Cau 1:: Tính chu vi diện tích Hình tròn

import math

try:

r=float(input("Mời bạn nhập bán kính hình tròn:"))

cv=2\*math.pi\*r dt=r\*\*2

print("Chu vi =",cv) print("Diện tích=",dt)

except:

print("Lỗi rồi!")

Câu 2: Tính giờ phút giây

t=int(input("Nhập số giây:"))

hour=(t//3600)%24

minute=(t%3600)//60

second=(t%3600)%60

print(hour,":",minute,":",second)

Câu 3: Tính điểm trung bình

toan=float(input("Nhập điểm Toán:"))

ly=float(input("Nhập điểm lý:"))

hoa=float(input("Nhập điểm hóa:"))

dtb=(toan+ly+hoa)/3

print("Điểm trung bình=",dtb)

print("Điểm làm tròn=",round(dtb,2))

Câu 4: Python hỗ trợ những kiểu dữ liệu cơ bản nào?

Kiểu số (Numeric Types):

int: Số nguyên

float: Số thực

complex: Số phức

Kiểu chuỗi (String Type):

str: Chuỗi ký tự

Kiểu luận lý (Boolean Type):

bool: Giá trị đúng hoặc sai

Kiểu danh sách (List Type):

list: Danh sách các phần tử

Kiểu bộ (Tuple Type):

tuple: Bộ các phần tử không thay đổi

Kiểu phạm vi (Range Type):

range: Dãy số, thường dùng trong vòng lặp,

Kiểu từ điển (Dictionary Type):

dict: Tập hợp các cặp khóa-giá trị

Kiểu tập hợp (Set Types):

set: Tập hợp các phần tử không trùng lặp, ví dụ: {1, 2, 3}.

frozenset: Tập hợp không thay đổi, ví dụ: frozenset({1, 2, 3}).

Kiểu nhị phân (Binary Types):

bytes: Dữ liệu nhị phân, ví dụ: b"hello".

bytearray: Mảng byte có thể thay đổi.

memoryview: Đối tượng bộ nhớ.

Câu 5: Trình bày các loại ghi chú trong Python.

Ghi chú một dòng (Single-line comments):

Được bắt đầu bằng dấu # và kéo dài đến hết dòng.

Ghi chú nhiều dòng (Multi-line comments):

Được bao quanh bởi ba dấu nháy đơn (''') hoặc ba dấu nháy kép ("""). Thường được dùng để ghi chú dài hoặc giải thích chi tiết hơn.

Ví dụ:

"""

Đây là một ghi chú nhiều dòng.

Nó có thể kéo dài qua nhiều dòng.

"""

Câu 6: Trình bày ý nghĩa toán tử /, //, %, \*\*, and, or, is.

Toán tử chia /:

Dùng để chia hai số và trả về kết quả là một **số thực** (float). Ví dụ: 5 / 2 sẽ trả về 2.5.

Toán tử chia lấy phần nguyên //:

Dùng để chia hai số và trả về **phần nguyên** của kết quả.

Ví dụ: 5 // 2 sẽ trả về 2.

Toán tử chia lấy phần dư %:

Dùng để chia hai số và trả về **phần dư** của phép chia.

Ví dụ: 5 % 2 sẽ trả về 1.

Toán tử lũy thừa \*\*:

Dùng để **tính lũy thừa** của một số.

Ví dụ: 2 \*\* 3 sẽ trả về 8.

Toán tử logic and:

Trả về **True** nếu cả hai biểu thức **đều đúng**, ngược lại trả về **False**.

Ví dụ: True and False sẽ trả về False.

Toán tử logic or:

Trả về True nếu **ít nhất** một trong hai biểu thức đúng, ngược lại trả về False.

Ví dụ: True or False sẽ trả về True.

Toán tử so sánh is:

Dùng để kiểm tra xem hai đối tượng **có cùng tham chiếu** trong bộ nhớ hay không.

Ví dụ: a is b sẽ trả về True nếu a và b là cùng một đối tượng.

Câu 7: Trình bày một số cách nhập dữ liệu từ bàn phím.

Sử dụng hàm input():

Đây là cách cơ bản nhất để nhập dữ liệu từ bàn phím. Hàm **input()** sẽ dừng chương trình và chờ người dùng nhập dữ liệu, sau đó trả về dữ liệu dưới dạng chuỗi.

VD:

name = input("Enter your name: ")

print("Hello, " + name)

Chuyển đổi kiểu dữ liệu:

Dữ liệu nhập vào từ hàm **input()** luôn là chuỗi. Để chuyển đổi sang các kiểu dữ liệu khác như **số nguyên (int)** hoặc **số thực (float)**, bạn có thể sử dụng các hàm chuyển đổi.

VD:

age = int(input("Enter your age: "))

height = float(input("Enter your height: "))

print("Age:", age)

print("Height:", height)

Nhập nhiều giá trị cùng lúc:

Bạn có thể sử dụng hàm **split()** để nhập nhiều giá trị từ một dòng duy nhất và tách chúng ra thành các biến khác nhau.

VD:

x, y = input("Enter two numbers separated by space: ").split()

print("First number:", x)

print("Second number:", y)

Sử dụng map() để chuyển đổi kiểu dữ liệu cho nhiều giá trị:

Kết hợp **split() và map()** để chuyển đổi kiểu dữ liệu cho nhiều giá trị cùng lúc.

VD:

x, y = map(int, input("Enter two integers separated by space: ").split())

print("Sum:", x + y)

Câu 8: Trình bày các loại lỗi khi lập trình và cách bắt lỗi trong Python.

Lỗi cú pháp (Syntax Errors): Xảy ra khi mã nguồn vi phạm các quy tắc cú pháp của Python. Ví dụ, thiếu dấu hai chấm (:) sau câu lệnh if.

Lỗi thời gian chạy (Runtime Errors): Xảy ra khi chương trình đang chạy và gặp phải tình huống không mong muốn, chẳng hạn như chia cho số 0 hoặc truy cập vào chỉ số ngoài phạm vi.

Lỗi logic (Logical Errors): Xảy ra khi mã nguồn chạy mà không gặp lỗi cú pháp hay thời gian chạy, nhưng kết quả không như mong đợi. Ví dụ, sai sót trong công thức tính toán.

Câu 9: Giải thích kết quả tính toán của các biểu thức

Cho các biến với giá trị

i1 = 2

i2 = 5

i3 = -3

d1 = 2.0

d2 = 5.0

d3 = -0.5

(a) i1 + (i2 \* i3) = -13

(b) i1 \* (i2 + i3) = 4

(c) i1 / (i2 + i3) =  2 / 2  =1.0

(d) i1 // (i2 + i3) = 2//2 = 1

(e) i1 / i2 + i3 = 2 / 5 + (-3)  = -2.6

(f) i1 // i2 + i3 = 2 / / 5 + (-3) = 0 + (-3) = -3

(g) 3 + 4 + 5 / 3 = 3 + 4 + 1.6667 = 8.6667

(h) 3 + 4 + 5 // 3 = 3 + 4 + 1 = 8

(i) (3 + 4 + 5) / 3 = 4.0

(j) (3 + 4 + 5) // 3 = 4

(k) d1 + (d2 \* d3) = 2.0 + (-2.5) = -0.5

(l) d1 + d2 \* d3 = 2.0 + (-2.5) = -0.5

(m) d1 / d2 - d3 = 2.0 / 5.0 - (-0.5) = 0.4 + 0.5 = 0.9

(n) d1 / (d2 - d3)= 2.0 / (5.0 - (-0.5)) = 2.0 / 5.5 = 0.3636

(o) d1 + d2 + d3 / 3 = 2.0 + 5.0 + (-0.5) / 3 = 2.0 + 5.0 - 0.1667 = 6.8333

(p) (d1 + d2 + d3) / 3 = 2.0 + 5.0 + (-0.5)) / 3 = 6.5 / 3 = 2.1667

(q) d1 + d2 + (d3 / 3) = 2.0 + 5.0 + (-0.5) / 3 = 2.0 + 5.0 - 0.1667 = 6.8333

(r) 3 \* (d1 + d2) \* (d1 - d3)= 3 \* (2.0 + 5.0) \* (2.0 - (-0.5)) = 3 \* 7.0 \* 2.5 = 52.5

Câu 10: Hãy viết ngắn gọn lại các lệnh dưới đây

1. x += 1
2. x /= 2
3. x -= 1
4. x += y
5. x -= (y + 7)
6. x \*= 2

(g) number\_of\_closed\_cases += 2 \* ncc