Câu 4:

* Int: số nguyên (ví dụ:1,-5,100)
* Float: số thực(ví dụ:3.14, 0.5)
* Bool: kiểu logic(True, False)
* Str: chuỗi kí tự(ví dụ:”Hello Python”)
* List: Danh sách(ví dụ:[1,2,3])
* Tuple: bộ giá trị(ví dụ:(1,2,3))
* Dict: từ điển(ví dụ:{“A”:1, “b”:2})
* Set: tập hợp(ví dụ:(1,2,3))

Câu 5:

1. Ghi chú 1 dòng

Sử dụng dấu # ở đầu dòng. Nội dung sau dấu # sẽ được Python bỏ qua khi thực thi.

Ví dụ:

X=5 #Ghi chú sau lệnh

1. Ghi chú nhiều dòng

(docstring or comment)

Dùng dấu ba nháy đơn (‘’’…’’’) hoặc ba dấu nháy kép (“””…”””)

Thường dùng để mô tả hàm, lớp, module.

Câu 6:

* /: Chia lấy kết quả số thực (float).

Ví dụ: 5/2 kết quả là 2.5

* //: chia lấy phần nguyên(floor division)

Ví dụ: 5//2 kết quả là 2

* %: chia lấy phần dư

Ví dụ: 5%2 kết quả là 1

* \*\*:lũy thừa

Ví dụ:2\*\*3 kết quả là 8

* And: toán tử logic “và”. Kết quả là True nếu cả 2 đều đúng.

ví dụ: True and False nếu kết quả là False.

* Or: toán tử logic “hoặc”. kết quả là true nếu ít nhất nột vế đúng.
* Is: so sánh hai đối tượng có cùng tham chiếu không(so sánh địa chỉ vùng nhớ)

Câu 7:

1. Sử dụng hàm

Input()

Nhập dữ liệu dưới dạng chuỗi(str).

Ví dụ: name = input(“Nhập tên của bạn: “)

1. Chuyển đổi kiểu dữ liệu khi nhập

Dùng hàm chuyển đổi như int(). Float() để nhập số nguyên hoặc số thực.

Ví dụ: age = int(input(“Nhập tuổi: “))

Score = float(input(“Nhập điểm: “))

1. Nhập nhiều giá trị trên một dòng

Dung split() để tách các giá trị nhập vào.

Ví dụ: a, b= input(“Nhập bởi dấu cách: “).split()

a = int(a)

b = int(b)

1. Nhập danh sách số

Dung list comprehension kết hợp với split() và chuyển kiểu.

Ví dụ: numbers = [int(x) for x in input(“Nhập các số, cách nhau bởi dấu cách: “).split()]

Câu 8:

1. Lỗi cú pháp
2. Lỗi khi chạy (Runtime Error)
3. Lỗi logic (Logic Error)

Cách bắt lỗi trong Python:

Try: đặt code có thể gây lỗi.

Except: xử lý lỗi cụ thể

Else: thực hiện khi không có lỗi.

Finally: luôn thực hiện, dù có lỗi hay không.

Câu 9:

…

(a) i1 + (i2 \* i3)

= 2 + (5 \* -3)

= 2 + (-15)

= -13

(b) i1 \* (i2 + i3)

= 2 \* (5 + -3)

= 2 \* 2

= 4

(c) i1 / (i2 + i3)

= 2 / (5 + -3)

= 2 / 2

= 1.0

(d) i1 // (i2 + i3)

= 2 // (5 + -3)

= 2 // 2

= 1

(e) i1 / i2 + i3

= 2 / 5 + (-3)

= 0.4 - 3

= -2.6

(f) i1 // i2 + i3

= 2 // 5 + (-3)

= 0 + (-3)

= -3

(g) 3 + 4 + 5 / 3

= 3 + 4 + 1.666...

= 8.666...

(h) 3 + 4 + 5 // 3

= 3 + 4 + 1

= 8

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

= 12 / 3

= 4.0

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

= 12 // 3

= 4

(k) d1 + (d2 \* d3)

= 2.0 + (5.0 \* -0.5)

= 2.0 - 2.5

= -0.5

(l) d1 + d2 \* d3

= 2.0 + 5.0 \* -0.5

= 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)

= 7.0 - 0.166...

= 6.833...

(p) (d1 + d2 + d3) / 3

= (2.0 + 5.0 - 0.5) / 3

= 6.5 / 3

≈ 2.166...

(q) d1 + d2 + (d3 / 3)

= 2.0 + 5.0 + (-0.5 / 3)

= 7.0 - 0.166...

= 6.833... (giống (o))

(r) 3 \* (d1 + d2) \* (d1 - d3)

= 3 \* (2.0 + 5.0) \* (2.0 - -0.5)

= 3 \* 7.0 \* 2.5

= 21 \* 2.5

= 52.5

'''

Câu 10:

# Viết ngắn gọn lại các lệnh dưới đây

'''

(a) x = x + 1

-> x += 1

(b) x = x / 2

-> x /= 2

(c) x = x - 1

-> x -= 1

(d) x = x + y

-> x += y

(e) x = x - (y + 7)

-> x -= (y + 7)

(f) x = 2\*x

-> x \*= 2

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

-> number\_of\_closed\_cases += 2 \* ncc

'''