**BÀI TẬP CHƯƠNG 2**

*Câu 4: Python hỗ trợ những loại dữ liệu cơ bản nào.*

* Kiểu số nguyên (int), số thực (float), kiểu logic (boolean) và chuỗi ký tự (string).

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

* Ghi chú một dòng: Sử dụng ký tự # để bắt đầu một ghi chú một dòng.
* Ghi chú nhiều dòng: Sử dụng ba cặp nháy đôi liên tiếp để bắt đầu và ba cặp nháy đôi liên tiếp để kết thúc hoặc ba cặp nháyđơn liên tiếp để bắt đầu và ba cặp nháy đơn liên tiếp để kết thúc.

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

* */: Chia thường (trả về kết quả số thực).*
* *//: Chia lấy phần nguyên.*
* *%: Chia lấy phần dư.*
* *\*\*: Lũy thừa.*
* *And: Được gọi là toán tử logic AND (và). Nếu cả hai toán tử đều có giá trị khác 0, thì điều kiện trở lên true.*
* *Or: Được gọi là toán tử logic OR (hoặc). Nếu một trong hai toán tử đều có giá trị khác 0, thì điều kiện là true.*
* *Is: Toán từ kiểm tra biến có hằng hay không, toán từ này tương đương với toán tử.*

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

* Python cung cấp hàm input() để nhập một dãy ký tự từ bàn phím.

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

* Ba loại lỗi thường gặp khi thông dịch một chương trình Python là lỗi cú pháp, lỗi thực thì và lỗi ngữ nghĩa. Trong đó, lỗi cú pháp sẽ được thông báo bởi trình dịch. Lỗi ngữ nghĩa cần phải sử dụng kinh nghiệm của lập trình viên để gỡ lỗi. Đối với lỗi thực thi, mỗi khi xảy ra chương trình sẽ tạo ra một đối tượng ngoại lệ gọi là exception.
* Có nhiều cách xử lý ngoại lệ trong Python, theo các cú pháp sau:
* Cú pháp 1: Xử lý ngoại lệ cơ bản.
* Cú pháp 2: Mệnh đề except mà không xác định Exception trong Python.
* Cú pháp 3: Mệnh đề except với nhiều exception trong Python.
* Củ pháp 4: Khối try-finally trong Python.

*Câu 9:*

Giá trị:

i1 = 2, i2 = 5, i3 = -3, d1 = 2.0, d2 = 5.0, d3 = -0.5

Thực hiện:

(a) i1 + (i2 \* i3)

- i2 \* i3 = 5 \* (-3) = -15

- i1 + (-15) = 2 + (-15) = -13

- Kết quả: -13

(b) i1 \* (i2 + i3)

- i2 + i3 = 5 + (-3) = 2

- i1 \* 2 = 2 \* 2 = 4

- Kết quả: 4

(c) i1 / (i2 + i3)

- i2 + i3 = 5 + (-3) = 2

- i1 / 2 = 2 / 2 = 1.0 (phép chia thường luôn trả về float)

- Kết quả: 1.0

(d) i1 // (i2 + i3)

- i2 + i3 = 5 + (-3) = 2

- i1 // 2 = 2 // 2 = 1 (phép chia nguyên)

- Kết quả: 1

(e) i1 / i2 + i3

- i1 / i2 = 2 / 5 = 0.4 , 0.4 + i3 = 0.4 + (-3) = -2.6, Kết quả: -2.6

(f) i1 // i2 + i3

- i1 // i2 = 2 // 5 = 0 (vì 2 chia 5 được 0.4, lấy phần nguyên là 0), 0 + i3 = 0 + (-3) = -3, Kết quả: -3

(g) 3 + 4 + 5 / 3

- 5 / 3 ≈ 1.666..., 3 + 4 + 1.666... = 8.666..., Kết quả: 8.666... (cụ thể là 8.666666666666666)

(h) 3 + 4 + 5 // 3

- 5 // 3 = 1 (chia nguyên), 3 + 4 + 1 = 8, Kết quả: 8

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

- 3 + 4 + 5 = 12, 12 / 3 = 4.0, Kết quả: 4.0

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

- 3 + 4 + 5 = 12, 12 // 3 = 4, Kết quả: 4

(k) d1 + (d2 \* d3)

- d2 \* d3 = 5.0 \* (-0.5) = -2.5, d1 + (-2.5) = 2.0 + (-2.5) = -0.5, Kết quả: -0.5

(l) d1 + d2 \* d3 (giống k vì có ưu tiên nhân trước)

- d2 \* d3 = 5.0 \* (-0.5) = -2.5, d1 + (-2.5) = 2.0 + (-2.5) = -0.5, Kết quả: -0.5

(m) d1 / d2 - d3

- d1 / d2 = 2.0 / 5.0 = 0.4, 0.4 - d3 = 0.4 - (-0.5) = 0.4 + 0.5 = 0.9, Kết quả: 0.9

(n) d1 / (d2 - d3)

- d2 - d3 = 5.0 - (-0.5) = 5.0 + 0.5 = 5.5, d1 / 5.5 ≈ 2.0 / 5.5 ≈ 0.363636..., Kết quả: 0.363636... (cụ thể là 0.36363636363636365)

(o) d1 + d2 + d3 / 3

- d3 / 3 = -0.5 / 3 ≈ -0.166666..., d1 + d2 + (-0.166666...) = 2.0 + 5.0 - 0.166666... = 6.833333..., Kết quả: 6.833333... (cụ thể là 6.833333333333333)

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

- d1 + d2 + d3 = 2.0 + 5.0 + (-0.5) = 6.5, 6.5 / 3 ≈ 2.166666..., Kết quả: 2.166666... (cụ thể là 2.1666666666666665)

(q) d1 + d2 + (d3 / 3) (giống o)

- d3 / 3 = -0.5 / 3 ≈ -0.166666..., d1 + d2 + (-0.166666...) = 2.0 + 5.0 - 0.166666... = 6.833333..., Kết quả: 6.833333...

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

- d1 + d2 = 2.0 + 5.0 = 7.0, d1 - d3 = 2.0 - (-0.5) = 2.0 + 0.5 = 2.5, 3 \* 7.0 \* 2.5 = 21.0 \* 2.5 = 52.5, Kết quả: 52.5

*Câu 10:*

(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