

Example of a Python code

The program here calculates depth below a ground surface using radiolocation.

```
# This program calculates the depth below the surface using radiolocation
# Uses the distance where the receiver is from ground zero and angle of the
receiver
# Assumes the angle of the receiver is in degrees
# Inputs entered with keyboard
# Outputs displayed on screen

def main():
    import math
    degree_sign= u'\N{DEGREE SIGN}'

    def CalculateDepth(angleDeg, distG):
        angle_radians = math.radians(angleDeg)
        caveDepth = ((2*distG)/(math.sqrt(9*math.tan(angle_radians)**2+8)-
(3*math.tan(angle_radians))))
        caveDepth = round(caveDepth, 1)
# round the depth to 1 decimal place
        return caveDepth

    # Display program purpose
    print("This application calculates depth below a ground surface")
    print("Assumption: depth calculated is not affected by transmitter
characteristics or rock variations")
    print()
    print('*****
*****')
    print()

    ##### Start of input #####

    # angle and distance lists to hold user input
    input_angles = [] # create empty list for receiver angles
    input_distances = [] # create empty list for distance to GZ

    # Obtain angle and distance for multiple locations from the user
    while True:
        angle_degrees = float(input("\tEnter the angle of the receiver in
degrees: "))
        input_angles.append(angle_degrees)
        # add angle to input_angles list
        distGZ = float(input("\tEnter the distance from GZ to the receiver in
metres: "))
        input_distances.append(distGZ)
        # add distance to input_distances list
```

```

print()
# determine whether user wants to enter another set of input values
end = str(input("Do you want to stop entering values (Y/N)? "))
print()
if end.upper() == 'Y' :
    break

##### End of input, Start of calculations #####

# cave depth list to hold calculated results
calc_depths = []          # create empty list for cave depths

# For each data set, calculate depth and add to depth list
for index in range(len(input_angles)):
# index should be 0, 1, 2, ... to last index in lists
    angle_degrees = input_angles[index]
# retrieve angle in index position from input_angles list
    distGZ = input_distances[index]
# retrieve distance in index position from input_distances list

    angle_radian = math.radians(angle_degrees)
    ResultDepth = CalculateDepth(angle_radian,distGZ)
    calc_depths.append(ResultDepth)
# add depth to calc_depths list

##### End of calculations, Start of output #####

print("-----")
-----")
# Display column header line for table
print("Angle (" + degree_sign + ")\tDistance(m)\t\tDepth(m)")

# Display angles, distances, depths from lists in table format
for index in range(len(input_angles)):
# index should be 0, 1, 2, ... to last index in lists

# print angle, distance, depth in current index position from lists with
decimals aligned
    print(input_angles[index], input_distances[index],
calc_depths[index])

print()
print("Done")

if __name__ == '__main__':
    main()

```

Note: The code here is copied from "Visual Studio Code" to replicate the color forms present in the file.