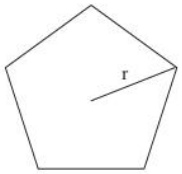
Date: __/09/2024

Assignment 02

Name: MEHRIN FARZANA
ID: 2101013

Problem no. 1.

3.1 (*Geometry: area of a pentagon*) Write a program that prompts the user to enter the length from the center of a pentagon to a vertex and computes the area of the pentagon, as shown in the following figure.



The formula for computing the area of a pentagon is $Area = \frac{3\sqrt{3}}{2}s^2$, where s is the length of a side. The side can be computed using the formula $s = 2r\sin\frac{\pi}{5}$, where r is the length from the center of a pentagon to a vertex. Here is a sample run:

```
Enter the length from the center to a vertex: 5.5 ↵Enter
The area of the pentagon is 108.61
```

Solution:

Code:

```
import math
x=float(input("Enter the length from the center to a vertex:"))
print("The area of the pentagon is", end=" ")
y= round(3*math.sqrt(3)*(2*x*math.sin(math.pi/5))**2/2, 2)
print(y)
```

Output:

```
Enter the length from the center to a vertex:5.5
The area of the pentagon is 108.61
```

Figure 1.1: Output

Problem no. 2.

***3.2** (*Geometry: great circle distance*) The great circle distance is the distance between two points on the surface of a sphere. Let $(x1, y1)$ and $(x2, y2)$ be the geographical latitude and longitude of two points. The great circle distance between the two points can be computed using the following formula:

$$d = radius \times \arccos(\sin(x_1) \times \sin(x_2) + \cos(x_1) \times \cos(x_2) \times \cos(y_1 - y_2))$$

Write a program that prompts the user to enter the latitude and longitude of two points on the earth in degrees and displays its great circle distance. The average earth radius is 6,371.01 km. Note that you need to convert the degrees into radians using the `math.radians` function since the Python trigonometric functions use radians. The latitude and longitude degrees in the formula are for north and west. Use negative to indicate south and east degrees. Here is a sample run:

```
Enter point 1 (latitude and longitude) in degrees:
39.55, -116.25 ↵ Enter
Enter point 2 (latitude and longitude) in degrees:
41.5, 87.37 ↵ Enter
The distance between the two points is 10691.79183231593 km
```

Solution:

Code:

```
import math
x1, y1 = map(float, input("Enter point 1 (latitude and
longitude) in degrees: ").split(","))
x2, y2 = map(float, input("Enter point 2 (latitude and
longitude) in degrees: ").split(","))
x1, y1, x2, y2 = math.radians(x1), math.radians(y1),
math.radians(x2), math.radians(y2)
d = (6371.01 * (math.acos(math.sin(x1) * math.sin(x2) +
math.cos(x1) * math.cos(x2) * math.cos(y1- y2))))
print("The distance between the two points is ", end=" ")
print(d, end=" ")
print("km")
```

Output:

```
Enter point 1 (latitude and longitude) in degrees: 39.55, -116.25
Enter point 2 (latitude and longitude) in degrees: 41.5, 87.37
The distance between the two points is  10691.79183231593 km
```

Figure 2.1: Output

Problem no. 3.

***3.3** (*Geography: estimate areas*) Find the GPS locations for Atlanta, Georgia; Orlando, Florida; Savannah, Georgia; and Charlotte, North Carolina from www.gps-data-team.com/map/ and compute the estimated area enclosed by these four cities. (Hint: Use the formula in Programming Exercise 3.2 to compute the distance between two cities. Divide the polygon into two triangles and use the formula in Programming Exercise 2.14 to compute the area of a triangle.)

Solution:

Code:

```
import math
x1, y1= -84.387917 , 33.757222
x2, y2= -81.5197376 , 28.3703828
x3, y3= -81.0983 , 32.0749
x4, y4= -80.8449 , 35.0589
x1, y1, x2, y2, x3, y3, x4, y4= math.radians(x1),
math.radians(y1), math.radians(x2), math.radians(y2),
math.radians(x3), math.radians(y3), math.radians(x4),
math.radians(y4)
d1 = (6371.01 * (math.acos(math.sin(x1) * math.sin(x2) +
math.cos(x1) * math.cos(x2) * math.cos(y1- y2))))
d2 = (6371.01 * (math.acos(math.sin(x2) * math.sin(x3) +
math.cos(x2) * math.cos(x3) * math.cos(y2- y3))))
d3 = (6371.01 * (math.acos(math.sin(x1) * math.sin(x3) +
math.cos(x1) * math.cos(x3) * math.cos(y1- y3))))
d4 = (6371.01 * (math.acos(math.sin(x1) * math.sin(x4) +
math.cos(x1) * math.cos(x4) * math.cos(y1- y4))))
d5 = (6371.01 * (math.acos(math.sin(x3) * math.sin(x4) +
math.cos(x3) * math.cos(x4) * math.cos(y3- y4))))
s1 = (d1 + d2 + d3) / 2
s2 = (d3 + d4 + d5) / 2
a1 = round(math.sqrt(s1*(s1- d1)*(s1- d2)*(s1- d3)), 1)
a2 = round(math.sqrt(s2*(s2- d3)*(s2- d4)*(s2- d5)), 1)
a = a1 + a2
print("Estimated area is", end=" ")
print(a, end=" ")
print("km^2")
```

Output:

```
Estimated area is 21460.6 km^2
```

Figure 3.1: Output

Problem no. 4.

3.4 (*Geometry: area of a pentagon*) The area of a pentagon can be computed using the following formula (`s` is the length of a side):

$$Area = \frac{5 \times s^2}{4 \times \tan\left(\frac{\pi}{5}\right)}$$

Write a program that prompts the user to enter the side of a pentagon and displays the area. Here is a sample run:

```
Enter the side: 5.5 ↵Enter
The area of the pentagon is 53.04444136781625
```

Solution:

Code:

```
import math
s=float(input("Enter the side: "))
area = (5 * s**2)/ (4 * math.tan(math.pi/5))
print("The area of the pentagon is", area)
```

Output:

```
Enter the side: 5.5
The area of the pentagon is 52.044441367816255
```

Figure 4.1: Output

Problem no. 5.

***3.5** (*Geometry: area of a regular polygon*) A regular polygon is an *n*-sided polygon in which all sides are of the same length and all angles have the same degree (i.e., the polygon is both equilateral and equiangular). The formula for computing the area of a regular polygon is

$$Area = \frac{n \times s^2}{4 \times \tan\left(\frac{\pi}{n}\right)}$$

Here, `s` is the length of a side. Write a program that prompts the user to enter the number of sides and their length of a regular polygon and displays its area. Here is a sample run:

```
Enter the number of sides: 5 Enter
Enter the side: 6.5 Enter
The area of the polygon is 73.69017017488385
```

Solution:

Code:

```
import math
n=float(input("Enter the number of sides: "))
s=float(input("Enter the side: "))
area = (n * s**2)/ (4 * math.tan(math.pi/n))
print("The area of the polygon is", area)
```

Output:

```
Enter the number of sides: 5
Enter the side: 6.5
The area of the polygon is 72.69017017488386
```

Figure 5.1: Output

Problem no. 6.

***3.6** (*Find the character of an ASCII code*) Write a program that receives an ASCII code (an integer between 0 and 127) and displays its character. For example, if the user enters 97, the program displays the character a. Here is a sample run:

```
Enter an ASCII code: 69 ←Enter
The character is E
```

Solution:

Code:

```
a=int(input("Enter an ASCII code: "))
c=chr(a)
print("The character is ", end="")
print(c)
```

Output:

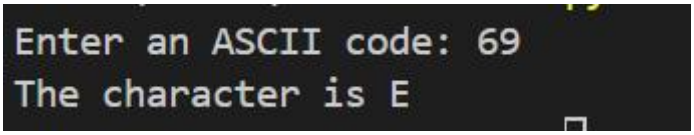


Figure 6.1: Output

Problem no. 7.

3.7 (*Random character*) Write a program that displays a random uppercase letter using the `time.time()` function.

Solution:

Code:

```
import time
import random
time.time()
#a=int(random.random() % 26 + 65)
a=random.randint(65,90)
c=chr(a)
print(c)
```

Output:

```
PS C:\Users\BAB AL SAFA> python -u "c:\Users\BAB AL SAFA\OneDrive\Desktop\abc.py"
C
PS C:\Users\BAB AL SAFA> python -u "c:\Users\BAB AL SAFA\OneDrive\Desktop\abc.py"
W
```

Figure 7.1: Output

Problem no. 8.

***3.8** (*Financial application: monetary units*) Rewrite Listing 3.4, ComputeChange.py, to fix the possible loss of accuracy when converting a float value to an int value. Enter the input as an integer whose last two digits represent the cents. For example, the input **1156** represents **11** dollars and **56** cents.

Solution:

Code:

```
amount = int(input("Enter an amount, for example, 1156 represents
11 dollars and 56 cents: "))
numberOfOneDollars = amount // 100
remainingAmount = amount % 100
numberOfQuarters = remainingAmount // 25
remainingAmount = remainingAmount % 25
numberOfDimes = remainingAmount // 10
remainingAmount = remainingAmount % 10
numberOfNickels = remainingAmount // 5
remainingAmount = remainingAmount % 5
numberOfPennies = remainingAmount
print("Your amount", amount, "consists of\n",
"\t", numberOfOneDollars, "dollars\n",
"\t", numberOfQuarters, "quarters\n",
"\t", numberOfDimes,  "dimes\n",
"\t", numberOfNickels, "nickels\n",
"\t", numberOfPennies, "pennies")
```

Output:

```
Enter an amount, for example, 1156 represents 11 dollars and 56 cents: 1156
Your amount 1156 consists of
        11 dollars
        2 quarters
        0 dimes
        1 nickels
        1 pennies
```

Figure 8.1: Output

Problem no. 9.

***3.9** (*Financial application: payroll*) Write a program that reads the following information and prints a payroll statement:

Employee's name (e.g., Smith)
Number of hours worked in a week (e.g., 10)
Hourly pay rate (e.g., 9.75)
Federal tax withholding rate (e.g., 20%)
State tax withholding rate (e.g., 9%)

A sample run is shown below:

```
Enter employee's name: Smith Enter
Enter number of hours worked in a week: 10 Enter
Enter hourly pay rate: 9.75 Enter
Enter federal tax withholding rate: 0.20 Enter
Enter state tax withholding rate: 0.09 Enter

Employee Name: Smith
Hours Worked: 10.0
Pay Rate: $9.75
Gross Pay: $97.5
Deductions:
  Federal Withholding (20.0%): $19.5
  State Withholding (9.0%): $8.77
  Total Deduction: $28.27
Net Pay: $69.22
```

Solution:

Code:

```
name = input("Enter employee's name: ")
work = float(input("Enter number of hours worked in a week: "))
rate = float(input("Enter hourly pay rate: "))
fedTax = float(input("Enter federal tax withholding rate: "))
stTax = float(input("Enter state tax withholding rate: "))

print("Employee Name: ", end="")
print(name)
print("Hours Worked: ", end ="")
print(work)
print("Pay Rate: $", end ="")
print(rate)
print("Gross Pay: $", end ="")
grossPay=round(work*rate, 2)
print(grossPay)
print("Deductions:")
print("Federal Withholding (", end="")
```

```
print(fedTax*100, end="")
print("%): $", end="")
fedTaxDed = round(grossPay*fedTax, 2)
print(fedTaxDed)
print("State Withholding (", end="")
print(stTax*100, end="")
print("%): $", end="")
stTaxDed = round(grossPay*stTax, 2)
print(stTaxDed)
print("Total Deduction: $", end="")
print(fedTaxDed + stTaxDed)
print("Net Pay: $", end="")
print(grossPay - (fedTaxDed + stTaxDed))
```

Output:

```
Enter employee's name: Smith
Enter number of hours worked in a week: 10
Enter hourly pay rate: 9.75
Enter federal tax withholding rate: 0.20
Enter state tax withholding rate: 0.09
Employee Name: Smith
Hours Worked: 10.0
Pay Rate: $9.75
Gross Pay: $97.5
Deductions:
Federal Withholding (20.0%): $19.5
State Withholding (9.0%): $8.78 2
Total Deduction: $28.28
Net Pay: $69.22
```

Figure 9.1: Output

Problem no. 10.

***3.10** (*Turtle: display Unicodes*) Write a program to display Greek letters $\alpha\beta\gamma\delta\epsilon\zeta\eta\theta$. The Unicode of these characters are `\u03b1 \u03b2 \u03b3 \u03b4 \u03b5 \u03b6 \u03b7 \u03b8`.

Solution:

Code:

```
import turtle

screen = turtle.Screen()
screen.title("Greek Letters Display")
pen = turtle.Turtle()
pen.hideturtle()
pen.penup()
greek_letters = ['\u03b1', '\u03b2', '\u03b3', '\u03b4',
'\u03b5', '\u03b6', '\u03b7', '\u03b8']
positions = [(-200, 0), (-150, 0), (-100, 0), (-50, 0), (0, 0),
(50, 0), (100, 0), (150, 0)]
for letter, position in zip(greek_letters, positions):
    pen.goto(position)
    pen.write(letter, font=("Arial", 40, "normal"))
turtle.done()
```

Output:

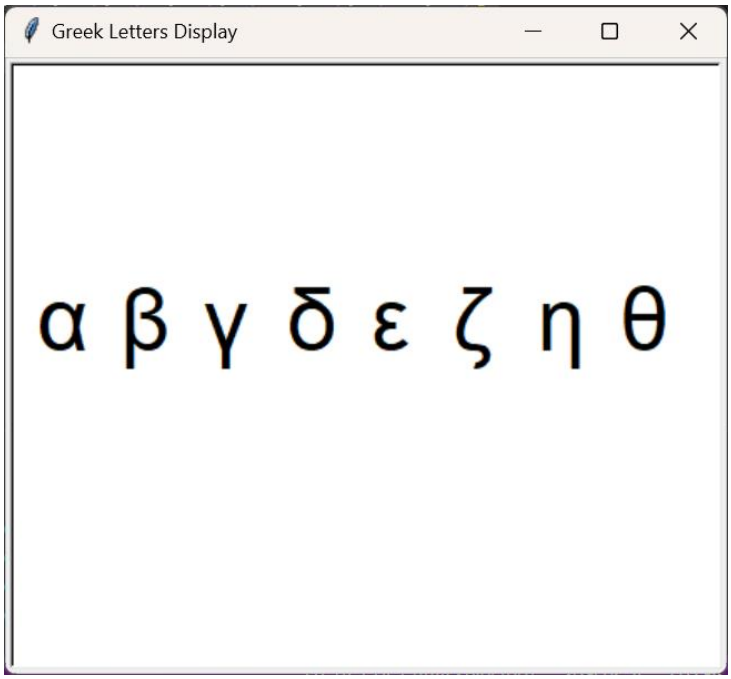


Figure 10.1: Output

Problem no. 11.

3.11 (*Reverse number*) Write a program that prompts the user to enter a four-digit integer and displays the number in reverse order. Here is a sample run:

```
Enter an integer: 3125 ↵ Enter
The reversed number is 5213
```

Solution:

Code:

```
n = int(input("Enter an integer: "))
print("The reversed number is", end=" ")
while(n>0):
 print(n%10, end="")
 n //=10
```

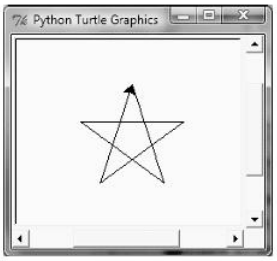
Output:

```
Enter an integer: 3125
The reversed number is 5213
```

Figure 11.1: Output

Problem no. 12.

**3.12 (*Turtle: draw a star*) Write a program that prompts the user to enter the length of the star and draw a star, as shown in Figure 3.5a. (Hint: The inner angle of each point in the star is 36 degrees.)



(a)

Solution:

Code:

```
import turtle

def draw_star(length):
    angle = 144
    for _ in range(5):
        turtle.right(angle)
        turtle.forward(length)
```

```
turtle.speed(1)
length = int(input("Enter the length of the star: "))
draw_star(length)
turtle.done()
```

Output:

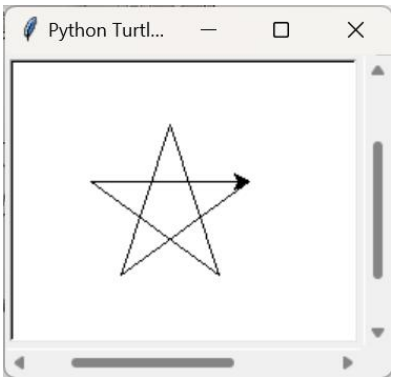


Figure 12.1: Output