


<b>Giảng viên ra đề:</b> (Chữ ký và Họ tên)	(Ngày ra đề)	<b>Người phê duyệt:</b> (Chữ ký và họ tên)	(Ngày duyệt đề)
--	--------------	---	-----------------

33/40 → 8.125

 <b>TRƯỜNG ĐH BÁCH KHOA - ĐHQG-HCM</b> <b>KHOA KH &amp; KT MÁY TÍNH</b>	<b>KT GIỮA KỲ</b>		Học kỳ / Năm học	2	2022-2023
			Ngày thi	06-03-2023	
	Môn học	Nguyên lý ngôn ngữ lập trình			
	Mã môn học	CO3005			
	Thời lượng	70 phút	Mã đề	2210	

**Ghi chú:**

- Sinh viên được phép sử dụng một tờ A4 tài liệu tham khảo (viết tay) và máy tính cầm tay.
- Sinh viên làm bài trên phiếu trả lời trắc nghiệm.
- Các câu hỏi chỉ có 1 đáp án đúng hoặc không có đáp án đúng.
- Nếu không có đáp án đúng, sinh viên chọn đáp án E.
- Sinh viên nộp đề cùng với phiếu trả lời trắc nghiệm sau khi kiểm tra.

✓ **Câu 1.** [L.O.1.1] Một danh hiệu trong ngôn ngữ lập trình Ruby là một chuỗi các ký tự số, chữ thường và dấu gạch dưới. Nó phải được bắt đầu bằng một dấu gạch dưới hoặc một ký tự chữ thường. Chọn một biểu thức chính quy phù hợp để mô tả danh hiệu nói trên?

- (A) `[a-z0-9_]+` (B) `[a-zA-Z0-9_]+` (C) `[a-z_] [a-z0-9_]*` (D) `[0-9_] [a-z0-9_]+`

✓ **Câu 2.** [L.O.2.1] Hãy cho biết kết quả xuất ra màn hình của đoạn mã (trong ngôn ngữ Python) sau:

```
def square(x): return x ** 2
def double(x): return x * 2
numbers = [1, 2, 3, 4, 5]
result = map(square, filter(lambda x: x % 2 == 0, map(double, numbers)))
print(list(result))
```

- (A) `[4, 16]` (B) `[4, 16, 36, 64, 100]` (C) `[1, 4, 9, 16, 25]` (D) `[2, 4, 6, 8, 10]`

✓ **Câu 3.** [L.O.1.2] Để liệt kê một tập hợp số nguyên, người ta sử dụng một danh sách các phần tử cách nhau bởi một dấu CM. Các phần tử đó có thể là một số nguyên IL hoặc một đoạn số nguyên. Đoạn số nguyên là hai số nguyên cách nhau bởi một dấu TP. Dạng EBNF của về phải luật sinh tập hợp số nguyên kể trên là

- (A) `IL CM IL* | (IL TP IL)*` (B) `(IL | IL TP IL) (CM (IL | IL TP IL))*`  
 (C) `IL CM (IL TP IL)* | (IL TP IL) CM IL*` (D) `IL CM (IL | IL TP IL)* | CM IL+`

✓ **Câu 4.** [L.O.1.2] Hãy cho biết văn phạm nào sau đây bị nhập nhằng

- (A)  $S \rightarrow \epsilon | aSbS$  (B)  $S \rightarrow \epsilon | aSa|bSb$  (C)  $S \rightarrow AB|BA, A \rightarrow \epsilon | aA, B \rightarrow \epsilon | bB$   
 (D)  $S \rightarrow \epsilon | a|b|aSa|bSb$

✓ **Câu 5.** [L.O.1.1] Chọn biểu thức chính qui chấp nhận ít nhất tất cả các chuỗi trong tập MATCH nhưng không chấp nhận bất kỳ chuỗi nào trong tập SKIP sau:

MATCH = {Cho, chi, Chung, Che, Chan }  
 SKIP = {Tro, Ching, Chu, Tre, Tran}

- (A) `[cCT][hr][aeuio]n?g?` (B) `[cC]h[aoiue]n?g?` (C) `[Cc]h[oie]lCh[au]ng?` (D) `(Clc)h(olile)lCh(alu)n?g?`

✓ **Câu 6.** [L.O.2.1] Cho một danh sách bao gồm các phần tử có thể lồng nhau trong một danh sách, ví dụ:

`nested_lst = [1, 2, [3, 4, [5, 6], 7], 8, [9]]`

Hàm `flatten` có thể nhận danh sách trên và trả ra danh sách ở dạng làm phẳng như `[1, 2, 3, 4, 5, 6, 7, 8, 9]`. Thân của `flatten` là

- (A) `return reduce(lambda prev, curr: prev + (flatten(curr) if type(curr) is list else curr), lst, [])`  
 (B) `return reduce(lambda prev, curr: prev + [curr], lst, [])`  
 (C) `return reduce(lambda prev, curr: prev + (flatten(curr) if type(curr) is list else [curr]), lst, [])`  
 (D) `return reduce(lambda prev, curr: prev + curr, lst, [])`

✓ **Câu 7.** [L.O.1.2] Một danh sách `lcase` bao gồm các `case` nối tiếp nhau và có thể rỗng. Một `case` bao gồm từ khóa `CASE`, sau đó là một biểu thức `exp` và một dấu `CL` và kết thúc bởi danh sách các phát biểu `stmtlist`. Dạng EBNF của vế phải luật sinh `lcase` là

- (A) `CASE (exp CL stmtlist)*?` (B) `(CASE exp* CL stmtlist)*?`  
(C) `(CASE exp CL stmtlist)*` (D) `CASE exp* CL stmtlist?`

✓ **Câu 8.** [L.O.2.1] Cho đoạn mã sau trong ngôn ngữ Python:

```
class A:
    def who_am_i(self):
        print("A")
class B(A):
    def who_am_i(self):
        print("B")
        super().who_am_i()
class C(A):
    def who_am_i(self):
        print("C")
        super().who_am_i()
class D(B, C):
    def who_am_i(self):
        print("D")
        super().who_am_i()
class E(C, B):
    def who_am_i(self):
        print("E")
        super().who_am_i()
class F(E, D):
    def who_am_i(self):
        print("F")
        super().who_am_i()

f = F()
f.who_am_i()
```

Kết quả in ra màn hình của đoạn mã trên là

- (A) FDBCEA (B) FBDCEA (C) FDBCAE (D) FBCDEA

✓ **Câu 9.** [L.O.2.1] Cho các khai báo trong một ngôn ngữ lập trình hướng đối tượng với kiểm tra kiểu tĩnh:

```
class A { def foo() = print("a") }
class B extends A { } // B is a subclass of A
class C extends A { override def foo() = print("c") } // C is a subclass of A
class D extends B { override def foo() = print("d") } // D is a subclass of B
```

Biết rằng, biến `b` được khai báo kiểu `B` và đang tham chiếu đến một đối tượng nào đó. Cho một số nhận định về kết quả được in ra khi gọi `b.foo()`

- (a) `c` (nếu `b` đang tham chiếu đến một đối tượng `C`)  
(b) `d` (nếu `b` đang tham chiếu đến một đối tượng `D`)  
(c) `a` (nếu `b` đang tham chiếu đến một đối tượng `A`)  
(d) `a` (nếu `b` đang tham chiếu đến một đối tượng `B`)

Số nhận định đúng là

- (A) 1 (B) 2 (C) 3 (D) 0

✓ **Câu 10.** [L.O.2.1] Cho đoạn mã sau trên Python:

```
x, y = 0, 4
while x < 3:
    x += 1
else: print(y)
```

Sau khi thực thi đoạn mã trên,

- (A) Giá trị 4 được in ra (B) Giá trị 3 được in ra  
(C) Không có giá trị nào được in ra (D) Đoạn mã trên gây ra lỗi văn phạm

✓ **Câu 11.** [L.O.1.1] Cho một mô tả từ vựng được định nghĩa trong ANTLR4 như sau:

```

FLOAT_CONSTANT: DIGIT_SEQUENCE EXPONENT? FLOAT_SUFFIX?;

fragment DIGIT_SEQUENCE: DIGIT+ ( '.' DIGIT+ )?;
fragment EXPONENT: ( 'e' | 'E' ) ( '+' | '-' )? DIGIT+;
fragment FLOAT_SUFFIX: ( 'f' | 'F' | 'l' | 'L' );
fragment DIGIT: [0-9];

```

Chuỗi nào sau đây là chuỗi nhập đúng cho token `FLOAT_CONSTANT` và đồng thời có giải thích đúng:

- (A) 0.0001E-2f, trong đó E-2 được tạo thành từ `EXPONENT`
- (B) 6.02e23L, trong đó e23L được tạo thành từ `EXPONENT`
- (C) 0.123\_456 và không có thành phần `FLOAT_SUFFIX`
- (D) 123.456E+7F, trong đó 123.456E+7 được tạo thành từ `DIGIT_SEQUENCE`

✓ **Câu 12.** [L.O.1.2] Cho một tập luật sinh như sau:

$S \rightarrow aSb|T$

$T \rightarrow cTd|\epsilon$

Một quá trình dẫn xuất trái nhất là

- (A)  $S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aacTdbb \Rightarrow aaccTddb \Rightarrow aaccddb$
- (B)  $S \Rightarrow aSb \Rightarrow aTb \Rightarrow acTdb \Rightarrow acedb$
- (C) Các lựa chọn khác đều đúng
- (D)  $S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aaaSbbb \Rightarrow aaaaSbbbb \Rightarrow aaaaTbbbb \Rightarrow aaaaabbbb$

✓ **Câu 13.** [L.O.2.1] Câu nào sau đây miêu tả đúng về hàm bậc cao (high-order function)?

- (A) Là một hàm có thể nhận đối số là một hàm khác
- (B) Là một hàm luôn phải nhận đầy đủ đối số là một mảng và một hàm để xử lý
- (C) Là một hàm sử dụng đệ quy để lặp qua một tập giá trị
- (D) Là một hàm luôn trả ra một hàm khác

✓ **Câu 14.** [L.O.2.1] Nhận định nào sau đây là đúng về MRO (Method Resolution Order) trong Python?

- (A) MRO xác định thứ tự các module nhập vào trên một file Python.
- (B) MRO xác định thứ tự thực thi một tập lệnh Python.
- (C) MRO xác định thứ tự tìm kiếm một phương thức hoặc thuộc tính trong một thứ tự kế thừa lớp.
- (D) MRO xác định thứ tự mà trình biên dịch Python biên dịch mã nguồn.

✓ **Câu 15.** [L.O.3.1] Trong các giai đoạn của hiện thực ngôn ngữ bằng phương pháp lai, giai đoạn nào sau đây nhận vào một cây phân tích cú pháp (hoặc cây cú pháp trừu tượng) và ném ra các lỗi về danh hiệu, hệ thống kiểu dữ liệu?

- (A) Kiểm tra tĩnh
- (B) Phân tích cú pháp
- (C) Phân tích từ vựng
- (D) Sinh mã trung gian

✓ **Câu 16.** [L.O.2.1] Theo quy ước trên ngôn ngữ Python, thuộc tính `ex` mang tính `protected` sẽ được khai báo bằng cách:

- (A) Đặt tên của thuộc tính là `_ex`
- (B) Đặt tên của thuộc tính là `__ex`
- (C) Khai báo `ex: private` trong phương thức `__init__`
- (D) Khai báo `ex` với một annotation `@private_attr()`

✓ **Câu 17.** [L.O.1.2] Cho văn phạm phi ngữ cảnh G với tập ký hiệu kết thúc là {ADD, MINUS, MUL, DIV, LB, RB}, tập ký hiệu không kết thúc là {exp, term, fact}, ký hiệu bắt đầu là exp, và tập luật sinh là:

$exp \rightarrow term \text{ MINUS } exp \mid term$

$term \rightarrow fact \text{ DIV } term \mid fact$

$fact \rightarrow fact \text{ ADD } factor \mid fact \text{ MUL } factor \mid factor$

$factor \rightarrow LB \text{ exp } RB \mid INT$

Cho INT là token của các số nguyên, ADD của phép cộng, MINUS của phép trừ, MUL của phép nhân và DIV của phép chia nguyên, LB của '(' và RB của ')'. Hãy xác định độ ưu tiên và tính kết hợp của các phép toán để từ đó tính toán giá trị của chuỗi nhập sau  $100 - 4 / 4 + 2 * 3 / 9 - 10$ ?

- (A) 108
- (B) 109
- (C) 90
- (D) 89

✓ **Câu 18.** [L.O.2.1] Cho lớp A là lớp cha của lớp B trên một ngôn ngữ lập trình hướng đối tượng kiểm tra kiểu tĩnh như Java, Scala, với biến a và b có kiểu tương ứng là A và B. Cho hai phát biểu gán sau:

`a = new B();` // phát biểu 1

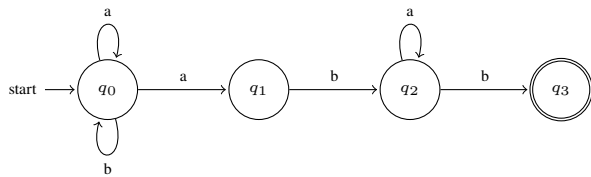
`b = new A();` // phát biểu 2

- (A) Phát biểu 1 đúng và phát biểu 2 sai
- (B) Phát biểu 1 sai và phát biểu 2 đúng
- (C) Cả hai phát biểu 1 và 2 đều đúng
- (D) Cả hai phát biểu 1 và 2 đều sai

✓ **Câu 19.** [L.O.2.1] Một biến trên Python sẽ

- (A) giữ con trỏ đến một đối tượng
- (B) nhận một giá trị để lưu giữ
- (C) phải cần khai báo kiểu
- (D) phải khai báo trước khi gán trị

**Câu 20.** [L.O.3.1] Cho một automata như sau:



Biểu thức chính quy tương đương của automata trên là

- ☒ (A)  $b^*a^*aba+b$ 
☐ (B)  $(ab)^*aba^*b$ 
☐ (C)  $a^*b^*aba^*b$ 
☐ (D)  $[ab]^*aba^*b$

**Câu 21.** [L.O.1.2] Về phải luật sinh của `ui_list` mô tả một danh sách gồm ít nhất một phần tử `UI` được viết dưới dạng BNF là

- ☒ (A) `UI ui_list |`
☐ (B) `UI ui_list`  
☐ (C) `ui_list | UI`
☐ (D) `UI ui_list | UI`

Đoạn mô tả sau áp dụng cho các câu 22–23:

Giả sử một chương trình trên Python có các lớp A, B, C, D(A,B), E(C,A) và F(D,E,B). Trong các lớp này, chỉ có hai lớp A và C có khai báo cho phương thức có tên là `method1`. Trong các lớp này, chỉ có lớp D có một khai báo cho phương thức có tên là `meth` và có thân là một lệnh gọi `self.method1()`.

**Câu 22.** [L.O.2.1] MRO của lớp F là:

- ☒ (A) [F,D,E,C,A,B,object]
☐ (B) [F,D,A,B,E,C,object]
☐ (C) [F,D,A,E,C,B,object]
☐ (D) [F,E,C,D,A,B,object]

**Câu 23.** [L.O.2.1] Hãy cho biết `F().meth()` và `D().meth()` sẽ gọi phương thức `method1` của lớp nào?

- ☒ (A) `F().meth()` gọi `method1` trong lớp C và `D().meth()` gọi `method1` trong lớp A  
☐ (B) `F().meth()` và `D().meth()` đều gọi `method1` trong C  
☐ (C) `F().meth()` và `D().meth()` đều gọi `method1` trong A  
☐ (D) `F().meth()` gọi `method1` trong lớp A và `D().meth()` gọi `method1` trong lớp C

**Câu 24.** [L.O.1.2] Cho đoạn mã sau trong ANTLR

```

decl: ID decl_tail;
decl_tail: CM decl | CL ID CM;

```

Về phải nào sau đây phù hợp cho luật sinh `decl` để tương đương đoạn mã trên?

- ☒ (A)  $(ID\ CM)^* (CL\ ID\ CM)?$ 
☐ (B)  $(ID\ CM)^* CL\ ID\ CM$   
☐ (C)  $ID\ CM\ ID (CL\ ID)^* CM$ 
☐ (D)  $ID (CM\ ID)^* CL\ ID\ CM$

**Câu 25.** [L.O.1.2] Ký hiệu `list` mô tả một danh sách (có thể rỗng) các phần tử `a` được viết cách nhau bởi một dấu `C`. Văn phạm cho `list` được viết trên ANTLR như sau:

```

list: elist | (1);
elist: a C elist | (2);

```

(1) và (2) lần lượt là (biết rằng ký tự `_` đại diện cho rỗng)

- ☒ (A) `_` và `a`
☐ (B) `C a` và `_`
☐ (C) `_` và `_`
☐ (D) `_` và `C`

**Câu 26.** [L.O.1.2] Câu nào sau đây đúng về ngôn ngữ sinh ra bởi văn phạm phi ngữ cảnh sau?

$S \rightarrow AB, A \rightarrow aA|\epsilon, B \rightarrow bB|\epsilon$

- ☐ (A) Ngôn ngữ này gồm tất cả các chuỗi trên ngôn ngữ với tập ký tự  $\{a, b\}$   
☐ (B) Ngôn ngữ này gồm tất cả các chuỗi dạng  $a^m b^n$ , trong đó  $m, n$  là số nguyên không âm.  
☐ (C) Ngôn ngữ này gồm chỉ chuỗi rỗng  
☐ (D) Ngôn ngữ này gồm tất cả các chuỗi dạng  $a^n b^n$ , trong đó  $n$  là số nguyên không âm.

**Câu 27.** [L.O.2.1] Khi lập trình bằng Python, để đưa ký hiệu xuống hàng trực tiếp vào trong chuỗi bằng cách nhấn phím Enter thay vì dùng chuỗi escape `\n`, chuỗi phải được đặt trong:

- ☐ (A) một cặp 3 dấu nháy kép
☐ (B) một cặp dấu nháy kép
☐ (C) một cặp dấu nháy đơn  
☐ (D) phải dùng escape, không thể nhập trực tiếp

✓ **Câu 28.** [L.O.2.1] Cho đoạn mã sau trên Python:

```
y = 0
for x in range(5):
    if x == 5: break
    y += 1
else: print(y)
```

Sau khi thực thi đoạn mã trên,

- (A) Giá trị 5 được in ra (B) Giá trị 4 được in ra  
(C) Không có giá trị nào được in ra (D) Đoạn mã trên gây ra lỗi văn phạm

✓ **Câu 29.** [L.O.1.1] Cho một mô tả từ vựng được định nghĩa trong ANTLR4 như sau:

```
UNIVERSE: A* S A A A A A+;
fragment A: D | C | S;
fragment D: [0-9];
fragment C: [a-zA-Z];
fragment S: [@!%*#?&];
```

Chuỗi nhập ứng với token UNIVERSE có tính chất nào sau đây?

- (A) Có ít nhất 4 ký tự và phải có chứa ít nhất một ký tự đặc biệt (@!%\*#?&)  
(B) Có ít nhất 6 ký tự và phải có chứa ít nhất một ký tự đặc biệt (@!%\*#?&)  
(C) Có nhiều nhất 8 ký tự và phải có chứa ký tự chữ thường hoặc chữ số  
(D) Có ít nhất 6 ký tự và khi chứa ký tự chữ thường thì không chứa ký tự chữ hoa và ngược lại

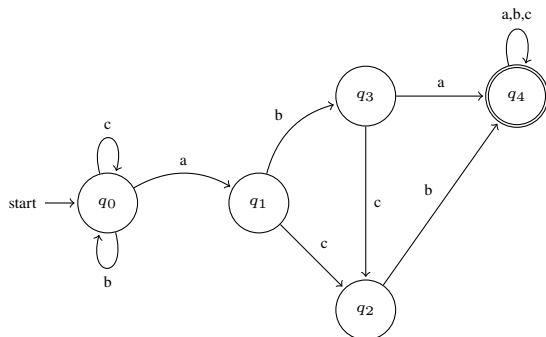
✓ **Câu 30.** [L.O.1.1] Cho biểu thức chính quy  $a[^{abc}]^*c$  và các chuỗi nhập gồm adc, abbc, ayyyyyyyyyc, abc, aabc, axc. Số chuỗi nhập thỏa mãn biểu thức chính quy là

- (A) 1 (B) 5 (C) 3 (D) 2

✓ **Câu 31.** [L.O.1.1] Chọn biểu thức chính qui tương đương với biểu thức chính qui sau:  $(alb)^*(abblb)a$

- (A)  $(b^*a^*)(ab)?ba$  (B)  $[alb]^*[abblb]a$  (C)  $[ab]^*[ab]?ba$  (D)  $[ab]^*(ab)+ba$

✓ **Câu 32.** [L.O.3.1] Cho một automata như sau:



Chuỗi nhập nào sau đây được chấp nhận bởi automata trên:

- (A) bccbabcbaab (B) abcaabcbaaa (C) baccbacbcca (D) bcabcaabcab

Áp dụng đoạn mã sau trong ngôn ngữ Python cho các câu 33–34:

```
result = (lst[0] * 2) + func(x, y) - (lst[-1] if lst[1] >= -1.2 else lst[2]) % 5 # cal result
```

✓ **Câu 33.** [L.O.2.1]  
Số token được phân tích từ vựng trả về khi phân tích từ vựng cho chuỗi trên là

- (A) 38 (B) 43 (C) 40 (D) 45

✓ **Câu 34.** [L.O.2.1]  
Chuỗi lexeme của token thứ 25 là:

- (A) -1 (B) ] (C) if (D) lst

✓ **Câu 35.** [L.O.1.1] Để biểu diễn dạng lũy thừa  $a^b$  với  $a$  và  $b$  là các số nguyên dương trong LaTeX, người ta viết cơ số  $a$ , tiếp theo là dấu mũ, tiếp theo là phân cơ số  $b$ . Nếu cơ số  $b$  có từ hai chữ số trở lên phải được bao trong dấu ngoặc nhọn. Nếu cơ số  $b$  chỉ có một chữ số, việc sử dụng cặp ngoặc là tùy chọn.  
Trong ANTLR4, cho các fragment mô tả ký tự số là DIGIT, dấu mũ là HAT, cặp ngoặc nhọn mở và đóng lần lượt là LP và RP. Phần mô tả nào sau đây thể hiện được dạng lũy thừa nói trên

- (A) DIGIT+ HAT (DIGIT | LP DIGIT DIGIT+ RP) (B) DIGIT+ HAT (LP (DIGIT | DIGIT+) RP)  
(C) DIGIT+ HAT DIGIT | LP DIGIT+ RP (D) DIGIT+ HAT (DIGIT | LP DIGIT+ RP)

✓ **Câu 36.** [L.O.1.2] Một văn phạm phi ngữ cảnh (context-free grammar) có thể bao gồm:

- Ⓐ Các biểu thức chính quy để mô tả ký hiệu không kết thúc      Ⓑ Tập các ký hiệu kết thúc, tập luật sinh  
Ⓒ Một ký hiệu bắt đầu và biểu thức chính quy mô tả nó      Ⓓ Chỉ có duy nhất một tập luật sinh thể hiện thứ tự của các từ loại

✓ **Câu 37.** [L.O.1.1] Trong ngôn ngữ X, lập trình viên có thể viết các hằng số nguyên không chính xác bằng cách thay thế các ký tự số chưa biết bằng dấu # với ký tự liền trước nó phải là ký tự số. Hằng số nguyên không chính xác phải được bắt đầu bằng ít nhất hai ký tự số. Ví dụ như 12#345, 123#45# là hợp lệ nhưng 123##45 là không hợp lệ. Trong ANTLR4, cho các fragment mô tả ký tự số là DIGIT, dấu # là SHARP. Phần mô tả nào sau đây thể hiện cho hằng số nguyên không chính xác trong ngôn ngữ X

- Ⓐ DIGIT (DIGIT SHARP?)+      Ⓑ DIGIT (DIGIT SHARP?)\*      Ⓒ DIGIT (DIGIT SHARP)+      Ⓓ DIGIT (DIGIT\* SHARP)\*

**Câu 38.** [L.O.2.1] Cho đoạn mã sau trên Python:

```
def square(f):  
    def wrap(x):  
        return f(x)**2  
    return wrap  
  
(1)  
def double(x):  
    return x * 2  
print(double(3)) # result is 36
```

Cần điền gì vào vị trí (1) để kết quả print(double(3)) là 36?

- Ⓐ @square      Ⓑ square      Ⓒ x = square(6)      Ⓓ #square

✓ **Câu 39.** [L.O.1.2] Phát biểu đa gán cho phép gán nhiều tên ID (về trái, cách nhau bởi một dấu CM) cho nhiều biểu thức exp (về phải, cách nhau bởi một dấu CM) với về trái và về phải cách nhau bởi một dấu bằng EQ. Hãy dùng assignment để mô tả cho phát biểu đa gán và viết về phải của assignment sao cho số lượng tên và biểu thức phải bằng nhau và có ít nhất một tên và một biểu thức trong phát biểu đa gán:

- Ⓐ (ID (CM ID)\* EQ exp (CM exp)\*      Ⓑ ID CM assignment CM exp | ID EQ exp  
Ⓒ ID CM assignment CM exp | EQ      Ⓓ ID EQ exp CM assignment | ID EQ exp

**Câu 40.** [L.O.2.1] Một chương trình được viết trong ngôn ngữ X có nội dung như sau:

- Cho người dùng nhập giá trị vào biến **a** từ bàn phím.
- Thực hiện một vòng lặp **a** lần với thân vòng lặp có nội dung:
  - Tính toán giá trị biến **b**.
  - Nếu biến **b** lớn hơn 5 thì in ra giá trị biến **b** và thoát khỏi vòng lặp.
  - Thực hiện phát biểu in **a** nhưng có chứa lỗi ngữ pháp.

Trong quá trình thực thi, người ta thấy có lúc thì chương trình thực hiện thành công có lúc lại ném ra lỗi. Phương pháp nào đã được sử dụng để hiện thực ngôn ngữ X nói trên?

- Ⓐ Trình thông dịch (Pure interpreter)      Ⓑ Trình biên dịch (Compiler)  
Ⓒ Phương pháp lai (Hybrid implementation)  
Ⓓ Phương pháp lai có sử dụng trình dịch tức thời (Hybrid implementation with just-in-time compiler)

## ĐÁP ÁN

Câu 1. C

Câu 2. B

Câu 3. B

Câu 4. C

Câu 5. C

Câu 6. C

Câu 7. C

Câu 8.

Câu 9. B

Câu 10. A

Câu 11. A

Câu 12. D

Câu 13. A

Câu 14. C

Câu 15. A

Câu 16. A

Câu 17. A

Câu 18. A

Câu 19. A

Câu 20. D

Câu 21. D

Câu 22. A

Câu 23. A

Câu 24. D

Câu 25. A

Câu 26. B

Câu 27. A

Câu 28. A

Câu 29. B

Câu 30. C

Câu 31. A

Câu 32. A

Câu 33. C

Câu 34. C

Câu 35. D

Câu 36. B


Câu 37. A

Câu 38. A

Câu 39. B

Câu 40. A

<b>Giảng viên ra đề:</b> (Chữ ký và Họ tên)	(Ngày ra đề)	<b>Người phê duyệt:</b> (Chữ ký và họ tên)	(Ngày duyệt đề)
--	--------------	---	-----------------

 <b>TRƯỜNG ĐH BÁCH KHOA - ĐHQG-HCM</b> <b><u>KHOA KH &amp; KT MÁY TÍNH</u></b>	<b>KT GIỮA KỲ</b>		Học kỳ / Năm học		2	2022-2023
			Ngày thi		06-03-2023	
	Môn học	Nguyên lý ngôn ngữ lập trình				
	Mã môn học	CO3005				
	Thời lượng	70 phút	Mã đề	2211		
<b>Ghi chú:</b> <ul style="list-style-type: none"><li>- Sinh viên được phép sử dụng một tờ A4 tài liệu tham khảo (viết tay) và máy tính cầm tay.</li><li>- Sinh viên làm bài trên phiếu trả lời trắc nghiệm.</li><li>- Các câu hỏi chỉ có 1 đáp án đúng hoặc không có đáp án đúng.</li><li>- Nếu không có đáp án đúng, sinh viên chọn đáp án E.</li><li>- Sinh viên nộp đề cùng với phiếu trả lời trắc nghiệm sau khi kiểm tra.</li></ul>						

**Câu 1.** [L.O.1.2] Một văn phạm phi ngữ cảnh (context-free grammar) có thể bao gồm:

- ☐ (A) Chỉ có duy nhất một tập luật sinh thể hiện thứ tự của các từ loại
 ☐ (B) Các biểu thức chính quy để mô tả ký hiệu không kết thúc
 ☐ (C) Tập các ký hiệu kết thúc, tập luật sinh
 ☐ (D) Một ký hiệu bắt đầu và biểu thức chính quy mô tả nó

**Câu 2.** [L.O.1.2] Để liệt kê một tập hợp số nguyên, người ta sử dụng một danh sách các phần tử cách nhau bởi một dấu CM. Các phần tử đó có thể là một số nguyên IL hoặc một đoạn số nguyên. Đoạn số nguyên là hai số nguyên cách nhau bởi một dấu TP. Dạng EBNF của về phải luật sinh tập hợp số nguyên kể trên là

- ☐ (A)  $IL\ CM\ (IL\ |\ IL\ TP\ IL)^* \mid CM\ IL^+$ 
☐ (B)  $IL\ CM\ IL^* \mid (IL\ TP\ IL)^*$ 
☐ (C)  $(IL\ |\ IL\ TP\ IL)\ (CM\ (IL\ |\ IL\ TP\ IL))^*$ 
☐ (D)  $IL\ CM\ (IL\ TP\ IL)^* \mid (IL\ TP\ IL)\ CM\ IL^*$

**Câu 3.** [L.O.1.1] Cho một mô tả từ vựng được định nghĩa trong ANTLR4 như sau:

```

FLOAT_CONSTANT: DIGIT_SEQUENCE EXPONENT? FLOAT_SUFFIX?;

fragment DIGIT_SEQUENCE: DIGIT+ ('.' DIGIT+)?;
fragment EXPONENT: ('e' | 'E') ('+' | '-')? DIGIT+;
fragment FLOAT_SUFFIX: ('f' | 'F' | 'l' | 'L');
fragment DIGIT: [0-9];
  
```

Chuỗi nào sau đây là chuỗi nhập đúng cho token `FLOAT_CONSTANT` và đồng thời có giải thích đúng:

- ☐ (A) 123.456E+7F, trong đó 123.456E+7 được tạo thành từ `DIGIT_SEQUENCE`  
☐ (B) 0.0001E-2f, trong đó E-2 được tạo thành từ `EXPONENT`  
☐ (C) 6.02e23L, trong đó e23L được tạo thành từ `EXPONENT`  
☐ (D) 0.123\_456 và không có thành phần `FLOAT_SUFFIX`

Đoạn mô tả sau áp dụng cho các câu 4–5:

Giả sử một chương trình trên Python có các lớp A, B, C, D(A,B), E(C,A) và F(D,E,B). Trong các lớp này, chỉ có hai lớp A và C có khai báo cho phương thức có tên là `method1`. Trong các lớp này, chỉ có lớp D có một khai báo cho phương thức có tên là `meth` và có thân là một lệnh gọi `self.method1()`.

**Câu 4.** [L.O.2.1] MRO của lớp F là:

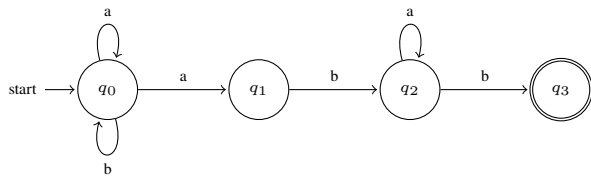
- ☐ (A) [F,E,C,D,A,B,object]
 ☐ (B) [F,D,E,C,A,B,object]
 ☐ (C) [F,D,A,B,E,C,object]
 ☐ (D) [F,D,A,E,C,B,object]

**Câu 5.** [L.O.2.1] Hãy cho biết `F().meth()` và `D().meth()` sẽ gọi phương thức `method1` của lớp nào?

- ☐ (A) `F().meth()` gọi `method1` trong lớp A và `D().meth()` gọi `method1` trong lớp C  
☐ (B) `F().meth()` gọi `method1` trong lớp C và `D().meth()` gọi `method1` trong lớp A  
☐ (C) `F.meth()` và `D().meth()` đều gọi `method1` trong C  
☐ (D) `F.meth()` và `D().meth()` đều gọi `method1` trong A



**Câu 6.** [L.O.3.1] Cho một automata như sau:



Biểu thức chính quy tương đương của automata trên là

- Ⓐ  $[ab]^*aba^*b$       Ⓑ  $b^*a^*aba^*b$       Ⓒ  $(ab)^*aba^*b$       Ⓓ  $a^*b^*aba^*b$

**Câu 7.** [L.O.2.1] Theo quy ước trên ngôn ngữ Python, thuộc tính `ex` mang tính `protected` sẽ được khai báo bằng cách:

- Ⓐ Khai báo `ex` với một annotation `@private_attr()`      Ⓑ Đặt tên của thuộc tính là `_ex`  
 Ⓒ Đặt tên của thuộc tính là `__ex`      Ⓓ Khai báo `ex: private` trong phương thức `__init__`

**Câu 8.** [L.O.2.1] Cho đoạn mã sau trong ngôn ngữ Python:

```

class A:
    def who_am_i(self):
        print("A")
class B(A):
    def who_am_i(self):
        print("B")
        super().who_am_i()
class C(A):
    def who_am_i(self):
        print("C")
        super().who_am_i()
class D(B, C):
    def who_am_i(self):
        print("D")
        super().who_am_i()
class E(C, B):
    def who_am_i(self):
        print("E")
        super().who_am_i()
class F(E, D):
    def who_am_i(self):
        print("F")
        super().who_am_i()
f = F()
f.who_am_i()
  
```

Kết quả in ra màn hình của đoạn mã trên là

- Ⓐ FBCDEA      Ⓑ FDBCEA      Ⓒ FBDCEA      Ⓓ FDBCAE

**Câu 9.** [L.O.2.1] Cho đoạn mã sau trên Python:

```

def square(f):
    def wrap(x):
        return f(x)**2
    return wrap
(1)
def double(x):
    return x * 2
print(double(3)) # result is 36
  
```

Cần điền gì vào vị trí (1) để kết quả `print(double(3))` là 36?

- Ⓐ `#square`      Ⓑ `@square`      Ⓒ `square`      Ⓓ `x = square(6)`

**Câu 10.** [L.O.2.1] Hãy cho biết kết quả xuất ra màn hình của đoạn mã (trong ngôn ngữ Python) sau:

```

def square(x): return x ** 2
def double(x): return x * 2
numbers = [1, 2, 3, 4, 5]
result = map(square, filter(lambda x: x % 2 == 0, map(double, numbers)))
print(list(result))
  
```

- Ⓐ  $[2, 4, 6, 8, 10]$       Ⓑ  $[4, 16]$       Ⓒ  $[4, 16, 36, 64, 100]$       Ⓓ  $[1, 4, 9, 16, 25]$

**Câu 11.** [L.O.3.1] Trong các giai đoạn của hiện thực ngôn ngữ bằng phương pháp lai, giai đoạn nào sau đây nhận vào một cây phân tích cú pháp (hoặc cây cú pháp trừu tượng) và ném ra các lỗi về danh hiệu, hệ thống kiểu dữ liệu?

- Ⓐ Sinh mã trung gian      Ⓑ Kiểm tra tĩnh      Ⓒ Phân tích cú pháp      Ⓓ Phân tích từ vựng

**Câu 12.** [L.O.2.1] Nhận định nào sau đây là đúng về MRO (Method Resolution Order) trong Python?

- (A) MRO xác định thứ tự mà trình biên dịch Python biên dịch mã nguồn.
- (B) MRO xác định thứ tự các module nhập vào trên một file Python.
- (C) MRO xác định thứ tự thực thi một tập lệnh Python.
- (D) MRO xác định thứ tự tìm kiếm một phương thức hoặc thuộc tính trong một thứ tự kế thừa lớp.

**Câu 13.** [L.O.2.1] Cho đoạn mã sau trên Python:

```
y = 0
for x in range(5):
    if x == 5: break
    y += 1
else: print(y)
```

Sau khi thực thi đoạn mã trên,

- (A) Đoạn mã trên gây ra lỗi văn phạm
- (B) Giá trị 5 được in ra
- (C) Giá trị 4 được in ra
- (D) Không có giá trị nào được in ra

**Câu 14.** [L.O.1.2] Phát biểu đa gán cho phép gán nhiều tên ID (về trái, cách nhau bởi một dấu CM) cho nhiều biểu thức exp (về phải, cách nhau bởi một dấu CM) với về trái và về phải cách nhau bởi một dấu bằng EQ. Hãy dùng assignment để mô tả cho phát biểu đa gán và viết về phải của assignment sao cho số lượng tên và biểu thức phải bằng nhau và có ít nhất một tên và một biểu thức trong phát biểu đa gán:

- (A) ID EQ exp CM assignment | ID EQ exp
- (B) (ID (CM ID)\* EQ exp (CM exp)\*)
- (C) ID CM assignment CM exp | ID EQ exp
- (D) ID CM assignment CM exp | EQ

**Câu 15.** [L.O.1.2] Về phải luật sinh của ui\_list mô tả một danh sách gồm ít nhất một phần tử UI được viết dưới dạng BNF là

- (A) UI ui\_list | UI
- (B) UI ui\_list |
- (C) UI ui\_list
- (D) ui\_list | UI

**Câu 16.** [L.O.2.1] Cho một danh sách bao gồm các phần tử có thể lồng nhau trong một danh sách, ví dụ:

```
nested_lst = [1, 2, [3, 4, [5, 6], 7], 8, [9]]
```

Hàm flatten có thể nhận danh sách trên và trả ra danh sách ở dạng làm phẳng như [1, 2, 3, 4, 5, 6, 7, 8, 9]. Thân của flatten là

- (A) return reduce(lambda prev, curr: prev + curr), lst, [])
- (B) return reduce(lambda prev, curr: prev + (flatten(curr) if type(curr) is list else curr), lst, [])
- (C) return reduce(lambda prev, curr: prev + [curr], lst, [])
- (D) return reduce(lambda prev, curr: prev + (flatten(curr) if type(curr) is list else [curr]), lst, [])

**Câu 17.** [L.O.2.1] Cho lớp A là lớp cha của lớp B trên một ngôn ngữ lập trình hướng đối tượng kiểm tra kiểu tĩnh như Java, Scala, với biến a và b có kiểu tương ứng là A và B. Cho hai phát biểu gán sau:

```
a = new B(); // phát biểu 1
```

```
b = new A(); // phát biểu 2
```

- (A) Cả hai phát biểu 1 và 2 đều sai
- (B) Phát biểu 1 đúng và phát biểu 2 sai
- (C) Phát biểu 1 sai và phát biểu 2 đúng
- (D) Cả hai phát biểu 1 và 2 đều đúng

**Câu 18.** [L.O.1.2] Cho một tập luật sinh như sau:

$$S \rightarrow aSb|T$$
$$T \rightarrow cTd|\epsilon$$

Một quá trình dẫn xuất trái nhất là

- (A)  $S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aaaSbbb \Rightarrow aaaaSbbbb \Rightarrow aaaaTbbbb \Rightarrow aaaabbbb$
- (B)  $S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aacTdbb \Rightarrow aaccTddb \Rightarrow aaccddbb$
- (C)  $S \Rightarrow aSb \Rightarrow aTb \Rightarrow acTdb \Rightarrow acedb$
- (D) Các lựa chọn khác đều đúng

**Câu 19.** [L.O.1.2] Cho văn phạm phi ngữ cảnh G với tập ký hiệu kết thúc là {ADD, MINUS, MUL, DIV, LB, RB}, tập ký hiệu không kết thúc là {exp, term, fact}, ký hiệu bắt đầu là exp, và tập luật sinh là:

$$\text{exp} \rightarrow \text{term MINUS exp} \mid \text{term}$$
$$\text{term} \rightarrow \text{fact DIV term} \mid \text{fact}$$
$$\text{fact} \rightarrow \text{fact ADD factor} \mid \text{fact MUL factor} \mid \text{factor}$$
$$\text{factor} \rightarrow \text{LB exp RB} \mid \text{INT}$$

Cho INT là token của các số nguyên, ADD của phép cộng, MINUS của phép trừ, MUL của phép nhân và DIV của phép chia nguyên, LB của '(' và RB của ')'. Hãy xác định độ ưu tiên và tính kết hợp của các phép toán để từ đó tính toán giá trị của chuỗi nhập sau  $100 - 4 / 4 + 2 * 3 / 9 - 10$  ?

- (A) 89
- (B) 108
- (C) 109
- (D) 90

**Câu 20.** [L.O.1.2] Cho đoạn mã sau trong ANTLR

```
decl: ID decl_tail;  
decl_tail: CM decl | CL ID CM;
```

Về phải nào sau đây phù hợp cho luật sinh `decl` để tương đương đoạn mã trên?

- (A) `ID (CM ID)* CL ID CM` (B) `(ID CM)* (CL ID CM)?`  
(C) `(ID CM)* CL ID CM` (D) `ID CM ID (CL ID)* CM`

**Câu 21.** [L.O.2.1] Câu nào sau đây miêu tả đúng về hàm bậc cao (high-order function)?

- (A) Là một hàm luôn trả ra một hàm khác  
(B) Là một hàm có thể nhận đối số là một hàm khác  
(C) Là một hàm luôn phải nhận đầy đủ đối số là một mảng và một hàm để xử lý  
(D) Là một hàm sử dụng đệ quy để lặp qua một tập giá trị

**Câu 22.** [L.O.2.1] Khi lập trình bằng Python, để đưa ký hiệu xuống hàng trực tiếp vào trong chuỗi bằng cách nhấn phím Enter thay vì dùng chuỗi escape `\n`, chuỗi phải được đặt trong:

- (A) phải dùng escape, không thể nhập trực tiếp (B) một cặp 3 dấu nháy kép (C) một cặp dấu nháy kép  
(D) một cặp dấu nháy đơn

**Câu 23.** [L.O.1.1] Cho biểu thức chính quy `a[^abc]*c` và các chuỗi nhập gồm `adc`, `abbc`, `ayyyyyyyyyyc`, `abc`, `aabc`, `axc`. Số chuỗi nhập thỏa mãn biểu thức chính quy là

- (A) 2 (B) 1 (C) 5 (D) 3

**Câu 24.** [L.O.1.2] Một danh sách `lcase` bao gồm các `case` nối tiếp nhau và có thể rỗng. Một `case` bao gồm từ khóa `CASE`, sau đó là một biểu thức `exp` và một dấu `CL` và kết thúc bởi danh sách các phát biểu `stmtlist`. Dạng EBNF của về phải luật sinh `lcase` là

- (A) `CASE exp* CL stmtlist?` (B) `CASE (exp CL stmtlist)*?`  
(C) `(CASE exp* CL stmtlist)*?` (D) `(CASE exp CL stmtlist)*`

**Câu 25.** [L.O.1.1] Trong ngôn ngữ X, lập trình viên có thể viết các hằng số nguyên không chính xác bằng cách thay thế các ký tự số chưa biết bằng dấu `#` với ký tự liền trước nó phải là ký tự số. Hằng số nguyên không chính xác phải được bắt đầu bằng ít nhất hai ký tự số. Ví dụ như `12#345`, `123#45#` là hợp lệ nhưng `123##45` là không hợp lệ.

Trong ANTLR4, cho các fragment mô tả ký tự số là `DIGIT`, dấu `#` là `SHARP`. Phần mô tả nào sau đây thể hiện cho hằng số nguyên không chính xác trong ngôn ngữ X

- (A) `DIGIT (DIGIT* SHARP)*` (B) `DIGIT (DIGIT SHARP?)+` (C) `DIGIT (DIGIT SHARP?)*` (D) `DIGIT (DIGIT SHARP)+`

**Câu 26.** [L.O.1.2] Ký hiệu `list` mô tả một danh sách (có thể rỗng) các phần tử `a` được viết cách nhau bởi một dấu `C`. Văn phạm cho `list` được viết trên ANTLR như sau:

```
list: elist | (1);  
elist: a C elist | (2);
```

(1) và (2) lần lượt là (biết rằng ký tự `_` đại diện cho rỗng)

- (A) `_` và `C` (B) `_` và `a` (C) `C a` và `_` (D) `_` và `_`

**Câu 27.** [L.O.1.1] Để biểu diễn dạng lũy thừa  $a^b$  với  $a$  và  $b$  là các số nguyên dương trong Latex, người ta viết cơ số  $a$ , tiếp theo là dấu mũ, tiếp theo là phần cơ số  $b$ . Nếu cơ số  $b$  có từ hai chữ số trở lên phải được bao trong dấu ngoặc nhọn. Nếu cơ số  $b$  chỉ có một chữ số, việc sử dụng cặp ngoặc là tùy chọn.

Trong ANTLR4, cho các fragment mô tả ký tự số là `DIGIT`, dấu mũ là `HAT`, cặp ngoặc nhọn mở và đóng lần lượt là `LP` và `RP`. Phần mô tả nào sau đây thể hiện được dạng lũy thừa nói trên

- (A) `DIGIT+ HAT (DIGIT | LP DIGIT+ RP)` (B) `DIGIT+ HAT (DIGIT | LP DIGIT DIGIT+ RP)`  
(C) `DIGIT+ HAT (LP (DIGIT | DIGIT+) RP)` (D) `DIGIT+ HAT DIGIT | LP DIGIT+ RP`

**Câu 28.** [L.O.1.1] Cho một mô tả từ vựng được định nghĩa trong ANTLR4 như sau:

```
UNIVERSE: A* S A A A A A+;  
fragment A: D | C | S;  
fragment D: [0-9];  
fragment C: [a-zA-Z];  
fragment S: [@!%*#?&];
```

Chuỗi nhập ứng với token `UNIVERSE` có tính chất nào sau đây?

- (A) Có ít nhất 6 ký tự và khi chứa ký tự chữ thường thì không chứa ký tự chữ hoa và ngược lại  
(B) Có ít nhất 4 ký tự và phải có chứa ít nhất một ký tự đặc biệt (`@!%*#?&`)  
(C) Có ít nhất 6 ký tự và phải có chứa ít nhất một ký tự đặc biệt (`@!%*#?&`)  
(D) Có nhiều nhất 8 ký tự và phải có chứa ký tự chữ thường hoặc chữ số

**Câu 29.** [L.O.1.2] Câu nào sau đây đúng về ngôn ngữ sinh ra bởi văn phạm phi ngữ cảnh sau?

$$S \rightarrow AB, A \rightarrow aA|\epsilon, B \rightarrow bB|\epsilon$$

- (A) Ngôn ngữ này gồm tất cả các chuỗi dạng  $a^n b^n$ , trong đó  $n$  là số nguyên không âm.
- (B) Ngôn ngữ này gồm tất cả các chuỗi trên ngôn ngữ với tập ký tự  $\{a, b\}$
- (C) Ngôn ngữ này gồm tất cả các chuỗi dạng  $a^m b^n$ , trong đó  $m, n$  là số nguyên không âm.
- (D) Ngôn ngữ này gồm chỉ chuỗi rỗng

**Câu 30.** [L.O.2.1] Một biến trên Python sẽ

- (A) phải khai báo trước khi gán trị
- (B) giữ con trỏ đến một đối tượng
- (C) nhận một giá trị để lưu giữ
- (D) phải cần khai báo kiểu

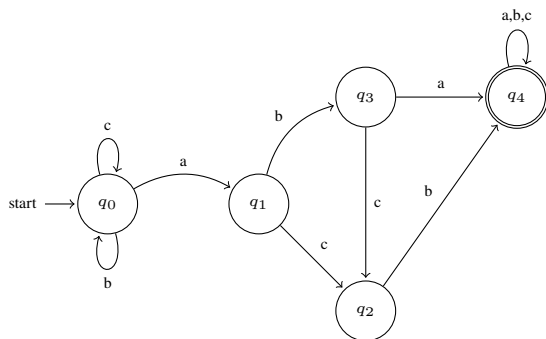
**Câu 31.** [L.O.2.1] Cho đoạn mã sau trên Python:

```
x, y = 0, 4
while x < 3:
    x += 1
else: print(y)
```

Sau khi thực thi đoạn mã trên,

- (A) Đoạn mã trên gây ra lỗi văn phạm
- (B) Giá trị 4 được in ra
- (C) Giá trị 3 được in ra
- (D) Không có giá trị nào được in ra

**Câu 32.** [L.O.3.1] Cho một automata như sau:



Chuỗi nhập nào sau đây được chấp nhận bởi automata trên:

- (A) bcabcaabcb
- (B) bccbabcbaab
- (C) abcaabcbaaa
- (D) baccbacbcca

**Câu 33.** [L.O.2.1] Một chương trình được viết trong ngôn ngữ X có nội dung như sau:

- Cho người dùng nhập giá trị vào biến **a** từ bàn phím.
- Thực hiện một vòng lặp **a** lần với thân vòng lặp có nội dung:
  - Tính toán giá trị biến **b**.
  - Nếu biến **b** lớn hơn 5 thì in ra giá trị biến **b** và thoát khỏi vòng lặp.
  - Thực hiện phát biểu in **a** nhưng có chứa lỗi ngữ pháp.

Trong quá trình thực thi, người ta thấy có lúc thì chương trình thực hiện thành công có lúc lại ném ra lỗi. Phương pháp nào đã được sử dụng để hiện thực ngôn ngữ X nói trên?

- (A) Phương pháp lai có sử dụng trình dịch tức thời (Hybrid implementation with just-in-time compiler)
- (B) Trình thông dịch (Pure interpreter)
- (C) Trình biên dịch (Compiler)
- (D) Phương pháp lai (Hybrid implementation)

**Câu 34.** [L.O.1.1] Chọn biểu thức chính qui chấp nhận ít nhất tất cả các chuỗi trong tập MATCH nhưng không chấp nhận bất kỳ chuỗi nào trong tập SKIP sau:

MATCH = {Cho, chi, Chung, Che, Chan }

SKIP = {Tro, Ching, Chu, Tre, Tran}

- (A) (Clc)h(olile)lCh(alu)n?g?
- (B) [cCT][hr][aeuio]n?g?
- (C) [cC]h[aoiue]n?g?
- (D) [Cc]h[oie]lCh[au]ng?

**Câu 35.** [L.O.1.1] Chọn biểu thức chính qui tương đương với biểu thức chính qui sau:  $(alb)^*(abblb)a$

- (A)  $[ab]^*(ab)+ba$
- (B)  $(b^*a^*)^*(ab)?ba$
- (C)  $[alb]^*[abblb]a$
- (D)  $[ab]^*[ab]?ba$

**Câu 36.** [L.O.1.2] Hãy cho biết văn phạm nào sau đây bị nhập nhằng

- (A)  $S \rightarrow \epsilon | a|b|aSa|bSb$
- (B)  $S \rightarrow \epsilon | aSbS$
- (C)  $S \rightarrow \epsilon | aSa|bSb$
- (D)  $S \rightarrow AB|BA, A \rightarrow \epsilon | aA, B \rightarrow \epsilon | bB$

**Câu 37.** [L.O.2.1] Cho các khai báo trong một ngôn ngữ lập trình hướng đối tượng với kiểm tra kiểu tĩnh:

```
class A { def foo() = print("a") }  
class B extends A { } // B is a subclass of A  
class C extends A { override def foo() = print("c") } // C is a subclass of A  
class D extends B { override def foo() = print("d") } // D is a subclass of B
```

Biết rằng, biến `b` được khai báo kiểu `B` và đang tham chiếu đến một đối tượng nào đó. Cho một số nhận định về kết quả được in ra khi gọi `b.foo()`

- (a) `c` (nếu `b` đang tham chiếu đến một đối tượng `C`)
- (b) `d` (nếu `b` đang tham chiếu đến một đối tượng `D`)
- (c) `a` (nếu `b` đang tham chiếu đến một đối tượng `A`)
- (d) `a` (nếu `b` đang tham chiếu đến một đối tượng `B`)

Số nhận định đúng là

- ☐ (A) 0                      ☐ (B) 1                      ☐ (C) 2                      ☐ (D) 3

**Câu 38.** [L.O.1.1] Một danh hiệu trong ngôn ngữ lập trình Ruby là một chuỗi các ký tự số, chữ thường và dấu gạch dưới. Nó phải được bắt đầu bằng một dấu gạch dưới hoặc một ký tự chữ thường. Chọn một biểu thức chính quy phù hợp để mô tả danh hiệu nói trên?

- ☐ (A) `[0-9_][a-z0-9_]+`                      ☐ (B) `[a-z0-9_]+`                      ☐ (C) `[a-zA-Z0-9_]+`                      ☐ (D) `[a-z_][a-z0-9_]*`

Áp dụng đoạn mã sau trong ngôn ngữ Python cho các câu 39–40:

```
result = (lst[0] * 2) + func(x, y) - (lst[-1] if lst[1] >= -1.2 else lst[2]) % 5 # calculate result
```

**Câu 39.** [L.O.2.1]

Số token được phân tích từ vệt trả về khi phân tích từ vệt cho chuỗi trên là

- ☐ (A) 45                      ☐ (B) 38                      ☐ (C) 43                      ☐ (D) 40

**Câu 40.** [L.O.2.1]

Chuỗi lexeme của token thứ 25 là:

- ☐ (A) `lst`                      ☐ (B) `-1`                      ☐ (C) `]`                      ☐ (D) `if`

## ĐÁP ÁN

Câu 1. C

Câu 2. C

Câu 3. B

Câu 4. B

Câu 5. B

Câu 6. A

Câu 7. B

Câu 8.

Câu 9. B

Câu 10. C

Câu 11. B

Câu 12. D

Câu 13. B

Câu 14. C

Câu 15. A

Câu 16. D

Câu 17. B

Câu 18. A

Câu 19. B

Câu 20. A

Câu 21. B

Câu 22. B

Câu 23. D

Câu 24. D

Câu 25. B

Câu 26. B

Câu 27. A

Câu 28. C

Câu 29. C

Câu 30. B

Câu 31. B

Câu 32. B

Câu 33. B

Câu 34. D

Câu 35. B

Câu 36. D

Câu 37. C

Câu 38. D

Câu 39. D

Câu 40. D