of_shaft)

key_length_crush=(key_torsional_shear_strength*4)/(int(key_thickness)*key_allowable_crushing_stress*diameter_of_shaft)


if key_length_shear < key_length_crush:

    print("Length of the key w.r.t shear stress is:"+str(key_length_crush)+"mm")

    length=key_length_crush

else:

    print("Length of the key w.r.t crushing stress is:" + str(key_length_shear) + "mm")

    length = key_length_shear
```

```
key_series=[6,8,10,12,16,18,20,22,25,28,32,36,40,45,50,56,63,70,80,90,100,110,125,140,160,180,200,220,250,280,315,355,400]

for j in range(len(key_series)):
    length_of_key=key_series[j]
    if length <= length_of_key:
        print("Length of the key according to preferred number is:"+str(length_of_key)+"mm")
        break


x = 50
y = 50
key_x=100
key_y=300


#max_length=500,max_diameter_of_shaft=330
scale=length_of_shaft/500
if scale >= 1:
    scl_dwn_length_of_shaft=round(length_of_shaft/scale,2)
    scl_dwn_diameter_of_shaft=round(diameter_of_shaft/scale,2)
    scl_dwn_length_of_key=round(length_of_key/scale,2)
    scl_dwn_key_thickness=round(key_thickness/scale,2)
    scl_dwn_key_width=round(key_width/scale,2)
else:
    scl_dwn_length_of_shaft=length_of_shaft
    scl_dwn_diameter_of_shaft=diameter_of_shaft
    scl_dwn_length_of_key=length_of_key
    scl_dwn_key_thickness=key_thickness
    scl_dwn_key_width=key_width


y_intercept=math.sqrt(((diameter_of_shaft/2)**2)-(key_width/2)**2)
y_intercept=round(abs(y_intercept),2)
#print(y_intercept)
print("we are going to draw shaft and key in this program")
print("To view the drawings open the python popup dialogue box!")
```

```
#print(scale,length,diameter_of_shaft)

my_window= Tk()

# creating A2 sheet in landscape mode

my_canvas= Canvas(my_window,width=594,height=420,background="white")

my_window.resizable(width="False",height="False")

my_canvas.grid(row=0,column=0)

my_canvas.create_rectangle(10,10,587,414)

#shaft top view

my_canvas.create_rectangle(x,y,x+scl_dwn_length_of_shaft,y+scl_dwn_diameter_of_shaft)

#To create rectangle for key slot

my_canvas.create_rectangle(x+3+(scl_dwn_key_width/2),y+(scl_dwn_diameter_of_shaft/2)-
(scl_dwn_key_width/2),x+3+(scl_dwn_key_width/2)+scl_dwn_length_of_key,y+(scl_dwn_diameter_
of_shaft/2)+(scl_dwn_key_width/2))

my_canvas.create_rectangle((x+scl_dwn_length_of_shaft)-
(10+scl_dwn_length_of_key),y+(scl_dwn_diameter_of_shaft/2)-
(scl_dwn_key_width/2),x+scl_dwn_length_of_shaft-
10,y+(scl_dwn_diameter_of_shaft/2)+(scl_dwn_key_width/2))


#shaft side view

my_canvas.create_oval(x+350,y+diameter_of_shaft+50,x+350+diameter_of_shaft,y+50+(2*diameter
_of_shaft))

my_canvas.create_line((x+350+(diameter_of_shaft/2)-
(key_width/2)),y+diameter_of_shaft+50+(key_thickness/2),(x+350+(diameter_of_shaft/2)+(key_widt
h/2)),y+diameter_of_shaft+50+(key_thickness/2))


#To draw the key slot

my_canvas.create_line((x+350+(diameter_of_shaft/2)-
(key_width/2)),y+diameter_of_shaft+50+(key_thickness/2),(x+350+(diameter_of_shaft/2)-
(key_width/2)),y+diameter_of_shaft+50+((diameter_of_shaft/2)-y_intercept))

my_canvas.create_line((x+350+(diameter_of_shaft/2)+(key_width/2)),y+diameter_of_shaft+50+(key
_thickness/2),(x+350+(diameter_of_shaft/2)+(key_width/2)),y+diameter_of_shaft+50+((diameter_of
_shaft/2)-y_intercept))

#key front view

taper_side_height=(key_thickness-(0.01*length_of_key))

my_canvas.create_line(key_x,key_y,key_x,key_y+key_thickness)

my_canvas.create_line(key_x,key_y+key_thickness,key_x+length_of_key,key_y+key_thickness)
```

```
my_canvas.create_line(key_x+length_of_key,key_y+key_thickness,key_x+length_of_key,key_y+key_thickness-taper_side_height)

my_canvas.create_line(key_x+length_of_key,key_y+key_thickness-taper_side_height,key_x,key_y)


#key side view

my_canvas.create_rectangle(key_x+length_of_key+30,key_y,key_x+length_of_key+30+key_width,key_y+key_thickness)

#To create a line to show the tapper

my_canvas.create_line(key_x+length_of_key+30,key_y+key_thickness-taper_side_height,key_x+length_of_key+30+key_width,key_y+key_thickness-taper_side_height)

my_window.mainloop()
```

SAMPLE OUTPUT:

In this program, you can find out the diameter of the shaft for required load conditions

Enter the length of the shaft in mm:2500

Enter factor of safety for the shaft:2

Enter various shaft loads

Enter the maximum bending moment in N.mm:562500

Enter true to calculate the torque using power and speed or false to directly enter the amount of torque: true

Enter power to be transmitted by shaft in Watt and corresponding speed in rpm:20000 200

Torque in the shaft is:954929.658551372N.mm

Enter the shaft material properties

Enter ultimate tensile stress of the material in Mpa:112

Enter ultimate shear stress of the material in Mpa:84

To get values for Kb and Kt, press GL for gradual loading, MSL for minor shock load and HSL for heavy shock load:GL

Equivalent bending moment is 1059017.9814572707N.mm

Equivalent twisting moment is 1274285.9629145414N.mm

Diameter of the shaft for bending is:57mm

Diameter of the shaft rounded off to R20 series is:63mm

To find out dimensions of rectangular sunk key

Key_width is:20mm and key_thickness is:12mm

Enter the key material properties

Enter crushing stress of the material in Mpa:140

Enter ultimate shear stress of the material in Mpa:84

Enter factor of safety for the shaft:2

Length of the key w.r.t shear stress is:155.8622655262236mm

Length of the key according to preferred number is:160mm

we are going to draw shaft and key in this program

To view the drawing open the python popup dialogue box!

Note:

Yellow coloured values have to be provided as input for the program

Green coloured values are the important dimension of shaft and key

The resultant shaft and key drawings can be seen in the below dialogue box