



KHOA <b>KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH</b>	Họ và tên:
BỘ MÔN <b>KHOA HỌC MÁY TÍNH</b>	MSSV:

Đề thi cuối kỳ

Môn thi: Nguyên Lý Ngôn Ngữ Lập Trình

Thời gian: 120 phút Ngày thi: 23-12-2015

□ Sinh viên được phép sử dụng tài liệu □ Sinh viên không được sử dụng tài liệu

Mã đề: 1151

- Sinh viên phải ghi tên và mã số sinh viên trên đề thi (**trang đầu và trang cuối**), giấy làm bài trắc nghiệm (giấy đỏ) và giấy làm bài. Sinh viên phải tô phần mã đề thi và mã số sinh viên trên giấy làm bài trắc nghiệm. Khi nộp bài, sinh viên phải nộp cả **đề thi, giấy làm bài trắc nghiệm và giấy làm bài**.
- Phần trắc nghiệm sẽ được chấm TỰ ĐỘNG trên giấy làm bài trắc nghiệm. Do đó, phần trắc nghiệm nếu làm trên đề thi sẽ KHÔNG được chấm.
- Đối với các câu hỏi phần trắc nghiêm, sinh viên chỉ chon MÔT phương án đúng nhất.
- Đối với phần câu hỏi tự luận (phần II), sinh viên làm ngay trên đề thi, ở phần dành riêng ngay dưới mỗi câu hỏi.
- Đối với câu hỏi bài tập lớn (phần III và IV), sinh viên làm trên giấy làm bài.
- Sinh viên lớp đại trà làm 3 phần (I, II và III). Sinh viên lớp tài năng làm cả 4 phần (I, II, III và IV). Chỉ có phần I và II được dùng để tính điểm cuối kỳ, các phần III và IV để tính điểm bài tập lớn 2 và 3.

#### I. Phần câu hỏi trắc nghiêm:(8 điểm))

- 1. Giả sử chương trình có một lỗi văn phạm nằm sau lệnh **print**. Khi em nhấn nút Run (để dịch và chạy chương trình) trên trình soạn thảo, kết quả được in ra bởi lệnh **print** trên trước khi lỗi văn phạm được báo. Hỏi chế độ dịch của trình soạn thảo là gì?
- (A) trình liên kết (link editor)

(B) trình biên dịch (compiler)

(C) trình thông dịch(interpreter)

(D) trình hợp ngữ (assembler)

- (E) trình biên dịch động(just-in-time compiler)
  - 2. Cho biết mã Jasmin của biểu thức viết bằng ngôn ngữ C như sau: a \* b 2

a b - 2

với a và b là 2 biến nguyên có chỉ số lần lượt là 0 và 1.

iload\_0 iload\_1 iload\_0 iload\_1 iload\_0 imul iload 1 iload\_0 iload\_1 iconst 2 (A) imul iconst\_2 isub ...

 $\begin{array}{c} \text{(B)} & \underset{\text{isub}}{\text{iconst}} -2 \\ & \underset{\text{isub}}{\text{imul}} \end{array}$ 

isub iconst\_2

isub imul

(E) isub iconst\_2 ...

... ... ...



Đoạn code sau, được viết trên ngôn ngữ dùng **qui tắc tầm vực tĩnh (static-scope rule)**, áp dụng cho các câu 3–6

- 3. Môi trường tham khảo tĩnh (static referencing environment) của thủ tục **sub3** KHÔNG chứa
- (A) a ở //1
- (B) a  $\dot{\sigma}$  //2
- (C) sub1
- (D) sub2
- (E) sub3
- 4. Giả sử chuỗi gọi là main  $\to$  sub<br/>1  $\to$  sub2  $\to$  sub3, tham khảo đến **a** trong **sub3** ứng với khai báo
- (A) Báo lỗi a chưa khai báo
- (B) a trong sub1 (//2)
- (C) a trong sub2 (//3)

- (D) a trong sub3
- (E) a trong main (//1)
- 5. Giả sử chuỗi gọi là main  $\to$  sub<br/>1  $\to$  sub2  $\to$  sub3 , tham khảo đến **a** trong sub3 **khi thực thi** ứng với kết hợp của **a** trong bản hoạt động của
- (A) main
- (B) sub1
- (C) sub2
- (D) sub3

- (E) Báo lỗi không tìm thấy a
  - 6. Nếu đoạn code trên được viết trên ngôn ngữ dùng qui tắc tầm vực động (dynamic-scope rule), và giả sử chuỗi gọi là main  $\rightarrow$  sub1  $\rightarrow$  sub2  $\rightarrow$  sub3, tham khảo đến a trong sub3 khi thực thi ứng với kết hợp của a trong bản hoạt động của
- (A) Báo lỗi không tìm thấy a
- (B) main
- (C) sub1
- (D) sub2

(E) sub3

Cho văn phạm của một biểu thức sau dùng cho các câu 7–9:

7. Chọn biểu thức được viết đúng văn phạm, có số lượng dấu '(' và ')' ít nhất mà tương đương với biểu thức sau:  $(\mathbf{a} + (\mathbf{b} = ((\mathbf{c} * \mathbf{d}) * \mathbf{e}))) > \mathbf{f}$ 

```
(A) (a + (b = (c * d) * e)) > f
```

(B) 
$$(a + (b = ((c * d) * e))) > f$$

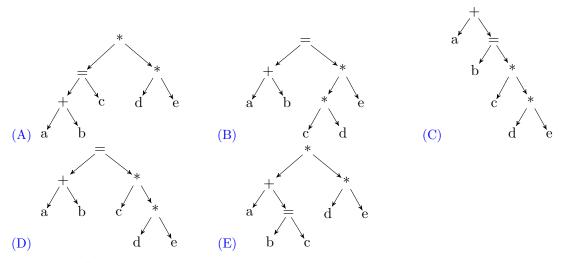
(C) a + b = c \* d \* e > f

(D) 
$$a + (b = (c * d) * e) > f$$

(E) a + (b = c \* d \* e) > f



8. Vẽ AST cho biểu thức sau được viết dựa vào văn phạm trên:  $\mathbf{a} + \mathbf{b} = \mathbf{c} * (\mathbf{d} * \mathbf{e})$ 



9. Cho cấu trúc AST được khai báo trên Scala như sau: trait Exp case class Bin(op:String,e1:Exp,e2:Exp) extends Exp

Chọn AST cho biểu thức sau:  $\mathbf{a} + \mathbf{b} > \mathbf{c} = \mathbf{d} * \mathbf{e} = \mathbf{f}$ 

- $(A) \ \operatorname{Bin}("=",\operatorname{Bin}(">",\operatorname{Bin}("+",\operatorname{Id}("a"),\operatorname{Id}("b")),\operatorname{Id}("c")),\operatorname{Bin}("*",\operatorname{Id}("d"),\operatorname{Bin}("=",\operatorname{Id}("e"),\operatorname{Id}("f"))))$
- $\begin{array}{l} \text{Bin}("=", \text{Bin}("=", \text{Bin}(">", \text{Bin}("+", \text{Id}("a"), \text{Id}("b")), \text{Id}("c")), \text{Bin}("*", \text{Id}("d"), \text{Id}("e"))), \text{Id}("f")) \end{array}$
- (C) Bin("=",Bin(">",Bin("+",Id("a"),Id("b")),Id("c")),Bin("=",Bin("\*",Id("d"),Id("e"))),Id("f"))
- (D) Bin("=",Bin("+",Id("a"),Bin(">",Id("b"),Id("c"))),Bin("=",Bin("\*",Id("d"),Id("e"))),Id("f"))
- (E) Bin("=",Bin("=",Bin("+",Id("a"),Bin(">",Id("b"),Id("c"))),Bin("\*",Id("d"),Id("e"))),Id("f"))

Phần trình bày sau dùng trong các câu hỏi 10–11:

case class Id(i:String) extends Exp

Cho mã giả của phát biểu for var = expr1 to expr2 do body như sau:

start: var = expr1

etemp = expr2

loop: if (var >= etemp) goto out

body

label1: var++

label2: goto loop

out:

- 10. Nếu trong thân *body* có thực thi lệnh *continue* thì điều khiển sẽ chuyển đến thực thi phát biểu có nhãn là:
- (A) out
- (B) start
- (C) loop
- (D) label1
- (E) label2
- 11. Với hiện thực của phát biểu *for* như trên, phát biểu nào đặt trong thân phát biểu **for** dưới đây sẽ làm cho phát biểu này bị lặp mãi mãi?

(A) i = i - 1;

(B) s = s - 1;

(C) n = n + 1;

- (D) câu A và C đúng
- (E) Không phát biểu nào trong các câu A, B và C làm phát biểu for lặp mãi mãi



(A)

12. Cho biết mã Jasmin của phát biểu gán sau:

$$a[i] = a[i] + 2$$

với i kiểu nguyên có chỉ số là 3 và a là kiểu dãy nguyên có chỉ số là 1.

```
aload 1
                                                        aload 1
iastore
                                                                           aload 1
                                     iload 3
                   aload 1
                                                        dup
iadd
                                                                           iload 3
                   iload 3
                                     aload 1
                                                        iload 3
iaload
                                                                           iastore
                   iaload
                                     iload 3
                                                        dup
aload 1
                                                                           aload 1
             (B) iconst 2
                                     iaload
                                (C)
                                                   (D)
                                                        iaload
                                                                      (E)
iload 3
                                                                           iload 3
                   iadd
                                     iconst 2
                                                        iconst 2
                                                                           iconst 2
dup
                   iastore
                                     iadd
                                                        iadd
iconst\_2
                                                                           iadd
                                     iastore
                                                        iastore
                                                                            ...
```

Đoạn code sau dùng cho các câu 13-15

```
int p;
int* foo(int x) {
    static int q;
    int *s = new int;
    switch (x) {
        case 1: return &p;
        case 2: return &q;
        case 3: return &x;
        case 4: return s;
        default: return foo(x-1);
    }
}
```

- 13. Phát biểu nào sẽ gây ra lỗi tham chiếu treo (dangling reference) khi thực thi
- (A) return &p
- (B) return &q
- (C) return &x
- (D) return s

- (E) Không phát biểu nào gây ra lỗi trên khi thực thi
- 14. Trong đoạn mã trên, đối tượng dữ liệu nào có thể trở thành rác (garbage)
- (A) p

(B) q

(C) x

(D) trỏ đến bởi s

- (E) Không có đối tượng nào có thể trở thành rác
- 15. Khi hàm **foo** được gọi đệ quy (recursive) thì các bản hoạt động của **foo** dùng chung những đối tượng dữ liệu nào?
- (A) q và s
- (B) p, q, x và s
- (C) p
- (D) p và q
- (E) x, q và s
- 16. Viết lại biểu thức trung tố (infix) sau sang dạng hậu tố (postfix) Polish? Giả sử là tất cả các phép toán đều có 2 toán hạng và có độ ưu tiên và tính kết hợp như trong ngôn ngữ lập trình C.

$$a + b * c - d * e + f$$

(A) 
$$a + b c * d e * - f +$$

(B) 
$$a b + c * d - e * f +$$

(C) a b c 
$$*$$
 + d - e f  $*$  +

(D) 
$$a b c * + d e * + f -$$

(E) 
$$a b c * + d e * - f +$$



17. Cho khai báo sau được viết trên ngôn ngữ lập trình Ada:

Cho kích thước của các kiểu *Integer*, *Float*, *Boolean và Enumeration* lần lượt là 2, 4, 1, và 2, và kích thước cho phần mô tả kiểu (type description) là 0. Cho biết kích thước của môt đối tương kiểu **Figure**?

- (A) 21
- (B) 13
- (C) 23
- (D) 15
- (E) Khác
- 18. Cơ chế gọi chương trình nào là cơ chế gọi chương trình con cơ bản của Lập trình hướng sự kiện (Event-driven Programming)
- (A) Biến cố Xử lý biến cố (Exception)
- (B) Gọi trở về đơn giản (Simple Call Return)
- (C) Gọi đệ qui (Recursive call)
- (D) Trình công hành (Coroutine)
- (E) Trình định thời (Scheduled Subprogram)
  - 19. Đoạn code viết bằng Scala nào dưới đây có thể dùng như một ví dụ minh họa cho khái niệm Hàm Currying
- (A)  $\operatorname{def} \operatorname{add1}(n:\operatorname{Int}) = n + 1$
- (B)  $List(1,2,3).foldLeft(0)(_ + _ )$
- (C) def add(n:Int) = (x:Int) => n + x; val add2 = add(2)
- (D)  $\operatorname{def} \operatorname{add}(n:\operatorname{Int})(x:\operatorname{Int}) = n + x$ ;  $\operatorname{val} \operatorname{add}2 = \operatorname{add}(2)$
- (E) List(1,2,3).filter( > 1)
- 20. Liệt kê theo thứ tự chiều dài tăng dần ít nhất 5 chuỗi ngắn nhất (nếu có nhiều hơn 5 chuỗi) của ngôn ngữ được mô tả bởi biểu thức chính quy (regular expression) sau: ab\*(a|b)? a
- (A) aa, aba, aaa, abaa, aaba
- (B)  $\epsilon$ , a, aa, aba, abaa
- (C) aa, aba, aaa, abaa, abba

- (D) a, aa, aba, aaa, aaba
- (E) aa, aba, aaa, abaa, aaaa
- 21. Cho giá trị ban đầu của biến **c** là 3, cho biết những giá trị **có thể có** của biến **a** sau khi thực hiện phép gán sau trên ngôn ngữ lập trình C? a = c \* (c = 5);
- (A) 15, 25, 9
- (B) 15
- (C) 25
- (D) 9
- (E) 15, 25
- 22. Cho biết kích thước của đối tượng dữ liệu  ${\bf x}$  được khai báo như sau: x: set of 1..16
- (A) 4 bytes
- (B) 2 bytes
- (C) 4 bits
- (D) 1 byte
- (E) 16 bytes



23. Cho lớp(class) A là lớp cha (superclass) của lớp B và B là lớp cha của hai lớp C và D. Trong lớp A, có khai báo phương thức thực thể (instance method) **foo** và phương thức này bị ghi đè(override) trên tất cả các lớp B, C và D. Cho biến x được khai báo như sau: **B** x;

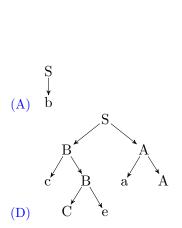
Qui ước viết  $func_T$  là phương thức func được khai báo trong lớp T, cho biết tập của phương thức đích của cuộc gọi sau:  $\mathbf{x.foo}()$ ;

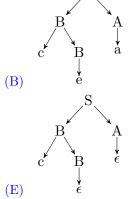
- (A)  $\{foo_C, foo_D\}$
- (B)  $\{foo_A, foo_B, foo_C, foo_D\}$
- (C)  $\{foo_B\}$

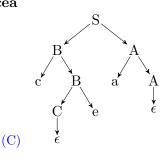
- (D)  $\{foo_A, foo_B\}$
- (E)  $\{foo_B, foo_C, foo_D\}$
- 24. Chọn phát biểu ĐÚNG trong các phát biểu về dãy (array) sau:
- (A) Trên C, kích thước của dãy có thể tăng thêm sau khi dãy đã được tao ra
- (B) Truy xuất một phần tử bên ngoài một dãy C không được phép, sẽ lập tức gây ra lỗi
- (C) Trên C và C++, kích thước của một dãy được lưu trữ để kiểm tra động
- (D) Trên Java, kích thước của một dãy KHÔNG được lưu trữ để kiểm tra đông
- (E) Truy xuất một phần tử bên ngoài một dãy Java không được phép, sẽ lập tức gây ra lỗi

Cho văn phạm sau dùng cho các câu 25–26:

- $S \rightarrow b \mid BA$
- $A \rightarrow a A \mid \epsilon$
- $B \rightarrow Ce \mid cB$
- $C \rightarrow dC \mid \epsilon$
- 25. Chuỗi nào KHÔNG thuộc ngôn ngữ được mô tả bởi văn pham trên?
- (A) e
- (B) b
- (C) eaaa
- (D) cccddddeaa
- (E) cccaaa
- 26. Chọn cây phân tích cú pháp (parse tree) cho chuỗi nhập: cea







Phần hướng dẫn này áp dụng cho các câu 27–31

Hãy viết thêm các đoạn mã cần thiết vào phương thức dưới đây để thực hiện sinh mã cho phát biểu *repeat*. Nhắc lại phát biểu *repeat* kết thúc quá trình lặp khi điều kiện của phát biểu trở thành **true**. AST của phát biểu **repeat** được định nghĩa như sau: case class Repeat(val stmts:List[Stmt],val exp:Expr) extends Stmt

```
override def visitRepeat(ast: Repeat, o:Context)= {
    val ctxt = o.asInstanceOf[SubBody]
    ctxt.frame.enterLoop();
```



```
val labelStart = ctxt.frame.getNewLabel()
val labelBreak = ctxt.frame.getBreakLabel()
val labelCont = ctxt.frame.getContinueLabel()
val str1 = //1
// sinh mã cho từng phát biểu trong thân của repeat
val str2 = //2
val str3 = //3
// sinh mã cho biểu thức điều kiện
val str4 = visit(ast.exp,...)
val str5 = //4
val str6 = //5
frame().exitLoop();
str1 + str2 + str3 + str4 + str5 + str6
}
```

- 27. Đoạn mã nào trong số các đoạn mã dưới đây cần phải xuất hiện ở vị trí //1 trong đoạn mã trên
- (A) ""

- (B) ctxt.emit.emitLABEL(labelStart)
- (C) ctxt.emit.emitLABEL(labelBreak)
- (D) ctxt.emit.emitLABEL(labelCont)
- (E) ctxt.emit.emitGOTO(labelStart)
- 28. Đoạn mã nào trong số các đoạn mã dưới đây cần phải xuất hiện ở vị trí //2 trong đoạn mã trên
- (A) visit(ast.stmts,o)
- (B) ast.stmts.map(x = > visit(x, o))
- (C) ast.stmts.filter(x = > visit(x, o))
- (D) ast.stmts.foldLeft("")((x,y)=>x + visit(y,o))
- (E) ast.stmts.foldLeft("")((x,y)=>y + visit(x,o))
- 29. Đoạn mã nào trong số các đoạn mã dưới đây cần phải xuất hiện ở vị trí //3 trong đoạn mã trên
- (A) ""

- (B) ctxt.emit.emitIFFALSE(labelStart)
- (C) ctxt.emit.emitLABEL(labelBreak)
- (D) ctxt.emit.emitLABEL(labelCont)
- (E) ctxt.emit.emitGOTO(labelStart)
- 30. Đoạn mã nào trong số các đoạn mã dưới đây cần phải xuất hiện ở vị trí //4 trong đoạn mã trên
- (A) ctxt.emit.emitGOTO(labelBreak)
- (B) ctxt.emit.emitIFFALSE(labelStart)
- (C) ctxt.emit.emitLABEL(labelBreak)
- (D) ctxt.emit.emitLABEL(labelCont)
- (E) ctxt.emit.emitGOTO(labelStart)
- 31. Đoạn mã nào trong số các đoạn mã dưới đây cần phải xuất hiện ở vị trí //5 trong đoạn mã trên
- (A) ctxt.emit.emitGOTO(labelBreak))
- (B) ctxt.emit.emitIFFALSE(labelStart))
- (C) ctxt.emit.emitLABEL(labelBreak))
- (D) ctxt.emit.emitLABEL(labelCont))
- (E) ctxt.emit.emitGOTO(labelStart))



- 32. Trình định thời (scheduled subprograms) thường được dùng trong
- (A) Lập trình song song (Parallel Programming)
- (B) Lập trình hướng đối tượng (Object-Oriented Programming)
- (C) Lập trình hàm (Functional Programming)
- (D) Lập trình hướng sự kiện (Event-driven Programming)
- (E) Lập trình thời gian thực (Real-time Programming)
- 33. Gọi append là hàm dùng để nối 2 danh sách thành 1 danh sách. Ví dụ append([1,2,3],[4,5,6]) sẽ có kết quả là [1,2,3,4,5,6]. Hãy hiện thực hàm append(a:List[Int],b:List[Int]) dùng hàm bậc cao (high-order function)?

```
(A) a.map(b)((x,y)=>x::y) (B) a.foldLeft(b)((x,y)=>x::y) (C) a.foldRight(b)((x,y)=>x::y) (D) b.foldLeft(a)((x,y)=>x::y) (E) b.foldRight(a)((x,y)=>x::y)
```

Đoạn mã sau được dùng trong các câu 34–35:

```
int *p = new int;
void foo(int * r) {
    delete r;
}
foo(p); //1
*p = 2; //2
```

- 34. Hiện tượng gì xảy ra khi p được truyền cho r ở phát biểu //1 trong đoạn mã trên:
- (A) Con trỏ chưa khai báo (undeclared pointer) (B) Bí danh (alias)
- (C) Khai báo trùng tên (redeclared) (D) Tham chiếu treo (dangling reference)
- (E) Rác (garbage)
- 35. Hiện tượng gì xảy ra khi thực thi phát biểu //2 trong đoạn mã trên:
- (A) Tham chiếu treo
- (B) Bí danh

(C) Da hình (polymorphism)

(D) Rác

(E) Con trỏ chưa khai báo

Đoạn mã sau được dùng cho các câu 36–41.

Cho một đoan chương trình được viết trên một ngôn ngữ tưa C như sau:

```
int A[5] = {1,3,5,7,9};// index of A starts from 0
int j = 0;
int n = 5;
int sumAndIncrease(int a, int i) {
    int s = 0;
    for ( ; i < n; i = i + 1) {
        s = s + a;
        A[j] = A[j] + 1;
    }
    return s;
}

void main(){
    int s = sumAndIncrease(A[j],j);
    printf("a = %i\n",s); //1
    printf("%i %i %i %i %i\n",A[0],A[1],A[2],A[3],A[4]);//2
}</pre>
```



	và $i$ được truyền bằng ${f t}$ ı ra ở phát biểu $//1$ là b	- \-	d by value-result)	, giá trị của biến $s$
A) 5	(B) 10	(C) 15	(D) 25	(E) Khác
được in	và $i$ được truyền bằng $\mathbf{t}_1$ 1 ra ở phát biểu $//2$ là b	ao nhiêu?	,	
A) 1 3 5 7 9	(B) 2 4 6 8 10	(C) 6 3 5 7 9	(D) 5 3 5 7 9	(E) Khác
	và $i$ được truyền bằng ${f th}$ phát biểu $//1$ là bao nh	\ <del>-</del>	by reference), giá	trị của biến $s$ được
A) 5	(B) 10	(C) 15	(D) 25	(E) Khác
được in	và $i$ được truyền bằng tra ở phát biểu $//2$ là ba	ao nhiêu?		
A) 13579			(D) 5 3 5 7 9	(E) Khác
biểu //	và $i$ được truyền bằng ${f t\hat e}$ 1 là bao nhiêu?		, · -	_
A) 5	(B) 15	(C) 20	(D) 25	(E) Khác
	và $i$ được truyền bằng $\mathfrak{d}$ ểu $//2$ là bao nhiêu?	·	•	iến A được in ra ở
A) 1 3 5 7 9	(B) 2 4 6 8 10	(C) 6 3 5 7 9	(D) 5 3 5 7 9	(E) Khác
Vi du: $var x =$ $x = 1;$	ến trên ngôn ngữ Javasc = "def"; vuộc giữa một biến và kiể			
C) Lập trìn	nc (implementation) h (programming) ng (loading)	· · · · · · · · · · · · · · · · · · ·	hay (running) ich (compiling)	
	$= \{a,b\}$ . Chọn biểu thức ỗi nào được tạo bởi các l p.	•	_ '	
A) a (abb*)	* (B) (b*ab*)*	(C) $b^*(abb^*)^*$	(D) b*a?(bb*a	)*b* (E) b*ab*ab*
	nion của ngôn ngữ C++ hác như trong ví dụ sau:		thành phần này như	rng đọc lên ở thành
ch } x; x.song	{     t songuyen;     ar kytu[2];      cuyen = 12;     < x.kytu[0] << " " <	< x.kytu[1] << er	ndl;	
Đặc tín	nh này của ngôn ngữ C+	+ gây ảnh hưởng X	ÁU NHẤT đối với	
A) tính trực	c giao (orthonogality) viết (writability)	(B) tín	nh đơn giản (simplicanh tin cậy (reliabilit	- /



- 45. Cơ chế gọi chương trình con nào mà điều khiển có thể chuyển vào một vị trí ở giữa chương trình con được gọi thay vì chuyển vào đầu chương trình con được gọi?
- (A) Trình định thời (Scheduled Subprogram)
- (B) Goi trở về đơn giản (Simple Call Return)
- (C) Goi đê qui (Recursive call)
- (D) Biến cố Xử lý biến cố (Exception)
- (E) Trình cộng hành (Coroutine)
- 46. Để thực hiện tìm kiếm một chuỗi (String) trên nhiều loại danh sách khác nhau, hãy chọn giải pháp thích hợp nhất để điền vào chỗ trống trong định nghĩa hàm sau:

```
def lookup[T](x:String,lst:List[T],f:T=>String):Option[T] =
   lst match {
     case List() => None
     case h::t =>______
}
```

- (A) if (h == x) Some(h) else lookup(x,t,f)
- (B) if (f(h) == x) Some(h) else lookup(x,t,f)
- (C) if (h == f(x) Some(h) else lookup(x,t,f)
- (D) if (f(h) == x) Some(f(h)) else lookup(x,t,f)
- (E) if (h == f(x)) Some(f(h)) else lookup(x,t,f)
  - 47. Khi cần khai thác đặc tính của kiểu union(các thành phần loại trừ lẫn nhau), làm thế nào để hiện thực trên những ngôn ngữ lập trình hướng đối tượng như Java(không có kiểu union)? Cụ thể làm thế nào để hiện thực biến z kiểu union sau trên những ngôn ngữ này?

```
union {
    string loikhen;
    string loiche;
} z;
```

- (A)  $\stackrel{\text{Dịnh nghĩa lớp cha A và hai lớp con B (có thuộc tính loikhen) và C (có thuộc tính loiche), và z có kiểu A$
- (B) Đinh nghĩa một lớp A có 2 thuộc tính loikhen và loiche; và z có kiểu A
- (C) Định nghĩa lớp A có thuộc tính lo<br/>ikhen và lớp B(con của A) có thuộc tính loiche, và z có kiểu A
- (D) Định nghĩa lớp A có thuộc tính lo<br/>ikhen và lớp B(con của A) có thuộc tính loiche, và z có kiểu B
- (E) Định nghĩa lớp trừu tượng (abstract class) A và 2 lớp con B (có thuộc tính loikhen) và C (có thuộc tính loiche), và z có kiểu A

## II. Phần câu hỏi tự luận: (dành cho tất cả sinh viên) (2 điểm)

48. Trên những ngôn ngữ lập trình cho phép rút ngắn tính toán (short-circuit evaluation) biểu thức luận lý, biểu thức sau sẽ không gây ra lỗi khi thực thi.

```
((a!=0) \&\& ((b/a) > 5))
```

Hãy dùng **if then else** để mô phỏng quá trình tính toán của biểu thức luận lý trên?



49. Hãy nêu những giải pháp đã được sử dụng để tránh tham chiếu treo (dangling reference) trong trường hợp trả về hàm như sau:

```
void->int F() {
    int x = 1;
    int g() {
        return x + 1;
    }
    return g;
}
void->int gg = F();
int z = gg();
```

50. Hãy viết lại biểu thức ở dạng trung tố (infix) sau sang dạng biểu thức tiền tố (prefix) Cambridge Polish:

```
a + b * c * d - e - f
```

Biết rằng độ ưu tiên và tính kết hợp của các phép toán trong biểu thức như thông thường (đều kết hợp trái và \* có ưu tiên cao hơn +, -). Trong biểu thức tiền tố nhiều toán hạng, thứ tự tính toán cũng từ trái sang phải.

Yêu cầu: Biểu thức dạng tiền tố Cambridge Polish phải thỏa các yêu cầu sau:

- Thứ tự xuất hiện các toán hạng trong biểu thức dạng tiềm tố phải có cùng thứ tự xuất hiện như trong biểu thức trung tố
- Số ( và ) là ít nhất
- Có cùng thứ tự tính toán các phép toán với thứ tự đó trong biểu thức dạng trung tố



#### III. Phần bài tập lớn: (dành cho tất cả sinh viên)

- 51. Viết hàm visitBlock(ast:Block,o:Context) để thực hiện kiểm tra kiểu. Hàm visitBlock sẽ thực hiện chức năng xây dựng bảng danh hiệu (symbol table) và truyền bảng này khi visit các phát biểu trong khối. Yêu cầu giải thích rõ về Context cụ thể được sử dụng trong hàm và ý tưởng mã của hàm. Nhắc lại AST của Block được khai báo như sau: case class Block(val decl:List[Decl],val stmts:List[Stmt]) extends Stmt
- 52. Viết hàm visitIf(ast:If,o:Context) để thực hiện sinh mã cho một phát biểu If. Nhắc lại AST của If được khai báo như sau: case class If(val expr:Expr, val thenStmt:Stmt, val elseStmt:Option[Stmt]) extends Stmt Yêu cầu giải thích rõ về Context cụ thể được sử dụng trong hàm và ý tưởng mã của hàm.

#### IV. Phần KSTN: (dành cho sinh viên KSTN)

- 53. Viết hàm visitIf(ast:If,o:Context) để thực hiện kiểm tra lỗi FunctionNotReTurn? Yêu cầu giải thích ý tưởng thực hiện thể hiện qua mã của hàm?





Chủ nhiệm bộ môn	Giảng viên ra đề
Chữ kí:	Chữ kí:
Họ tên:	Họ tên:



#### KHOA **KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH** BỘ MÔN **KHOA HOC MÁY TÍNH**

# ĐÁP ÁN cho Đề thi cuối kỳ

Môn thi: Nguyên Lý Ngôn Ngữ Lập Trình

Thời gian: 120 phút

□ Sinh viên được phép sử dụng tài liệu

⊠ Sinh viên không được sử dụng tài liệu

Mã đề: 1151

## I. Phần câu hỏi trắc nghiệm:(8 điểm))

- 1. Giả sử chương trình có một lỗi văn phạm nằm sau lệnh **print**. Khi em nhấn nút Run (để dịch và chạy chương trình) trên trình soạn thảo, kết quả được in ra bởi lệnh **print** trên trước khi lỗi văn phạm được báo. Hỏi chế độ dịch của trình soạn thảo là gì?
- (A) trình liên kết (link editor)

iload 0

(B) trình biên dịch (compiler)

c trình thông dịch(interpreter)

(D) trình hợp ngữ (assembler)

iload 0

imul

- (E) trình biên dịch động(just-in-time compiler)
  - 2. Cho biết mã Jasmin của biểu thức viết bằng ngôn ngữ C như sau: a \* b 2

với a và b là 2 biến nguyên có chỉ số lần lượt là 0 và 1.

iload 0

```
iload 1
                        iload 1
                                                              iload 1
                                           imul
                                                                                 load 0
                                                                                 iload 1
    imul
                        iconst 2
                                           iload 1
                                                              iconst 2
                                                        (D)
A
                                                                           (E)
    iconst 2
                        imul
                                           isub
                                                              isub
                                                                                 isub
     isub
                        isub
                                           iconst\_2
                                                              imul
                                                                                 iconst 2
```

iload 0

Đoạn code sau, được viết trên ngôn ngữ dùng **qui tắc tầm vực tĩnh (static-scope rule)**, áp dụng cho các câu 3–6



- 3. Môi trường tham khảo tĩnh (static referencing environment) của thủ tục sub3 KHÔNG chứa
- A a ở //1
- (B) a ở //2
- (C) sub1
- (D) sub2
- (E) sub3
- 4. Giả sử chuỗi gọi là main  $\rightarrow$  sub1  $\rightarrow$  sub2  $\rightarrow$  sub3, tham khảo đến  $\bf a$  trong sub3 ứng với
- (A) Báo lỗi a chưa khai báo
- $|\mathbf{B}|$  a trong sub1 (//2)
- (C) a trong sub2 (//3)

- (D) a trong sub3
- (E) a trong main (//1)
- 5. Giả sử chuỗi gọi là main  $\rightarrow$  sub1  $\rightarrow$  sub2  $\rightarrow$  sub3 , tham khảo đến  ${\bf a}$  trong sub3  ${\bf khi}$  thực thi ứng với kết hợp của a trong bản hoạt động của
- (A) main
- B sub1
- (C) sub2
- (D) sub3

- (E) Báo lỗi không tìm thấy a
  - 6. Nếu đoạn code trên được viết trên ngôn ngữ dùng qui tắc tầm vực động (dynamic-scope rule), và giả sử chuỗi gọi là main  $\rightarrow$  sub1  $\rightarrow$  sub2  $\rightarrow$  sub3, tham khảo đến a trong sub3 khi thực thi ứng với kết hợp của a trong bản hoạt đông của
- (A) Báo lỗi không tìm thấy a
- (B) main
- (C) sub1
- D sub2

(E) sub3

Cho văn pham của một biểu thức sau dùng cho các câu 7-9:

$$\exp \rightarrow \text{term '='} \exp \mid \text{term}$$

term 
$$\rightarrow$$
 term '+' fact | term '>' fact | fact

 $\rightarrow$  ope '\*' ope | ope

ope 
$$\rightarrow$$
 '(' exp ')' | ID

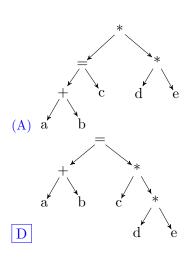
với ID là một danh hiệu.

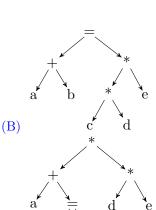
- 7. Chọn biểu thức được viết đúng văn phạm, có số lượng dấu '(' và ')' ít nhất mà tương đương với biểu thức sau:  $(\mathbf{a} + (\mathbf{b} = ((\mathbf{c} * \mathbf{d}) * \mathbf{e}))) > \mathbf{f}$
- (A) (a + (b = (c \* d) \* e)) > f

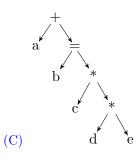
(C) a + b = c \* d \* e > f

(B) (a + (b = ((c \* d) \* e))) > fD) a + (b = (c \* d) \* e) > f

- (E) a + (b = c \* d \* e) > f
  - 8. Vẽ AST cho biểu thức sau được viết dựa vào văn phạm trên:  $\mathbf{a} + \mathbf{b} = \mathbf{c} * (\mathbf{d} * \mathbf{e})$







b

(E)



9. Cho cấu trúc AST được khai báo trên Scala như sau: trait Exp case class Bin(op:String,e1:Exp,e2:Exp) extends Exp case class Id(i:String) extends Exp

Chọn AST cho biểu thức sau:  $\mathbf{a} + \mathbf{b} > \mathbf{c} = \mathbf{d} * \mathbf{e} = \mathbf{f}$ 

- (A) Bin("=",Bin(">",Bin("+",Id("a"),Id("b")),Id("c")),Bin("\*",Id("d"),Bin("=",Id("e"),Id("f"))))
- (B) Bin("=",Bin("=",Bin(">",Bin("+",Id("a"),Id("b")),Id("c")),Bin("\*",Id("d"),Id("e"))),Id("f"))
- $\begin{array}{c} \textbf{(D)} \ \, \mathrm{Bin}("=",\mathrm{Bin}("+",\mathrm{Id}("a"),\mathrm{Bin}(">",\mathrm{Id}("b"),\mathrm{Id}("c"))),\mathrm{Bin}("=",\mathrm{Bin}("*",\mathrm{Id}("d"),\mathrm{Id}("e"))),\mathrm{Id}("f")) \\ \end{array}$
- (E) Bin("=",Bin("=",Bin("+",Id("a"),Bin(">",Id("b"),Id("c"))),Bin("\*",Id("d"),Id("e"))),Id("f"))

Phần trình bày sau dùng trong các câu hỏi 10–11:

Cho mã giả của phát biểu  $for \ var = expr1 \ to \ expr2 \ do \ body$  như sau:

start: var = expr1

etemp = expr2

loop: if (var >= etemp) goto out

body

label1: var++

label2: goto loop

out:

- 10. Nếu trong thân body có thực thi lệnh continue thì điều khiển sẽ chuyển đến thực thi phát biểu có nhãn là:
- (A) out
- (B) start
- (C) loop
- D label1
- (E) label2
- 11. Với hiện thực của phát biểu *for* như trên, phát biểu nào đặt trong thân phát biểu **for** dưới đây sẽ làm cho phát biểu này bị lặp mãi mãi?

|A| i = i - 1;

(B) s = s - 1;

(C) n = n + 1;

- (D) câu A và C đúng
- (E) Không phát biểu nào trong các câu A, B và C làm phát biểu for lặp mãi mãi
  - 12. Cho biết mã Jasmin của phát biểu gán sau:

$$a[i] = a[i] + 2$$

với i kiểu nguyên có chỉ số là 3 và a là kiểu dãy nguyên có chỉ số là 1.

(A)	iastore iadd iaload aload_1 iload_3 dup iconst_2	aload_1 iload_3 iaload 3) iconst_2 iadd iastore	C	aload_1 iload_3 aload_1 iload_3 iaload iconst_2 iadd iastore	(D)	aload_1 dup iload_3 dup iaload iconst_2 iadd iastore	(E)	aload_1 iload_3 iastore aload_1 iload_3 iconst_2 iadd
-----	--	---	---	--	-----	--	-----	---



Đoạn code sau dùng cho các câu 13-15

```
int p;
int* foo(int x) {
    static int q;
    int *s = new int;
    switch (x) {
        case 1: return &p;
        case 2: return &q;
        case 3: return &x;
        case 4: return s;
        default: return foo(x-1);
    }
}
```

- 13. Phát biểu nào sẽ gây ra lỗi tham chiếu treo (dangling reference) khi thực thi
- (A) return &p
- (B) return &q
- C return &x
- (D) return s

- (E) Không phát biểu nào gây ra lỗi trên khi thực thi
  - 14. Trong đoạn mã trên, đối tượng dữ liệu nào có thể trở thành rác (garbage)
- (A) p

(B) q

(C) x

- D trỏ đến bởi s
- (E) Không có đối tượng nào có thể trở thành rác
- 15. Khi hàm **foo** được gọi đệ quy (recursive) thì các bản hoạt động của **foo** dùng chung những đối tượng dữ liệu nào?
- (A) q và s
- (B) p, q, x và s
- (C) p
- D p và q

- (E) x, q và s
  - 16. Viết lại biểu thức trung tố (infix) sau sang dạng hậu tố (postfix) Polish? Giả sử là tất cả các phép toán đều có 2 toán hạng và có độ ưu tiên và tính kết hợp như trong ngôn ngữ lập trình C.

$$a + b * c - d * e + f$$

(A) 
$$a + b c * d e * - f +$$

(B) 
$$a b + c * d - e * f +$$

(C) a b c 
$$^*$$
 + d - e f  $^*$  +

(D) a b c 
$$*$$
 + d e  $*$  + f -



17. Cho khai báo sau được viết trên ngôn ngữ lập trình Ada:

Cho kích thước của các kiểu *Integer*, *Float*, *Boolean và Enumeration* lần lượt là 2, 4, 1, và 2, và kích thước cho phần mô tả kiểu (type description) là 0. Cho biết kích thước của môt đối tương kiểu **Figure**?

- (A) 21 B 13 (C) 23 (D) 15 (E) Khác
  - 18. Cơ chế gọi chương trình nào là cơ chế gọi chương trình con cơ bản của Lập trình hướng sự kiện (Event-driven Programming)
- A Biến cố Xử lý biến cố (Exception)
- (B) Gọi trở về đơn giản (Simple Call Return)
- (C) Gọi đệ qui (Recursive call)
- (D) Trình công hành (Coroutine)
- (E) Trình định thời (Scheduled Subprogram)
  - 19. Đoạn code viết bằng Scala nào dưới đây có thể dùng như một ví dụ minh họa cho khái niêm Hàm Currying
- (A)  $\operatorname{def} \operatorname{add1}(\operatorname{n:Int}) = \operatorname{n} + 1$
- (B)  $List(1,2,3).foldLeft(0)(_ + _ )$
- (C)  $\operatorname{def} \operatorname{add}(\operatorname{n:Int}) = (\operatorname{x:Int}) = n + \operatorname{x;val} \operatorname{add}2 = \operatorname{add}(2)$
- D def add(n:Int)(x:Int) = n + x; val add2 = add (2)
- (E) List(1,2,3).filter( > 1)
- 20. Liệt kê theo thứ tự chiều dài tăng dần ít nhất 5 chuỗi ngắn nhất (nếu có nhiều hơn 5 chuỗi) của ngôn ngữ được mô tả bởi biểu thức chính quy (regular expression) sau: ab\*(a|b)? a
- (A) aa, aba, aaa, abaa, aaba (B)  $\epsilon$ , a, aa, aba, abaa (C) aa, aba, aaa, abaa, abba
- (D) a, aa, aba, aaa, aaba (E) aa, aba, aaa, abaa, aaaa
- 21. Cho giá trị ban đầu của biến **c** là 3, cho biết những giá trị **có thể có** của biến **a** sau khi thực hiện phép gán sau trên ngôn ngữ lập trình C? a = c \* (c = 5);
- (A) 15, 25, 9 (B) 15 (C) 25 (D) 9 E 15, 25



- 22. Cho biết kích thước của đối tượng dữ liệu x được khai báo như sau: x: set of 1..16
- (A) 4 bytes
- B 2 bytes
- (C) 4 bits
- (D) 1 byte

- (E) 16 bytes
- 23. Cho lớp(class) A là lớp cha (superclass) của lớp B và B là lớp cha của hai lớp C và D. Trong lớp A, có khai báo phương thức thực thể (instance method) foo và phương thức này bị ghi đề(override) trên tất cả các lớp B, C và D. Cho biến x được khai báo như sau: Bx;

Qui ước viết  $func_T$  là phương thức func được khai báo trong lớp T, cho biết tập của phương thức đích của cuộc gọi sau: **x.foo()**;

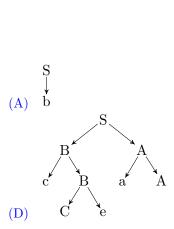
- (A)  $\{foo_C, foo_D\}$
- $\begin{array}{c} \text{(B)} \ \{foo_A, foo_B, foo_C, foo_D\} \\ \hline \text{E} \ \{foo_B, foo_C, foo_D\} \end{array}$

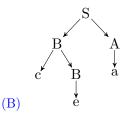
- (D)  $\{foo_A, foo_B\}$
- 24. Chon phát biểu ĐÚNG trong các phát biểu về dãy (array) sau:
- (A) Trên C, kích thước của dãy có thể tăng thêm sau khi dãy đã được tạo ra
- (B) Truy xuất một phần tử bên ngoài một dãy C không được phép, sẽ lập tức gây ra lỗi
- (C) Trên C và C++, kích thước của một dãy được lưu trữ để kiểm tra động
- (D) Trên Java, kích thước của một dãy KHÔNG được lưu trữ để kiểm tra động
- E Truy xuất một phần tử bên ngoài một dãy Java không được phép, sẽ lập tức gây ra lỗi

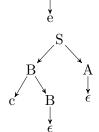
Cho văn phạm sau dùng cho các câu 25–26:

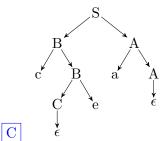
- S BA
- Α a A  $\epsilon$
- В Се c B
- $\mathbf{C}$ d C
- 25. Chuỗi nào KHÔNG thuộc ngôn ngữ được mô tả bởi văn phạm trên?
- (A) e
- (B) b
- (C) eaaa
- (D) cccddddeaa
- E cccaaa
- 26. Chọn cây phân tích cú pháp (parse tree) cho chuỗi nhập: cea

(E)











Phần hướng dẫn này áp dụng cho các câu 27–31

Hãy viết thêm các đoạn mã cần thiết vào phương thức dưới đây để thực hiện sinh mã cho phát biểu *repeat*. Nhắc lại phát biểu *repeat* kết thúc quá trình lặp khi điều kiện của phát biểu trở thành **true**. AST của phát biểu **repeat** được định nghĩa như sau: case class Repeat(val stmts:List[Stmt],val exp:Expr) extends Stmt

```
override def visitRepeat(ast: Repeat, o:Context)= {
        val ctxt = o.asInstanceOf[SubBody]
        ctxt.frame.enterLoop();
        val labelStart = ctxt.frame.getNewLabel()
        val labelBreak = ctxt.frame.getBreakLabel()
        val labelCont = ctxt.frame.getContinueLabel()
        val str1 = //1
        // sinh mã cho từng phát biểu trong thân của repeat
        val str2 = //2
        val str3 = //3
        // sinh mã cho biểu thức điều kiện
        val str4 = visit(ast.exp,....)
        val str5 = //4
        val str6 = //5
        frame().exitLoop();
        str1 + str2 + str3 + str4 + str5 + str6
   }
```

- 27. Đoạn mã nào trong số các đoạn mã dưới đây cần phải xuất hiện ở vị trí //1 trong đoạn mã trên
- (A) ""

- B ctxt.emit.emitLABEL(labelStart)
- (C) ctxt.emit.emitLABEL(labelBreak)
- (D) ctxt.emit.emitLABEL(labelCont)
- (E) ctxt.emit.emitGOTO(labelStart)
- 28. Đoạn mã nào trong số các đoạn mã dưới đây cần phải xuất hiện ở vị trí //2 trong đoạn mã trên
- (A) visit(ast.stmts,o)
- (B) ast.stmts.map(x = > visit(x, o))
- (C) ast.stmts.filter(x = > visit(x,o))
- D ast.stmts.foldLeft("")((x,y)=>x + visit(y,o))
- (E) ast.stmts.foldLeft("")((x,y)=>y + visit(x,o))
- 29. Đoạn mã nào trong số các đoạn mã dưới đây cần phải xuất hiện ở vị trí //3 trong đoạn mã trên
- (A) ""

- (B) ctxt.emit.emitIFFALSE(labelStart)
- (C) ctxt.emit.emitLABEL(labelBreak)
- D ctxt.emit.emitLABEL(labelCont)
- (E) ctxt.emit.emitGOTO(labelStart)
- 30. Đoạn mã nào trong số các đoạn mã dưới đây cần phải xuất hiện ở vị trí //4 trong đoạn mã trên
- (A) ctxt.emit.emitGOTO(labelBreak)
- B ctxt.emit.emitIFFALSE(labelStart)
- (C) ctxt.emit.emitLABEL(labelBreak)
- (D) ctxt.emit.emitLABEL(labelCont)
- (E) ctxt.emit.emitGOTO(labelStart)



- 31. Đoạn mã nào trong số các đoạn mã dưới đây cần phải xuất hiện ở vị trí //5 trong đoạn mã trên
- (A) ctxt.emit.emitGOTO(labelBreak)) (B) ctxt.
  - (B) ctxt.emit.emitIFFALSE(labelStart))
- C ctxt.emit.emitLABEL(labelBreak))
- (D) ctxt.emit.emitLABEL(labelCont))
- (E) ctxt.emit.emitGOTO(labelStart))
  - 32. Trình định thời (scheduled subprograms) thường được dùng trong
- (A) Lập trình song song (Parallel Programming)
- (B) Lập trình hướng đối tượng (Object-Oriented Programming)
- (C) Lập trình hàm (Functional Programming)
- (D) Lập trình hướng sự kiện (Event-driven Programming)
- E | Lập trình thời gian thực (Real-time Programming)
- 33. Gọi append là hàm dùng để nối 2 danh sách thành 1 danh sách. Ví dụ append([1,2,3],[4,5,6]) sẽ có kết quả là [1,2,3,4,5,6]. Hãy hiện thực hàm append(a:List[Int],b:List[Int]) dùng hàm bậc cao (high-order function)?
- (A) a.map(b)((x,y)=>x::y)

(B) a.foldLeft(b)((x,y)=>x::y)

C a.foldRight(b)((x,y)=>x::y)

(D) b.foldLeft(a)((x,y)=>x::y)

(E) b.foldRight(a)((x,y)=>x::y)

Đoạn mã sau được dùng trong các câu 34–35:

```
int *p = new int;
void foo(int * r) {
    delete r;
}
foo(p); //1
*p = 2; //2
```

- 34. Hiện tượng gì xảy ra khi p được truyền cho r ở phát biểu //1 trong đoạn mã trên:
- (A) Con trỏ chưa khai báo (undeclared pointer)
- B Bí danh (alias)
- (C) Khai báo trùng tên (redeclared)
- (D) Tham chiếu treo (dangling reference)
- (E) Rác (garbage)
- 35. Hiện tượng gì xảy ra khi thực thi phát biểu //2 trong đoạn mã trên:
- A Tham chiếu treo
- (B) Bí danh

(C) Da hình (polymorphism)

(D) Rác

(E) Con trỏ chưa khai báo

Đoạn mã sau được dùng cho các câu 36–41.

Cho một đoạn chương trình được viết trên một ngôn ngữ tựa C như sau:

```
int A[5] = {1,3,5,7,9};// index of A starts from 0
int j = 0;
int n = 5;
int sumAndIncrease(int a, int i) {
   int s = 0;
   for (; i < n; i = i + 1) {
      s = s + a;
   }
}</pre>
```



```
A[j] = A[j] + 1;
      return s;
void main(){
      int s = sumAndIncrease(A[j], j);
      printf("a = \%i \setminus n", s); //1
      printf("\%i\%i\%i\%i\%i\%i^*,A[0],A[1],A[2],A[3],A[4]);//2
}
 36. Nếu a và i được truyền bằng trị-kết quả (passed by value-result), giá trị của biến s
     được in ra ở phát biểu //1 là bao nhiêu?
A 5
                   (B) 10
                                                         (D) 25
                                      (C) 15
                                                                             (E) Khác
 37. Nếu a và i được truyền bằng trị-kết quả (passed by value-result), giá trị của biến A
     được in ra ở phát biểu //2 là bao nhiêu?
                                      C 6 3 5 7 9
                                                         (D) 5 3 5 7 9
(A) 1 3 5 7 9
                   (B) 2 4 6 8 10
                                                                             (E) Khác
 38. Nếu a và i được truyền bằng tham khảo (passed by reference), giá trị của biến s được
     in ra ở phát biểu //1 là bao nhiêu?
(A) 5
                   (B) 10
                                      (C) 15
                                                         (D) 25
                                                                             E Khác
 39. Nếu a và i được truyền bằng tham khảo (passed by reference), giá tri của biến A
     được in ra ở phát biểu //2 là bao nhiêu?
                   B 2 4 6 8 10
                                      (C) 6 3 5 7 9
                                                         (D) 5 3 5 7 9
(A) 1 3 5 7 9
                                                                             (E) Khác
 40. Nếu a và i được truyền bằng tên (passed by name), giá trị của biến s được in ra ở phát
     biểu //1 là bao nhiêu?
(A) 5
                   (B) 15
                                      (C) 20
                                                          D 25
                                                                             (E) Khác
 41. Nếu a và i được truyền bằng tên (passed by name), giá trị của biến A được in ra ở
     phát biểu //2 là bao nhiêu?
                   B 2 4 6 8 10
(A) 1 3 5 7 9
                                      (C) 6 3 5 7 9
                                                         (D) 5 3 5 7 9
                                                                             (E) Khác
 42. Một biến trên ngôn ngữ Javascript có thể nhân những giá tri thuộc các kiểu khác nhau.
     Ví du:
     var x = "def";
     x = 1;
     Ràng buộc giữa một biến và kiểu của nó trên ngôn ngữ Javascript xảy ra trong thời gian
                                                B Chay (running)
(A) Hiện thực (implementation)
                                               (D) Dich (compiling)
(C) Lập trình (programming)
(E) Khởi động (loading)
```

43. Cho  $X = \{a,b\}$ . Chọn biểu thức chính quy (regular expression) mô tả ngôn ngữ chứa bất kỳ chuỗi nào được tạo bởi các ký tư trên tập X nhưng KHÔNG chứa chuỗi có 2 ký tư a liên tiếp.

D b\*a?(bb\*a)\*b\* (A)  $a|(abb^*)^*$ (B) (b\*ab\*)\*(C)  $b^*(abb^*)^*$ 

(E) b\*ab\*ab\*



44. Kiểu *union* của ngôn ngữ C++ cho phép ghi vào ở thành phần này nhưng đọc lên ở thành phần khác như trong ví dụ sau:

```
union {
    int songuyen;
    char kytu[2];
} x;
x.songuyen = 12;
cout << x.kytu[0] << " " << x.kytu[1] << endl;</pre>
```

Đặc tính này của ngôn ngữ C++ gây ảnh hưởng XẤU NHẤT đối với

- (A) tính trực giao (orthonogality)
- (B) tính đơn giản (simplicity)

(C) tính dễ viết (writability)

tính tin cậy (reliability)

- (E) chi phí (cost)
  - 45. Cơ chế gọi chương trình con nào mà điều khiển có thể chuyển vào một vị trí ở giữa chương trình con được gọi thay vì chuyển vào đầu chương trình con được gọi?
- (A) Trình định thời (Scheduled Subprogram)
- (B) Gọi trở về đơn giản (Simple Call Return)
- (C) Gọi đệ qui (Recursive call)
- (D) Biến cố Xử lý biến cố (Exception)
- E Trình cộng hành (Coroutine)
- 46. Để thực hiện tìm kiếm một chuỗi (String) trên nhiều loại danh sách khác nhau, hãy chọn giải pháp thích hợp nhất để điền vào chỗ trống trong định nghĩa hàm sau:

```
def lookup[T](x:String,lst:List[T],f:T=>String):Option[T] =
   lst match {
     case List() => None
     case h::t =>______
}
```

- (A) if (h == x) Some(h) else lookup(x,t,f)
- $B \mid \text{if } (f(h) == x) \text{ Some}(h) \text{ else lookup}(x,t,f)$
- (C) if (h == f(x) Some(h) else lookup(x,t,f)
- (D) if (f(h) == x) Some(f(h)) else lookup(x,t,f)
- (E) if (h == f(x)) Some(f(h)) else lookup(x,t,f)



47. Khi cần khai thác đặc tính của kiểu union(các thành phần loại trừ lẫn nhau), làm thế nào để hiện thực trên những ngôn ngữ lập trình hướng đối tượng như Java(không có kiểu union)? Cụ thể làm thế nào để hiện thực biến z kiểu union sau trên những ngôn ngữ này?

```
union {
    string loikhen;
    string loiche;
} z;
```

- (A) Định nghĩa lớp cha A và hai lớp con B (có thuộc tính lo<br/>ikhen) và C (có thuộc tính lo<br/>iche), và z có kiểu A
- (B) Định nghĩa một lớp A có 2 thuộc tính loikhen và loiche; và z có kiểu A
- (C) Định nghĩa lớp A có thuộc tính lo<br/>ikhen và lớp B(con của A) có thuộc tính loiche, và z có kiểu A
- (D) Định nghĩa lớp A có thuộc tính lo<br/>ikhen và lớp B(con của A) có thuộc tính loiche, và z có kiểu B
- Dịnh nghĩa lớp trừu tượng (abstract class) A và 2 lớp con B (có thuộc tính loikhen) và C (có thuộc tính loiche), và z có kiểu A

### II. Phần câu hỏi tự luận: (dành cho tất cả sinh viên) (2 điểm)

48. Trên những ngôn ngữ lập trình cho phép rút ngắn tính toán (short-circuit evaluation) biểu thức luận lý, biểu thức sau sẽ không gây ra lỗi khi thực thi.

```
((a!=0) \&\& ((b/a) > 5))
```

Hãy dùng **if then else** để mô phỏng quá trình tính toán của biểu thức luận lý trên?

```
Lời giải. if ((a != 0) \text{ then if } ((b / a) > 8) \text{ then ...}
Giải pháp đề xuất đúng: 2
```

49. Hãy nêu những giải pháp đã được sử dụng để tránh tham chiếu treo (dangling reference) trong trường hợp trả về hàm như sau:

```
void->int F() {
    int x = 1;
    int g() {
        return x + 1;
    }
    return g;
}
void->int gg = F();
int z = gg();
```




50. Hãy viết lại biểu thức ở dạng trung tố (infix) sau sang dạng biểu thức tiền tố (prefix) Cambridge Polish:

$$a + b * c * d - e - f$$

Biết rằng độ ưu tiên và tính kết hợp của các phép toán trong biểu thức như thông thường (đều kết hợp trái và \* có ưu tiên cao hơn +, -). Trong biểu thức tiền tố nhiều toán hạng, thứ tự tính toán cũng từ trái sang phải.

Yêu cầu: Biểu thức dạng tiền tố Cambridge Polish phải thỏa các yêu cầu sau:

- Thứ tự xuất hiện các toán hạng trong biểu thức dạng tiềm tố phải có cùng thứ tự xuất hiện như trong biểu thức trung tố
- Số ( và ) là ít nhất
- Có cùng thứ tự tính toán các phép toán với thứ tự đó trong biểu thức dạng trung tố

**Lời giải.** (+ a (- (\* b c d) e f))

Nếu viết đúng biểu thức dạng tiền tố Cambridge Polish: 1 Nếu viết đúng và có số () ít nhất: 1

# III. Phần bài tập lớn: (dành cho tất cả sinh viên)

- 51. Viết hàm visitBlock(ast:Block,o:Context) để thực hiện kiểm tra kiểu. Hàm visitBlock sẽ thực hiện chức năng xây dựng bảng danh hiệu (symbol table) và truyền bảng này khi visit các phát biểu trong khối. Yêu cầu giải thích rõ về Context cụ thể được sử dụng trong hàm và ý tưởng mã của hàm. Nhắc lại AST của Block được khai báo như sau: case class Block(val decl:List[Decl],val stmts:List[Stmt]) extends Stmt
- 52. Viết hàm visitIf(ast:If,o:Context) để thực hiện sinh mã cho một phát biểu If. Nhắc lại AST của If được khai báo như sau:
  case class If(val expr:Expr, val thenStmt:Stmt, val elseStmt:Option[Stmt]) extends Stmt
  Yêu cầu giải thích rõ về Context cụ thể được sử dụng trong hàm và ý tưởng mã của hàm.



### IV. Phần KSTN: (dành cho sinh viên KSTN)

- 53. Viết hàm visitIf(ast:If,o:Context) để thực hiện kiểm tra lỗi FunctionNotReTurn? Yêu cầu giải thích ý tưởng thực hiện thể hiện qua mã của hàm?
- 54. Viết hàm visitBlock(ast:Block,o:Context) để thực hiện sinh mã cho một khối (block)? Yêu cầu giải thích ý tưởng thực hiện sinh mã cho các khai báo array trong một khối?

Question Distribution

Content	Level 1	Level 2	Level 3	Level 4
Introduction		1		
Lexical	1	4	5	
Syntax	2	4	7	
AST		2		1
Total	3	11	12	1

-HẾT

Chủ nhiệm bộ môn	Giảng viên ra đề
Chữ kí:	Chữ kí:
Họ tên:	Họ tên:



KHOA <b>KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH</b>	Họ và tê
BÔ MÔN <b>KHOA HOC MÁY TÍNH</b>	MSSV:

Họ và tế	in:	••••	 • • • • •	 	 	
MSSV:			 	 	 	

# Đề thi cuối kỳ

Môn thi: Nguyên Lý Ngôn Ngữ Lập Trình

Thời gian: 120 phút Ngày thi: 23-12-2015

□ Sinh viên được phép sử dụng tài liệu □ Sinh viên không được sử dụng tài liệu

Mã đề: 1152

- Sinh viên phải ghi tên và mã số sinh viên trên đề thi (**trang đầu và trang cuối**), giấy làm bài trắc nghiệm (giấy đỏ) và giấy làm bài. Sinh viên phải tô phần mã đề thi và mã số sinh viên trên giấy làm bài trắc nghiệm. Khi nộp bài, sinh viên phải nộp cả **đề thi, giấy làm bài trắc nghiệm và giấy làm bài**.
- Phần trắc nghiệm sẽ được chấm TỰ ĐỘNG trên giấy làm bài trắc nghiệm. Do đó, phần trắc nghiệm nếu làm trên đề thi sẽ KHÔNG được chấm.
- Đối với các câu hỏi phần trắc nghiệm, sinh viên chỉ chọn MỘT phương án đúng nhất.
- Đối với phần câu hỏi tự luận (phần II), sinh viên làm ngay trên đề thi, ở phần dành riêng ngay dưới mỗi câu hỏi.
- Đối với câu hỏi bài tập lớn (phần III và IV), sinh viên làm trên giấy làm bài.
- Sinh viên lớp đại trà làm 3 phần (I, II và III). Sinh viên lớp tài năng làm cả 4 phần (I, II, III và IV). Chỉ có phần I và II được dùng để tính điểm cuối kỳ, các phần III và IV để tính điểm bài tập lớn 2 và 3.

## I. Phần câu hỏi trắc nghiệm:(8 điểm))

Phần trình bày sau dùng trong các câu hỏi 1–2:

Cho mã giả của phát biểu  $for\ var = expr1\ to\ expr2\ do\ body$  như sau:

start: var = expr1

etemp = expr2

loop: if (var >= etemp) goto out

body

label1: var++

label2: goto loop

out:

1. Nếu trong thân body có thực thi lệnh continue thì điều khiển sẽ chuyển đến thực thi phát biểu có nhãn là:

- (A) out
- (B) label2
- (C) start
- (D) loop
- (E) label1



2. Với hiện thực của phát biểu for như trên, phát biểu nào đặt trong thân phát biểu for dưới đây sẽ làm cho phát biểu này bị lặp mãi mãi?

```
n = 10; s = 0;
     for i = s to n do _____
(A) i = i - 1;
                                  (B) s = s - 1;
                                                                     (C) n = n + 1;
```

- (D) câu A và C đúng
- (E) Không phát biểu nào trong các câu A, B và C làm phát biểu for lặp mãi mãi
  - 3. Cơ chế gọi chương trình nào là cơ chế gọi chương trình con cơ bản của Lập trình hướng sự kiện (Event-driven Programming)
- (A) Biến cố Xử lý biến cố (Exception)
- (B) Trình đinh thời (Scheduled Subprogram)
- (C) Goi trở về đơn giản (Simple Call Return)
- (D) Gọi đệ qui (Recursive call)
- (E) Trình cộng hành (Coroutine)
  - 4. Cho biết kích thước của đối tượng dữ liệu  $\mathbf{x}$  được khai báo như sau: x: set of 1..16
- (A) 4 bytes
- (B) 16 bytes
- (C) 2 bytes
- (D) 4 bits
- (E) 1 byte

5. Cho khai báo sau được viết trên ngôn ngữ lập trình Ada:

```
type Shape is (Circle, Triangle, Rectangle);
type Colors is (Red, Green, Blue);
type Figure (Form: Shape) is record
        Filled: Boolean;
        Color: Colors;
        case Form is
                when Circle => Diameter: Float;
                when Triangle =>
                         Leftside, Rightside: Integer;
                         Angle: Float;
                when Rectangle => Side1, Side2: Integer;
        end case;
end record;
```

Cho kích thước của các kiểu Integer, Float, Boolean và Enumeration lần lượt là 2, 4, 1, và 2, và kích thước cho phần mô tả kiểu (type description) là 0. Cho biết kích thước của một đối tượng kiểu **Figure**?

- (A) 15
- (B) 21
- (C) 13
- (D) 23
- (E) Khác



6. Để thực hiện tìm kiếm một chuỗi (String) trên nhiều loại danh sách khác nhau, hãy chọn giải pháp thích hợp nhất để điền vào chỗ trống trong định nghĩa hàm sau:

```
def lookup[T](x:String,lst:List[T],f:T=>String):Option[T] =
    lst match {
      case List() => None
      case h::t =>______
}
```

- (A) if (h == x) Some(h) else lookup(x,t,f)
- (B) if (h == f(x)) Some(f(h)) else lookup(x,t,f)
- (C) if (f(h) == x) Some(h) else lookup(x,t,f)
- (D) if (h == f(x) Some(h) else lookup(x,t,f)
- (E) if (f(h) == x) Some(f(h)) else lookup(x,t,f)

Cho văn phạm sau dùng cho các câu 7-8:

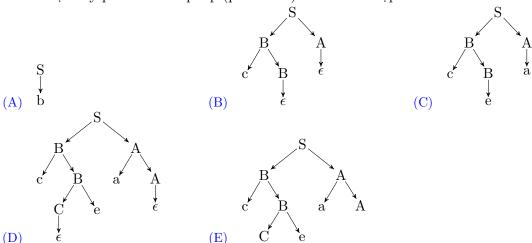
$$B \rightarrow C e \mid c B$$

 $C \rightarrow dC \mid \epsilon$ 

7. Chuỗi nào KHÔNG thuộc ngôn ngữ được mô tả bởi văn phạm trên?

- (A) e
- (B) cccaaa
- (C) b
- (D) eaaa
- (E) cccddddeaa

8. Chọn cây phân tích cú pháp (parse tree) cho chuỗi nhập: cea



Đoạn mã sau được dùng cho các câu 9–14.

Cho một đoan chương trình được viết trên một ngôn ngữ tưa C như sau:



```
return s;
}
void main(){
    int s = sumAndIncrease(A[j],j);
    printf("a = %i\n",s); //1
    printf(" %i %i %i %i %i \n",A[0],A[1],A[2],A[3],A[4]); //2
}
```

9. Nếu a và i được truyền bằng **trị-kết quả (passed by value-result)**, giá trị của biến s được in ra ở phát biểu //1 là bao nhiêu?

- (A) 25
- (B) 5
- (C) 10
- (D) 15
- (E) Khác

10. Nếu a và i được truyền bằng **trị-kết quả (passed by value-result)**, giá trị của biến A được in ra ở phát biểu //2 là bao nhiêu?

- (A) 5 3 5 7 9
- (B) 1 3 5 7 9
- (C) 2 4 6 8 10
- (D) 6 3 5 7 9
- (E) Kháo

11. Nếu a và i được truyền bằng **tham khảo (passed by reference)**, giá trị của biến s được in ra ở phát biểu //1 là bao nhiêu?

- (A) 25
- (B) 5
- (C) 10
- (D) 15
- (E) Khác

12. Nếu a và i được truyền bằng **tham khảo (passed by reference)**, giá trị của biến A được in ra ở phát biểu //2 là bao nhiêu?

- (A) 5 3 5 7 9
- (B) 1 3 5 7 9
- (C) 2 4 6 8 10
- (D) 6 3 5 7 9
- (E) Khác

13. Nếu a và i được truyền bằng **tên (passed by name)**, giá trị của biến s được in ra ở phát biểu //1 là bao nhiêu?

- (A) 25
- (B) 5
- (C) 15
- (D) 20
- (E) Khác

14. Nếu a và i được truyền bằng **tên (passed by name)**, giá trị của biến A được in ra ở phát biểu //2 là bao nhiêu?

- (A) 5 3 5 7 9
- (B) 1 3 5 7 9
- (C) 2 4 6 8 10
- (D) 6 3 5 7 9
- (E) Khác

15. Kiểu *union* của ngôn ngữ C++ cho phép ghi vào ở thành phần này nhưng đọc lên ở thành phần khác như trong ví dụ sau:

```
union {
    int songuyen;
    char kytu[2];
} x;
x.songuyen = 12;
cout << x.kytu[0] << " " << x.kytu[1] << endl;</pre>
```

Đặc tính này của ngôn ngữ C++ gây ảnh hưởng XẤU NHẤT đối với

- (A) tính trưc giao (orthonogality)
- (B) chi phí (cost)

(C) tính đơn giản (simplicity)

(D) tính dễ viết (writability)

(E) tính tin cậy (reliability)

. . .



(A)

...

16. Cho biết mã Jasmin của biểu thức viết bằng ngôn ngữ C như sau:

```
a * b - 2
với a và b là 2 biến nguyên có chỉ số lần lượt là 0 và 1.
iload 0
                   imul
                                      iload 0
                                                         iload 0
                                                                             iload 0
iload 1
                                      iload 1
                   load 0
                                                         imul
                                                                             iload 1
imul
                   iload 1
                                      iconst 2
                                                         iload 1
                                                                             iconst 2
                                                    (D)
                                                                       (E)
iconst 2
                   isub
                                      imul
                                                         isub
                                                                             isub
isub
                   iconst 2
                                      isub
                                                         iconst_2
                                                                             imul
```

17. Viết lại biểu thức trung tố (infix) sau sang dạng hậu tố (postfix) Polish? Giả sử là tất cả các phép toán đều có 2 toán hạng và có độ ưu tiên và tính kết hợp như trong ngôn ngữ lập trình C.

```
a + b * c - d * e + f
(A) a + b c * d e * - f +
(B) a b c * + d e * - f +
(C) a b + c * d - e * f +
(D) a b c * + d - e f * +
(E) a b c * + d e * + f -
```

Phần hướng dẫn này áp dụng cho các câu 18–22

Hãy viết thêm các đoạn mã cần thiết vào phương thức dưới đây để thực hiện sinh mã cho phát biểu *repeat*. Nhắc lại phát biểu *repeat* kết thúc quá trình lặp khi điều kiện của phát biểu trở thành **true**. AST của phát biểu **repeat** được định nghĩa như sau: case class Repeat(val stmts:List[Stmt],val exp:Expr) extends Stmt

```
override def visitRepeat(ast: Repeat, o:Context)= {
        val ctxt = o.asInstanceOf[SubBody]
        ctxt.frame.enterLoop();
        val labelStart = ctxt.frame.getNewLabel()
        val labelBreak = ctxt.frame.getBreakLabel()
        val labelCont = ctxt.frame.getContinueLabel()
        val str1 = //1
        // sinh mã cho từng phát biểu trong thân của repeat
        val str2 = //2
        val str3 = //3
        // sinh mã cho biểu thức điều kiện
        val str4 = visit(ast.exp,...)
        val str5 = //4
        val str6 = //5
        frame().exitLoop();
        str1 + str2 + str3 + str4 + str5 + str6
   }
```

- 18. Đoạn mã nào trong số các đoạn mã dưới đây cần phải xuất hiện ở vị trí //1 trong đoạn mã trên
- (A) "" (B) ctxt.emit.emitGOTO(labelStart) (C) ctxt.emit.emitLABEL(labelStart) (D) ctxt.emit.emitLABEL(labelBreak)
- (E) ctxt.emit.emitLABEL(labelCont)



- 19. Đoạn mã nào trong số các đoạn mã dưới đây cần phải xuất hiện ở vị trí //2 trong đoạn mã trên
- (A) visit(ast.stmts,o)
- (B) ast.stmts.foldLeft("")((x,y)=>y + visit(x,o))
- (C) ast.stmts.map(x = > visit(x, o))
- (D) ast.stmts.filter(x = > visit(x, o))
- (E) ast.stmts.foldLeft("")((x,y)=>x + visit(y,o))
  - 20. Đoạn mã nào trong số các đoạn mã dưới đây cần phải xuất hiện ở vị trí //3 trong đoạn mã trên
- (A) ""

- (B) ctxt.emit.emitGOTO(labelStart)
- (C) ctxt.emit.emitIFFALSE(labelStart)
- (D) ctxt.emit.emitLABEL(labelBreak)
- (E) ctxt.emit.emitLABEL(labelCont)
- 21. Đoạn mã nào trong số các đoạn mã dưới đây cần phải xuất hiện ở vi trí //4 trong đoạn mã trên
- (A) ctxt.emit.emitGOTO(labelBreak)
- (B) ctxt.emit.emitGOTO(labelStart)
- (C) ctxt.emit.emitIFFALSE(labelStart)
- (D) ctxt.emit.emitLABEL(labelBreak)
- (E) ctxt.emit.emitLABEL(labelCont)
  - 22. Đoan mã nào trong số các đoan mã dưới đây cần phải xuất hiện ở vi trí //5 trong đoan mã trên
- (A) ctxt.emit.emitGOTO(labelBreak))
- (B) ctxt.emit.emitGOTO(labelStart))
- (C) ctxt.emit.emitIFFALSE(labelStart))
- (D) ctxt.emit.emitLABEL(labelBreak))
- (E) ctxt.emit.emitLABEL(labelCont))
- 23. Cho lớp(class) A là lớp cha (superclass) của lớp B và B là lớp cha của hai lớp C và D. Trong lớp A, có khai báo phương thức thực thể (instance method) foo và phương thức này bị ghi đè(override) trên tất cả các lớp B, C và D. Cho biến x được khai báo như sau:  $\mathbf{B} \mathbf{x};$

Qui ước viết  $func_T$  là phương thức func được khai báo trong lớp T, cho biết tập của phương thức đích của cuộc gọi sau: **x.foo()**;

- (A)  $\{foo_C, foo_D\}$
- (B)  $\{foo_B, foo_C, foo_D\}$  (C)  $\{foo_A, foo_B, foo_C, foo_D\}$

(D)  $\{foo_B\}$ 

- (E)  $\{foo_A, foo_B\}$
- 24. Liệt kệ theo thứ tư chiều dài tăng dần ít nhất 5 chuỗi ngắn nhất (nếu có nhiều hơn 5 chuỗi) của ngôn ngữ được mô tả bởi biểu thức chính quy (regular expression) sau: ab\*(a|b)? a
- (A) aa, aba, aaa, abaa, aaba
- (B) aa, aba, aaa, abaa, aaaa
- (C)  $\epsilon$ , a, aa, aba, abaa

- (D) aa, aba, aaa, abaa, abba
- (E) a, aa, aba, aaa, aaba

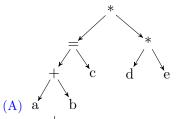
Cho văn pham của một biểu thức sau dùng cho các câu 25–27:

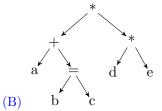
```
\rightarrow term '=' exp | term
   exp
                     \operatorname{term} '+'\operatorname{fact} | \operatorname{term} '>'\operatorname{fact} | \operatorname{fact}
  term
                     ope '*' ope | ope
   fact
                     '(' exp ')' | ID
   ope
với ID là một danh hiệu.
```

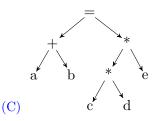


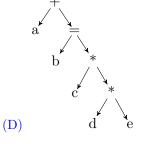
- 25. Chọn biểu thức được viết đúng văn phạm, có số lượng dấu '(' và ')' ít nhất mà tương đương với biểu thức sau: (a + (b = ((c \* d) \* e))) > f
- (A) (a + (b = (c \* d) \* e)) > f
- (B) a + (b = c \* d \* e) > f
- (C) (a + (b = ((c \* d) \* e))) > f
- (D) a + b = c \* d \* e > f

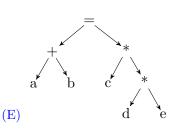
- (E) a + (b = (c \* d) \* e) > f
- 26. Vẽ AST cho biểu thức sau được viết dựa vào văn phạm trên:  $\mathbf{a} + \mathbf{b} = \mathbf{c} * (\mathbf{d} * \mathbf{e})$











27. Cho cấu trúc AST được khai báo trên Scala như sau: trait Exp case class Bin(op:String,e1:Exp,e2:Exp) extends Exp case class Id(i:String) extends Exp

Chọn AST cho biểu thức sau:  $\mathbf{a} + \mathbf{b} > \mathbf{c} = \mathbf{d} * \mathbf{e} = \mathbf{f}$ 

- $(A) \ \operatorname{Bin}("=",\operatorname{Bin}(">",\operatorname{Bin}("+",\operatorname{Id}("a"),\operatorname{Id}("b")),\operatorname{Id}("c")),\operatorname{Bin}("*",\operatorname{Id}("d"),\operatorname{Bin}("=",\operatorname{Id}("e"),\operatorname{Id}("f"))))$
- $\mathrm{(B)} \ \mathrm{Bin}("=",\!\mathrm{Bin}("=",\!\mathrm{Bin}("+",\!\mathrm{Id}("a"),\!\mathrm{Bin}(">",\!\mathrm{Id}("b"),\!\mathrm{Id}("c"))),\!\mathrm{Bin}("*",\!\mathrm{Id}("d"),\!\mathrm{Id}("e"))),\!\mathrm{Id}("f"))$
- (C) Bin("=",Bin("=",Bin(">",Bin("+",Id("a"),Id("b")),Id("c")),Bin("\*",Id("d"),Id("e"))),Id("f"))(D) Bin("=",Bin(">",Bin("+",Id("a"),Id("b")),Id("c")),Bin("=",Bin("\*",Id("d"),Id("e"))),Id("f"))
- (E) Bin("=",Bin("+",Id("a"),Bin(">",Id("b"),Id("c"))),Bin("=",Bin("\*",Id("d"),Id("e"))),Id("f"))
- 28. Cơ chế gọi chương trình con nào mà điều khiển có thể chuyển vào một vị trí ở giữa chương trình con được gọi thay vì chuyển vào đầu chương trình con được gọi?
- (A) Trình định thời (Scheduled Subprogram)
- (B) Trình công hành (Coroutine)
- (C) Gọi trở về đơn giản (Simple Call Return)
- (D) Gọi đệ qui (Recursive call)
- (E) Biến cố Xử lý biến cố (Exception)
- 29. Giả sử chương trình có một lỗi văn phạm nằm sau lệnh **print**. Khi em nhấn nút Run (để dịch và chạy chương trình) trên trình soạn thảo, kết quả được in ra bởi lệnh **print** trên trước khi lỗi văn phạm được báo. Hỏi chế độ dịch của trình soạn thảo là gì?
- (A) trình liên kết (link editor)
- (B) trình biên dich đông(just-in-time compiler)
- (C) trình biên dich (compiler)

(D) trình thông dịch(interpreter)

(E) trình hợp ngữ (assembler)



30. Cho giá trị ban đầu của biến  $\mathbf{c}$  là 3, cho biết những giá trị  $\mathbf{c}$   $\mathbf{\acute{o}}$  thể  $\mathbf{\acute{c}}$   $\mathbf{\acute{o}}$  của biến  $\mathbf{a}$  sau khi thực hiện phép gán sau trên ngôn ngữ lập trình  $\mathbf{C}$ ?

```
a = c * (c = 5);
```

- (A) 15, 25, 9
- (B) 15, 25
- (C) 15
- (D) 25
- (E) 9

31. Đoạn code viết bằng Scala nào dưới đây có thể dùng như một ví dụ minh họa cho khái niệm Hàm Currying

```
(A) \operatorname{def} \operatorname{add1}(n:\operatorname{Int}) = n + 1
```

- (B)  $List(1,2,3).filter(_ > 1)$
- (C) List(1,2,3).foldLeft(0)(+)
- (D)  $\operatorname{def} \operatorname{add}(\operatorname{n:Int}) = (\operatorname{x:Int}) = n + \operatorname{x;val} \operatorname{add} 2 = \operatorname{add}(2)$
- (E) def add(n:Int)(x:Int) = n + x; val add2 = add (2)

Đoạn mã sau được dùng trong các câu 32–33:

```
int *p = new int;
void foo(int * r) {
    delete r;
}
foo(p); //1
*p = 2; //2
```

- 32. Hiện tượng gì xảy ra khi p được truyền cho r ở phát biểu //1 trong đoạn mã trên:
- (A) Con trỏ chưa khai báo (undeclared pointer) (B) Rác (garbage)
- (C) Bí danh (alias)

- (D) Khai báo trùng tên (redeclared)
- (E) Tham chiếu treo (dangling reference)
- 33. Hiện tượng gì xảy ra khi thực thi phát biểu //2 trong đoạn mã trên:
- (A) Tham chiếu treo
- (B) Con trỏ chưa khai báo
- (C) Bí danh

- (D) Da hình (polymorphism)
- (E) Rác
- 34. Khi cần khai thác đặc tính của kiểu union(các thành phần loại trừ lẫn nhau), làm thế nào để hiện thực trên những ngôn ngữ lập trình hướng đối tượng như Java(không có kiểu union)? Cụ thể làm thế nào để hiện thực biến z kiểu union sau trên những ngôn ngữ này?

```
union {
    string loikhen;
    string loiche;
} z;
```

- (A) Định nghĩa lớp cha A và hai lớp con B (có thuộc tính loikhen) và C (có thuộc tính loiche), và z có kiểu A
- (B)  $\theta$  Định nghĩa lớp trừu tượng (abstract class) A và 2 lớp con B (có thuộc tính loikhen) và C (có thuộc tính loiche), và z có kiểu A
- (C) Đinh nghĩa một lớp A có 2 thuộc tính loikhen và loiche; và z có kiểu A
- (D) Định nghĩa lớp A có thuộc tính loikhen và lớp B(con của A) có thuộc tính loiche, và z có kiểu A
- (E)  $\frac{\text{Dịnh nghĩa lớp A có thuộc tính loikhen và lớp B(con của A) có thuộc tính loiche, và z có kiểu B}$



- 35. Trình định thời (scheduled subprograms) thường được dùng trong
- (A) Lập trình song song (Parallel Programming)
- (B) Lập trình thời gian thực (Real-time Programming)
- (C) Lập trình hướng đối tượng (Object-Oriented Programming)
- (D) Lập trình hàm (Functional Programming)
- (E) Lập trình hướng sự kiện (Event-driven Programming)
  - 36. Cho biết mã Jasmin của phát biểu gán sau:

```
a[i] = a[i] + 2
```

với i kiểu nguyên có chỉ số là 3 và a là kiểu dãy nguyên có chỉ số là 1.

```
aload 1
                                                                                   aload 1
                         aload_1
     iastore
                                                               iload 3
                                            aload 1
                                                                                   dup
     iadd
                         iload 3
                                            iload 3
                                                               aload 1
                                                                                   iload 3
     iaload
                         iastore
                                            iaload
                                                               iload 3
                                                                                   dup
     aload 1
                         aload 1
                                            iconst 2
                                                          (D)
                                                                                   iaload
(A)
                                                               iaload
                                                                              (E)
     iload 3
                         iload 3
                                            iadd
                                                               iconst 2
                                                                                   iconst 2
     dup
                         iconst 2
                                                               iadd
                                                                                   iadd
                                            iastore
                         iadd
     iconst 2
                                                               iastore
                                                                                   iastore
     ...
                                                                                   . . .
```

- 37. Chọn phát biểu ĐÚNG trong các phát biểu về dãy (array) sau:
- (A) Trên C, kích thước của dãy có thể tăng thêm sau khi dãy đã được tạo ra
- (B) Truy xuất một phần tử bên ngoài một dãy Java không được phép, sẽ lập tức gây ra lỗi
- (C) Truy xuất một phần tử bên ngoài một dãy C không được phép, sẽ lập tức gây ra lỗi
- (D) Trên C và C++, kích thước của một dãy được lưu trữ để kiểm tra động
- (E) Trên Java, kích thước của một dãy KHÔNG được lưu trữ để kiểm tra động

Đoạn code sau dùng cho các câu 38–40

```
int p;
int* foo(int x) {
    static int q;
    int *s = new int;
    switch (x) {
        case 1: return &p;
        case 2: return &q;
        case 3: return &x;
        case 4: return s;
        default: return foo(x-1);
    }
}
```

- 38. Phát biểu nào sẽ gây ra lỗi tham chiếu treo (dangling reference) khi thực thi
- (A) return s
- (B) return &p
- (C) return &q
- (D) return &x

- (E) Không phát biểu nào gây ra lỗi trên khi thực thi
- 39. Trong đoạn mã trên, đối tượng dữ liệu nào có thể trở thành rác (garbage)
- (A) trỏ đến bởi s
- (B) p

(C) q

(D) x

(E) Không có đối tượng nào có thể trở thành rác



40. Khi hàm **foo** được gọi đệ quy (recursive) thì các bản hoạt động của **foo** dùng chung những đối tượng dữ liệu nào?

```
(A) q và s
```

```
(B) x, q và s
```

```
(C) p, q, x và s
```

```
(D) p
```

(E) p và q

41. Gọi append là hàm dùng để nối 2 danh sách thành 1 danh sách. Ví dụ append([1,2,3],[4,5,6]) sẽ có kết quả là [1,2,3,4,5,6]. Hãy hiện thực hàm append(a:List[Int],b:List[Int]) dùng hàm bậc cao (high-order function)?

```
(A) a.map(b)((x,y)=>x::y)
```

(B) b.foldRight(a)((x,y)=>x::y)

(C) a.foldLeft(b)((x,y)=>x::y)

(D) a.foldRight(b)((x,y)=>x::y)

- (E) b.foldLeft(a)((x,y)=>x::y)
  - 42. Một biến trên ngôn ngữ Javascript có thể nhận những giá trị thuộc các kiểu khác nhau. Ví du:

```
var x = "def"; 
 x = 1:
```

Ràng buộc giữa một biến và kiểu của nó trên ngôn ngữ Javascript xảy ra trong thời gian

- (A) Hiện thực (implementation)
- (B) Khởi động (loading)

(C) Chay (running)

(D) Lập trình (programming)

(E) Dich (compiling)

Đoạn code sau, được viết trên ngôn ngữ dùng **qui tắc tầm vực tĩnh (static-scope rule)**, áp dụng cho các câu 43–46

43. Môi trường tham khảo tĩnh (static referencing environment) của thủ tục  ${f sub3}$  KHÔNG chứa

- (A) sub2
- (B) a  $\dot{\sigma} / / 1$
- (C) a  $\dot{\sigma}$  //2
- (D) sub1
- (E) sub3
- 44. Giả sử chuỗi gọi là main  $\to$  sub<br/>1  $\to$  sub2  $\to$  sub3, tham khảo đến **a** trong **sub3** ứng với khai báo
- (A) Báo lỗi a chưa khai báo
- (B) a trong main (//1)
- (C) a trong sub1 (//2)

- (D) a trong sub2 (//3)
- (E) a trong sub3
- 45. Giả sử chuỗi gọi là main  $\to$  sub1  $\to$  sub2  $\to$  sub3 , tham khảo đến  $\bf a$  trong sub3  $\bf khi$  thực thi ứng với kết hợp của  $\bf a$  trong bản hoạt động của
- (A) main
- (B) Báo lỗi không tìm thấy a
- (C) sub1
- (D) sub2

(E) sub3



46.	Nếu đoạn code trên được viết trên ngôn ngữ dùng qui tắc tầm vực động (dynamic-scope
	rule), và giả sử chuỗi gọi là main $\to \mathrm{sub1} \to \mathrm{sub2} \to \mathrm{sub3}$ , tham khảo đến a trong sub3
	khi thực thi ứng với kết hợp của a trong bản hoạt động của

- (A) Báo lỗi không tìm thấy a
- (B) sub3
- (C) main
- (D) sub1

- (E) sub2
- 47. Cho  $X = \{a,b\}$ . Chọn biểu thức chính quy (regular expression) mô tả ngôn ngữ chứa bất kỳ chuỗi nào được tạo bởi các ký tự trên tập X nhưng  $KH\^ONG$  chứa chuỗi có 2 ký tự a liên tiếp.
- (A) b\*a?(bb\*a)\*b\* (B) a|(abb\*)\* (C) (b\*ab\*)\* (D) b\*(abb\*)\* (E) b\*ab\*ab\* II. Phần câu hỏi tự luận: (dành cho tất cả sinh viên) (2 điểm)
  - 48. Hãy nêu những giải pháp đã được sử dụng để tránh tham chiếu treo (dangling reference) trong trường hợp trả về hàm như sau:

```
void->int F() {
    int x = 1;
    int g() {
        return x + 1;
    }
    return g;
}

void->int gg = F();
int z = gg();
```

49. Trên những ngôn ngữ lập trình cho phép rút ngắn tính toán (short-circuit evaluation) biểu thức luận lý, biểu thức sau sẽ không gây ra lỗi khi thực thi.

$$((a == 0) || ((b / a) > 8))$$

Hãy dùng **if then else** để mô phỏng quá trình tính toán của biểu thức luận lý trên?



50.	Hãy	viết	lại	biểu	$th\acute{u}c$	ở	dạng	trung	${\rm t\acute{o}}$	(infix)	sau	sang	dạng	biểu	$th\acute{u}c$	${\rm ti}\grave{\rm e}{\rm n}$	tố	(prefix)
	Cam	brid	ge I	Polish	:													

$$a * b * c + d + e - f$$

Biết rằng độ ưu tiên và tính kết hợp của các phép toán trong biểu thức như thông thường (đều kết hợp trái và \* có ưu tiên cao hơn +, -). Trong biểu thức tiền tố nhiều toán hạng, thứ tự tính toán cũng từ trái sang phải.

Yêu cầu: Biểu thức dạng tiền tố Cambridge Polish phải thỏa các yêu cầu sau:

- Thứ tự xuất hiện các toán hạng trong biểu thức dạng tiềm tố phải có cùng thứ tự xuất hiện như trong biểu thức trung tố
- Số ( và ) là ít nhất
- Có cùng thứ tự tính toán các phép toán với thứ tự đó trong biểu thức dạng trung tố

## III. Phần bài tập lớn: (dành cho tất cả sinh viên)

- 51. Viết hàm visitBlock(ast:Block,o:Context) để thực hiện kiểm tra kiểu. Hàm visitBlock sẽ thực hiện chức năng xây dựng bảng danh hiệu (symbol table) và truyền bảng này khi visit các phát biểu trong khối. Yêu cầu giải thích rõ về Context cụ thể được sử dụng trong hàm và ý tưởng mã của hàm. Nhắc lại AST của Block được khai báo như sau: case class Block(val decl:List[Decl],val stmts:List[Stmt]) extends Stmt
- 52. Viết hàm visitIf(ast:If,o:Context) để thực hiện sinh mã cho một phát biểu If. Nhắc lại AST của If được khai báo như sau:
  case class If(val expr:Expr, val thenStmt:Stmt, val elseStmt:Option[Stmt]) extends Stmt Yêu cầu giải thích rõ về Context cụ thể được sử dụng trong hàm và ý tưởng mã của hàm.

#### IV. Phần KSTN: (dành cho sinh viên KSTN)

- 53. Viết hàm visitIf(ast:If,o:Context) để thực hiện kiểm tra lỗi FunctionNotReTurn? Yêu cầu giải thích ý tưởng thực hiện thể hiện qua mã của hàm?
- 54. Viết hàm visitBlock(ast:Block,o:Context) để thực hiện sinh mã cho một khối (block)? Yêu cầu giải thích ý tưởng thực hiện sinh mã cho các khai báo array trong một khối?

—————————————————————————————————————	
Họ và tên:	Diểm:
MSSV:	





Chủ nhiệm bộ môn	Giảng viên ra đề
Chữ kí:	Chữ kí:
Họ tên:	Họ tên:



# KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH $\mbox{BỘ MÔN KHOA HỌC MÁY TÍNH} \label{eq:homographical}$

	ĐÁI	P ÁN cho	Đề thi	cuối kỳ	
	<b>thi: Nguyên Lý Ngôn</b> gian: 120 phút	Ngữ Lập Tr	ình		
	nh viên được phép sử dụng tề: 1152	g tài liệu	⊠ Sin	nh viên không đ	lược sử dụng tài liệu
I. Ph	ần câu hỏi trắc nghiệ	m:(8 điểm))	)		
	trình bày sau dùng trong c nã giả của phát biểu for vo			pdy như sau:	
loop: label	<pre>: var = expr1   etemp = expr2   if (var &gt;= etemp) g   body 1: var++ 2: goto loop</pre>	oto out			
	Nếu trong thân <i>body</i> có th biểu có nhãn là:	ực thi lệnh <i>cor</i>	tinue thì đ	tiều khiển sẽ chư	uyển đến thực thi phá
	ut (B) label2 Với hiện thực của phát biểu đây sẽ làm cho phát biểu r	u $for$ như trên,	_	, , _	E label1 hân phát biểu <b>for</b> dướ
	n = 10;s = 0; for i = s to n do				
(D) cá	= i - 1; àu A và C đúng Thông phát biểu nào trong	(B) s = các câu A, B $\cdot$	,	hát biểu <b>for</b> lặp	(C) $n = n + 1$ ; o mãi mãi
	Cơ chế gọi chương trình n sự kiện (Event-driven Prog	_	oi chương t	rình con cơ bản	n của Lập trình hướng
(B) T (C) G (D) G	iến cố - Xử lý biến cố (Ex rình định thời (Scheduled ọi trở về đơn giản (Simple ọi đệ qui (Recursive call) rình cộng hành (Coroutine	Subprogram) e Call - Return	)		
	Cho biết kích thước của đỏ x: set of 116	ối tượng dữ liệ	u <b>x</b> được k	hai báo như sau	1:

C 2 bytes

(D) 4 bits

(E) 1 byte

(B) 16 bytes

(A) 4 bytes



5. Cho khai báo sau được viết trên ngôn ngữ lập trình Ada:

Cho kích thước của các kiểu *Integer, Float, Boolean và Enumeration* lần lượt là 2, 4, 1, và 2, và kích thước cho phần mô tả kiểu (type description) là 0. Cho biết kích thước của một đối tượng kiểu **Figure**?

- (A) 15
- (B) 21
- C 13
- (D) 23
- (E) Khác
- 6. Để thực hiện tìm kiếm một chuỗi (String) trên nhiều loại danh sách khác nhau, hãy chọn giải pháp thích hợp nhất để điền vào chỗ trống trong định nghĩa hàm sau:

- (A) if (h == x) Some(h) else lookup(x,t,f)
- (B) if (h == f(x)) Some(f(h)) else lookup(x,t,f)
- C if (f(h) == x) Some(h) else lookup(x,t,f)
- (D) if (h == f(x) Some(h) else lookup(x,t,f)
- (E) if (f(h) == x) Some(f(h)) else lookup(x,t,f)

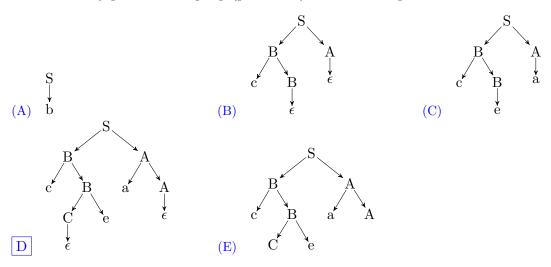
Cho văn phạm sau dùng cho các câu 7–8:

7. Chuỗi nào KHÔNG thuộc ngôn ngữ được mô tả bởi văn phạm trên?

- (A) e
- B cccaaa
- (C) b
- (D) eaaa
- (E) cccddddeaa



8. Chọn cây phân tích cú pháp (parse tree) cho chuỗi nhập: cea



Đoạn mã sau được dùng cho các câu 9–14.

Cho một đoạn chương trình được viết trên một ngôn ngữ tựa C như sau:

```
int A[5] = {1,3,5,7,9}; // index of A starts from 0
int j = 0;
int n = 5;
int sumAndIncrease(int a, int i) {
    int s = 0;
    for ( ; i < n; i = i + 1) {
        s = s + a;
        A[j] = A[j] + 1;
    }
    return s;
}

void main(){
    int s = sumAndIncrease(A[j],j);
    printf("a = %i\n",s); //1
    printf("%i %i %i %i %i\n",A[0],A[1],A[2],A[3],A[4]); //2
}</pre>
```

- 9. Nếu *a* và *i* được truyền bằng **trị-kết quả (passed by value-result)**, giá trị của biến *s* được in ra ở phát biểu //1 là bao nhiêu?
- (A) 25
- B 5
- (C) 10
- (D) 15
- (E) Khác
- 10. Nếu *a* và *i* được truyền bằng **trị-kết quả (passed by value-result)**, giá trị của biến *A* được in ra ở phát biểu //2 là bao nhiêu?
- (A) 5 3 5 7 9
- (B) 1 3 5 7 9
- (C) 2 4 6 8 10
- D 63579
- (E) Khác
- 11. Nếu a và i được truyền bằng **tham khảo (passed by reference)**, giá trị của biến s được in ra ở phát biểu //1 là bao nhiêu?
- (A) 25
- (B) 5
- (C) 10
- (D) 15
- E Khác



12. Nếu a và i được truyền bằng tham khảo (passed by reference), giá trị của biến A được in ra ở phát biểu //2 là bao nhiêu?

(A) 5 3 5 7 9

(B) 1 3 5 7 9

C 2 4 6 8 10

(D) 6 3 5 7 9

(E) Khác

13. Nếu a và i được truyền bằng **tên (passed by name)**, giá trị của biến s được in ra ở phát biểu //1 là bao nhiêu?

A 25

(B) 5

(C) 15

(D) 20

(E) Khác

14. Nếu a và i được truyền bằng tên (passed by name), giá trị của biến A được in ra ở phát biểu //2 là bao nhiêu?

(A) 5 3 5 7 9

(B) 1 3 5 7 9

C 2 4 6 8 10

(D) 6 3 5 7 9

(E) Khác

15. Kiểu union của ngôn ngữ C++ cho phép ghi vào ở thành phần này nhưng đọc lên ở thành phần khác như trong ví du sau:

```
union {
    int songuyen;
    char kytu[2];
} x;
x.songuyen = 12;
cout << x.kytu[0] << " " << x.kytu[1] << endl;</pre>
```

Đặc tính này của ngôn ngữ C++ gây ảnh hưởng XÂU NHẤT đối với

(A) tính trực giao (orthonogality)

(B) chi phí (cost)

(C) tính đơn giản (simplicity)

(D) tính dễ viết (writability)

E tính tin cậy (reliability)

16. Cho biết mã Jasmin của biểu thức viết bằng ngôn ngữ C như sau:

với a và b là 2 biến nguyên có chỉ số lần lượt là 0 và 1.

iload 0 iload 1 imul iconst 2 isub

...

imul load 0 iload 1

isub

iconst 2

iload 1 iconst 2 imul

isub

. . .

iload 0

iload 1 (D) isub iconst 2

iload 0

imul

iconst 2 (E) isub imul

...

iload 0

iload 1

17. Viết lại biểu thức trung tố (infix) sau sang dạng hậu tố (postfix) Polish? Giả sử là tất cả các phép toán đều có 2 toán hạng và có độ ưu tiên và tính kết hợp như trong ngôn ngữ lập trình C.

$$a + b * c - d * e + f$$

(A) a + b c \* d e \* - f +(D) a b c \* + d - e f \* +

B a b c \* + d e \* - f + (C) a b + c \* d - e \* f + (E) a b c \* + d e \* + f -

Phần hướng dẫn này áp dụng cho các câu 18–22

Hãy viết thêm các đoan mã cần thiết vào phương thức dưới đây để thực hiên sinh mã cho phát biểu repeat. Nhắc lai phát biểu repeat kết thúc quá trình lặp khi điều kiên của phát biểu trở thành **true**. AST của phát biểu **repeat** được định nghĩa như sau: case class Repeat(val stmts:List[Stmt],val exp:Expr) extends Stmt



```
override def visitRepeat(ast: Repeat, o:Context)= {
        val ctxt = o.asInstanceOf[SubBody]
        ctxt.frame.enterLoop();
        val labelStart = ctxt.frame.getNewLabel()
        val labelBreak = ctxt.frame.getBreakLabel()
        val labelCont = ctxt.frame.getContinueLabel()
        val str1 = //1
        // sinh mã cho từng phát biểu trong thân của repeat
        val str2 = //2
        val str3 = //3
        // sinh mã cho biểu thức điều kiện
        val str4 = visit(ast.exp,....)
        val str5 = //4
        val str6 = //5
        frame().exitLoop();
        str1 + str2 + str3 + str4 + str5 + str6
    }
 18. Đoạn mã nào trong số các đoạn mã dưới đây cần phải xuất hiện ở vị trí //1 trong đoạn
     mã trên
(A) ""
                                              (B) ctxt.emit.emitGOTO(labelStart)
ctxt.emit.emitLABEL(labelStart)
                                              (D) ctxt.emit.emitLABEL(labelBreak)
(E) ctxt.emit.emitLABEL(labelCont)
 19. Đoạn mã nào trong số các đoạn mã dưới đây cần phải xuất hiện ở vị trí //2 trong đoạn
     mã trên
(A) visit(ast.stmts,o)
(B) ast.stmts.foldLeft("")((x,y)=>y + visit(x,o))
(C) ast.stmts.map(x = > visit(x, o))
(D) ast.stmts.filter(x = > visit(x, o))
E ast.stmts.foldLeft("")((x,y)=>x + visit(y,o))
 20. Đoạn mã nào trong số các đoạn mã dưới đây cần phải xuất hiện ở vị trí //3 trong đoạn
     mã trên
(A) ""
                                              (B) ctxt.emit.emitGOTO(labelStart)
(C) ctxt.emit.emitIFFALSE(labelStart)
                                             (D) ctxt.emit.emitLABEL(labelBreak)
E ctxt.emit.emitLABEL(labelCont)
 21. Đoạn mã nào trong số các đoạn mã dưới đây cần phải xuất hiện ở vị trí //4 trong đoạn
     mã trên
(A) ctxt.emit.emitGOTO(labelBreak)
                                            (B) ctxt.emit.emitGOTO(labelStart)
ctxt.emit.emitIFFALSE(labelStart)
                                             (D) ctxt.emit.emitLABEL(labelBreak)
(E) ctxt.emit.emitLABEL(labelCont)
 22. Đoạn mã nào trong số các đoạn mã dưới đây cần phải xuất hiện ở vị trí //5 trong đoạn
     mã trên
```

(A) ctxt.emit.emitGOTO(labelBreak))(C) ctxt.emit.emitIFFALSE(labelStart))

(E) ctxt.emit.emitLABEL(labelCont))

(B) ctxt.emit.emitGOTO(labelStart))

D ctxt.emit.emitLABEL(labelBreak))



23. Cho lớp(class) A là lớp cha (superclass) của lớp B và B là lớp cha của hai lớp C và D. Trong lớp A, có khai báo phương thức thực thể (instance method) foo và phương thức này bị ghi đè(override) trên tất cả các lớp B, C và D. Cho biến x được khai báo như sau: B x;

Qui ước viết  $func_T$  là phương thức func được khai báo trong lớp T, cho biết tập của phương thức đích của cuộc gọi sau: **x.foo()**;

- (A)  $\{foo_C, foo_D\}$
- $\begin{array}{|c|c|c|c|c|}\hline \textbf{B} & \{foo_B, foo_C, foo_D\} \\ \hline \textbf{(E)} & \{foo_A, foo_B\} \\ \end{array}$

(D)  $\{foo_B\}$ 

- 24. Liệt kệ theo thứ tư chiều dài tăng dần ít nhất 5 chuỗi ngắn nhất (nếu có nhiều hơn 5 chuỗi) của ngôn ngữ được mô tả bởi biểu thức chính quy (regular expression) sau: ab\*(a|b)? a
- (A) aa, aba, aaa, abaa, aaba
- (B) aa, aba, aaa, abaa, aaaa
- (C)  $\epsilon$ , a, aa, aba, abaa

- D aa, aba, aaa, abaa, abba
- (E) a, aa, aba, aaa, aaba

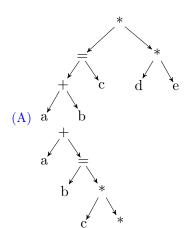
Cho văn phạm của một biểu thức sau dùng cho các câu 25–27:

- term '=' exp | term
- term '+' fact | term '>' fact | fact term
- ope '\*' ope | ope fact
- '(' exp ')' | ID ope

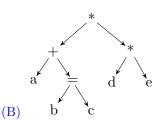
với ID là một danh hiệu.

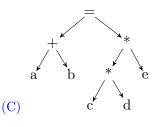
- 25. Chọn biểu thức được viết đúng văn phạm, có số lượng dấu '(' và ')' ít nhất mà tương đương với biểu thức sau:  $(\mathbf{a} + (\mathbf{b} = ((\mathbf{c} * \mathbf{d}) * \mathbf{e}))) > \mathbf{f}$
- (A) (a + (b = (c \* d) \* e)) > f(C) (a + (b = ((c \* d) \* e))) > f(E) a + (b = (c \* d) \* e) > f
- (B) a + (b = c \* d \* e) > f(D) a + b = c \* d \* e > f

- 26. Vẽ AST cho biểu thức sau được viết dựa vào văn phạm trên:  $\mathbf{a} + \mathbf{b} = \mathbf{c} * (\mathbf{d} * \mathbf{e})$

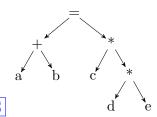


(D)





 $\mathbf{E}$ 





27. Cho cấu trúc AST được khai báo trên Scala như sau: trait Exp case class Bin(op:String,e1:Exp,e2:Exp) extends Exp case class Id(i:String) extends Exp

Chon AST cho biểu thức sau:  $\mathbf{a} + \mathbf{b} > \mathbf{c} = \mathbf{d} * \mathbf{e} = \mathbf{f}$ 

```
(A) Bin("=",Bin(">",Bin("+",Id("a"),Id("b")),Id("c")),Bin("*",Id("d"),Bin("=",Id("e"),Id("f"))))
```

- (B) Bin("=",Bin("=",Bin("+",Id("a"),Bin(">",Id("b"),Id("c"))),Bin("\*",Id("d"),Id("e"))),Id("f"))
- (C) Bin("=",Bin("=",Bin(">",Bin("+",Id("a"),Id("b")),Id("c")),Bin("\*",Id("d"),Id("e"))),Id("f"))
- D Bin("=",Bin(">",Bin("+",Id("a"),Id("b")),Id("c")),Bin("=",Bin("\*",Id("d"),Id("e"))),Id("f"))
- $(E) \ \operatorname{Bin}("=",\operatorname{Bin}("+",\operatorname{Id}("a"),\operatorname{Bin}(">",\operatorname{Id}("b"),\operatorname{Id}("c"))),\operatorname{Bin}("=",\operatorname{Bin}("*",\operatorname{Id}("d"),\operatorname{Id}("e"))),\operatorname{Id}("f"))$ 
  - 28. Cơ chế gọi chương trình con nào mà điều khiển có thể chuyển vào một vị trí ở giữa chương trình con được gọi thay vì chuyển vào đầu chương trình con được gọi?
- (A) Trình định thời (Scheduled Subprogram)
- B Trình cộng hành (Coroutine)
- (C) Goi trở về đơn giản (Simple Call Return)
- (D) Gọi đệ qui (Recursive call)
- (E) Biến cố Xử lý biến cố (Exception)
- 29. Giả sử chương trình có một lỗi văn phạm nằm sau lệnh **print**. Khi em nhấn nút Run (để dịch và chạy chương trình) trên trình soạn thảo, kết quả được in ra bởi lệnh **print** trên trước khi lỗi văn phạm được báo. Hỏi chế độ dịch của trình soạn thảo là gì?
- (A) trình liên kết (link editor)
- (B) trình biên dich đông(just-in-time compiler) (C) trình biên dich (compiler)
- D trình thông dịch(interpreter)

- (E) trình hợp ngữ (assembler)
- 30. Cho giá trị ban đầu của biến **c** là 3, cho biết những giá trị **có thể có** của biến **a** sau khi thực hiện phép gán sau trên ngôn ngữ lập trình C? a = c \* (c = 5);
- (A) 15, 25, 9
- B 15, 25
- (C) 15
- (D) 25
- (E) 9
- 31. Đoạn code viết bằng Scala nào dưới đây có thể dùng như một ví dụ minh họa cho khái niệm Hàm Currying
- (A) def add1(n:Int) = n + 1
- (B)  $List(1,2,3).filter(_ > 1)$
- (C) List(1,2,3).foldLeft $(0)(_ + _ )$
- (D)  $\operatorname{def} \operatorname{add}(\operatorname{n:Int}) = (\operatorname{x:Int}) => \operatorname{n} + \operatorname{x;val} \operatorname{add} 2 = \operatorname{add}(2)$
- $\mathbf{E}$  def add(n:Int)(x:Int) = n + x; val add2 = add (2)

Đoạn mã sau được dùng trong các câu 32–33:

```
int *p = new int;
void foo(int * r) {
    delete r;
}
foo(p); //1
```



\*p = 2; //2

- 32. Hiện tượng gì xảy ra khi p được truyền cho r ở phát biểu //1 trong đoạn mã trên:
- (A) Con trỏ chưa khai báo (undeclared pointer) (B) Rác (garbage)
- C Bí danh (alias) (D) Khai báo trùng tên (redeclared)
- (E) Tham chiếu treo (dangling reference)
- 33. Hiện tượng gì xảy ra khi thực thi phát biểu //2 trong đoạn mã trên:
- A Tham chiếu treo (B) Con trỏ chưa khai báo (C) Bí danh
- (D) Đa hình (polymorphism) (E) Rác
  - 34. Khi cần khai thác đặc tính của kiểu union(các thành phần loại trừ lẫn nhau), làm thế nào để hiện thực trên những ngôn ngữ lập trình hướng đối tượng như Java(không có kiểu union)? Cụ thể làm thế nào để hiện thực biến z kiểu union sau trên những ngôn ngữ này?

```
union {
    string loikhen;
    string loiche;
} z;
```

- (A)  $\begin{array}{c} {\rm Dinh\ nghĩa\ lớp\ cha\ A\ và\ hai\ lớp\ con\ B\ (có\ thuộc\ tính\ loikhen)\ và\ C\ (có\ thuộc\ tính\ loiche),} \\ {\rm và\ z\ có\ kiểu\ A} \end{array}$
- Binh nghĩa lớp trừu tượng (abstract class) A và 2 lớp con B (có thuộc tính loikhen) và C (có thuộc tính loiche), và z có kiểu A
- (C) Định nghĩa một lớp A có 2 thuộc tính loikhen và loiche; và z có kiểu A
- (D) Định nghĩa lớp A có thuộc tính lo<br/>ikhen và lớp B(con của A) có thuộc tính loiche, và z có kiểu A
- (E) Định nghĩa lớp A có thuộc tính lo<br/>ikhen và lớp B(con của A) có thuộc tính loiche, và z có kiểu B
- 35. Trình đinh thời (scheduled subprograms) thường được dùng trong
- (A) Lập trình song song (Parallel Programming)
- B Lập trình thời gian thực (Real-time Programming)
- (C) Lâp trình hướng đối tương (Object-Oriented Programming)
- (D) Lâp trình hàm (Functional Programming)
- (E) Lập trình hướng sự kiện (Event-driven Programming)
- 36. Cho biết mã Jasmin của phát biểu gán sau:

$$a[i] = a[i] + 2$$

với i kiểu nguyên có chỉ số là 3 và a là kiểu dãy nguyên có chỉ số là 1.

```
aload 1
                                                                                  aload 1
                        aload 1
     iastore
                                           aload 1
                                                               iload 3
                                                                                  dup
     iadd
                        iload 3
                                           iload 3
                                                               aload 1
                                                                                  iload 3
     iaload
                        iastore
                                           iaload
                                                               iload 3
                                                                                  dup
                        aload 1
     aload 1
(A)
                                      (C)
                                           iconst 2
                                                          D
                                                               iaload
                                                                            (E)
                                                                                  iaload
                        iload 3
     iload 3
                                           iadd
                                                               iconst 2
                                                                                  iconst 2
                        iconst 2
     dup
                                           iastore
                                                               iadd
                                                                                  iadd
                        iadd
     iconst 2
                                                               iastore
                                                                                  iastore
                                            ...
```



- 37. Chon phát biểu ĐÚNG trong các phát biểu về dãy (array) sau:
- (A) Trên C, kích thước của dãy có thể tăng thêm sau khi dãy đã được tạo ra
- B Truy xuất một phần tử bên ngoài một dãy Java không được phép, sẽ lập tức gây ra lỗi
- (C) Truy xuất một phần tử bên ngoài một dãy C không được phép, sẽ lập tức gây ra lỗi
- (D) Trên C và C++, kích thước của một dãy được lưu trữ để kiểm tra động
- (E) Trên Java, kích thước của một dãy KHÔNG được lưu trữ để kiểm tra động

Đoạn code sau dùng cho các câu 38–40

```
int p;
int* foo(int x) {
      static int q;
      int *s = new int;
      switch (x) {
           case 1: return &p;
           case 2: return &q;
           case 3: return &x;
           case 4: return s;
           default: return foo(x-1);
       }
}
```

38. Phát biểu nào sẽ gây ra lỗi tham chiếu treo (dangling reference) khi thực thi

- (A) return s
- (B) return &p
- (C) return &q
- D return &x

- (E) Không phát biểu nào gây ra lỗi trên khi thực thi
  - 39. Trong đoạn mã trên, đối tượng dữ liệu nào có thể trở thành rác (garbage)
- A trỏ đến bởi s

(C) q

(D) x

- (E) Không có đối tương nào có thể trở thành rác
  - 40. Khi hàm **foo** được gọi đệ quy (recursive) thì các bản hoạt động của **foo** dùng chung những đối tượng dữ liệu nào?
- (A) q và s
- (B) x, q  $v\grave{a}$  s (C) p, q, x  $v\grave{a}$  s (D) p
- E p và q
- 41. Goi append là hàm dùng để nối 2 danh sách thành 1 danh sách. Ví du append([1,2,3],[4,5,6]) sẽ có kết quả là [1,2,3,4,5,6]. Hãy hiện thực hàm append(a:List[Int],b:List[Int]) dùng hàm bậc cao (high-order function)?
- (A) a.map(b)((x,y)=>x::y)

(B) b.foldRight(a)((x,y)=>x::y)

(C) a.foldLeft(b)((x,y)=>x::y)

D a.foldRight(b)((x,y)=>x::y)

- (E) b.foldLeft(a)((x,y)=>x::y)
  - 42. Một biến trên ngôn ngữ Javascript có thể nhân những giá tri thuộc các kiểu khác nhau. Ví du:

```
var x = "def";
x = 1:
```

Ràng buộc giữa một biến và kiểu của nó trên ngôn ngữ Javascript xảy ra trong thời gian

- (A) Hiện thực (implementation)
- (B) Khởi động (loading)

C Chay (running)

(D) Lập trình (programming)

(E) Dich (compiling)



Đoạn code sau, được viết trên ngôn ngữ dùng **qui tắc tầm vực tĩnh (static-scope rule)**, áp dụng cho các câu 43–46

43. Môi trường tham khảo tĩnh (static referencing environment) của thủ tục **sub3** KHÔNG chứa

- (A) sub2  $\boxed{\mathrm{B}}$  a  $\mathring{\sigma}$  //1 (C) a  $\mathring{\sigma}$  //2 (D) sub1 (E) sub3
  - 44. Giả sử chuỗi gọi là main  $\to$  sub<br/>1  $\to$  sub2  $\to$  sub3, tham khảo đến **a** trong **sub3** ứng với khai báo
- (A) Báo lỗi a chưa khai báo (B) a trong main (//1) C a trong sub1 (//2)
- (D) a trong sub2 (//3) (E) a trong sub3
  - 45. Giả sử chuỗi gọi là main  $\to$  sub1  $\to$  sub2  $\to$  sub3 , tham khảo đến  $\bf a$  trong sub3  $\bf khi$  thực thi ứng với kết hợp của  $\bf a$  trong bản hoạt động của
- (A) main (B) Báo lỗi không tìm thấy a  $\hbox{$\bf C$}$  sub1 (D) sub2 (E) sub3
- 46. Nếu đoạn code trên được viết trên ngôn ngữ dùng qui tắc tầm vực động (dynamic-scope rule), và giả sử chuỗi gọi là main  $\rightarrow$  sub1  $\rightarrow$  sub2  $\rightarrow$  sub3, tham khảo đến a trong sub3 **khi thực thi** ứng với kết hợp của a trong bản hoạt động của
- (A) Báo lỗi không tìm thấy a (B) sub3 (C) main (D) sub1  $\boxed{\text{E}}$  sub2
- 47. Cho  $X = \{a,b\}$ . Chọn biểu thức chính quy (regular expression) mô tả ngôn ngữ chứa bất kỳ chuỗi nào được tạo bởi các ký tự trên tập X nhưng  $KH\^ONG$  chứa chuỗi có 2 ký tự a liên tiếp.

# II. Phần câu hỏi tự luận: (dành cho tất cả sinh viên) (2 điểm)



48. Hãy nêu những giải pháp đã được sử dụng để tránh tham chiếu treo (dangling reference) trong trường hợp trả về hàm như sau:

```
void->int F() {
    int x = 1;
    int g() {
        return x + 1;
    }
    return g;
}
void->int gg = F();
int z = gg();
```

49. Trên những ngôn ngữ lập trình cho phép rút ngắn tính toán (short-circuit evaluation) biểu thức luận lý, biểu thức sau sẽ không gây ra lỗi khi thực thi.

```
((a == 0) || ((b / a) > 8))
```

Hãy dùng **if then else** để mô phỏng quá trình tính toán của biểu thức luận lý trên?

```
Lời giải. if (a == 0) then ... else if ((b / a) > 10) then ... Giải pháp đề xuất đúng: 2
```

50. Hãy viết lại biểu thức ở dạng trung tố (infix) sau sang dạng biểu thức tiền tố (prefix) Cambridge Polish:

```
a * b * c + d + e - f
```

Biết rằng độ ưu tiên và tính kết hợp của các phép toán trong biểu thức như thông thường (đều kết hợp trái và \* có ưu tiên cao hơn +, -). Trong biểu thức tiền tố nhiều toán hạng, thứ tự tính toán cũng từ trái sang phải.

Yêu cầu: Biểu thức dạng tiền tố Cambridge Polish phải thỏa các yêu cầu sau:



- Thứ tự xuất hiện các toán hạng trong biểu thức dạng tiềm tố phải có cùng thứ tự xuất hiện như trong biểu thức trung tố
- Số ( và ) là ít nhất
- Có cùng thứ tự tính toán các phép toán với thứ tự đó trong biểu thức dạng trung tố

**Lời giải.** (- (+ (\* a b c) d e) f)

Nếu viết đúng biểu thức dạng tiền tố Cambridge Polish: 1 Nếu viết đúng và có số () ít nhất: 1

### III. Phần bài tập lớn: (dành cho tất cả sinh viên)

- 51. Viết hàm visitBlock(ast:Block,o:Context) để thực hiện kiểm tra kiểu. Hàm visitBlock sẽ thực hiện chức năng xây dựng bảng danh hiệu (symbol table) và truyền bảng này khi visit các phát biểu trong khối. Yêu cầu giải thích rõ về Context cụ thể được sử dụng trong hàm và ý tưởng mã của hàm. Nhắc lại AST của Block được khai báo như sau: case class Block(val decl:List[Decl],val stmts:List[Stmt]) extends Stmt
- 52. Viết hàm visitIf(ast:If,o:Context) để thực hiện sinh mã cho một phát biểu If. Nhắc lại AST của If được khai báo như sau: case class If(val expr:Expr, val thenStmt:Stmt, val elseStmt:Option[Stmt]) extends Stmt Yêu cầu giải thích rõ về Context cụ thể được sử dụng trong hàm và ý tưởng mã của hàm.

## IV. Phần KSTN: (dành cho sinh viên KSTN)

- 53. Viết hàm visitIf(ast:If,o:Context) để thực hiện kiểm tra lỗi FunctionNotReTurn? Yêu cầu giải thích ý tưởng thực hiện thể hiện qua mã của hàm?
- 54. Viết hàm visitBlock(ast:Block,o:Context) để thực hiện sinh mã cho một khối (block)? Yêu cầu giải thích ý tưởng thực hiện sinh mã cho các khai báo array trong một khối?

Question Distribution

	& account	Distribut	1011	
Content	Level 1	Level 2	Level 3	Level 4
Introduction		1		
Lexical	1	4	5	
Syntax	2	4	7	
AST		2		1
Total	3	11	12	1

-HÊT-

Chủ nhiệm bộ môn	Giảng viên ra đề
Chữ kí:	Chữ kí:
Họ tên:	Họ tên:



KHOA <b>KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH</b>	Họ và tên:
BỘ MÔN <b>KHOA HỌC MÁY TÍNH</b>	MSSV:

# Đề thi cuối kỳ

Môn thi: Nguyên Lý Ngôn Ngữ Lập Trình

Thời gian: 120 phút Ngày thi: 23-12-2015

☐ Sinh viên được phép sử dụng tài liệu ☐ Sinh viên không được sử dụng tài liệu

Mã đề: 1153

- Sinh viên phải ghi tên và mã số sinh viên trên đề thi (**trang đầu và trang cuối**), giấy làm bài trắc nghiệm (giấy đỏ) và giấy làm bài. Sinh viên phải tô phần mã đề thi và mã số sinh viên trên giấy làm bài trắc nghiệm. Khi nộp bài, sinh viên phải nộp cả **đề thi, giấy làm bài trắc nghiệm và giấy làm bài**.
- Phần trắc nghiệm sẽ được chấm TỰ ĐỘNG trên giấy làm bài trắc nghiệm. Do đó, phần trắc nghiệm nếu làm trên đề thi sẽ KHÔNG được chấm.
- Đối với các câu hỏi phần trắc nghiệm, sinh viên chỉ chọn MỘT phương án đúng nhất.
- Đối với phần câu hỏi tự luận (phần II), sinh viên làm ngay trên đề thi, ở phần dành riêng ngay dưới mỗi câu hỏi.
- Đối với câu hỏi bài tập lớn (phần III và IV), sinh viên làm trên giấy làm bài.
- Sinh viên lớp đại trà làm 3 phần (I, II và III). Sinh viên lớp tài năng làm cả 4 phần (I, II, III và IV). Chỉ có phần I và II được dùng để tính điểm cuối kỳ, các phần III và IV để tính điểm bài tập lớn 2 và 3.

# I. Phần câu hỏi trắc nghiệm:(8 điểm))

- 1. Liệt kê theo thứ tự chiều dài tăng dần ít nhất 5 chuỗi ngắn nhất (nếu có nhiều hơn 5 chuỗi) của ngôn ngữ được mô tả bởi biểu thức chính quy (regular expression) sau: ab\*(a|b)? a
- (A) aa, aba, aaa, abaa, aaba (B)  $\epsilon$ , a, aa, aba, abaa (C) aa, aba, aaa, abaa, aaaa (D) aa, aba, aaa, abaa, abaa (E) a, aa, aba, aaa, aaba
- 2. Gọi append là hàm dùng để nối 2 danh sách thành 1 danh sách. Ví dụ append([1,2,3],[4,5,6]) sẽ có kết quả là [1,2,3,4,5,6]. Hãy hiện thực hàm append(a:List[Int],b:List[Int]) dùng hàm
- bậc cao (high-order function)?

  (A) a.map(b)((x,y)=>x::y)
  (B) a.foldLeft(b)((x,y)=>x::y)
  (C) b.foldRight(a)((x,y)=>x::y)
  (D) a.foldRight(b)((x,y)=>x::y)
- (E) b.foldLeft(a)((x,y)=>x::y)
  - 3. Đoạn code viết bằng Scala nào dưới đây có thể dùng như một ví dụ minh họa cho khái niêm Hàm Currying
- (A) def add1(n:Int) = n + 1
- (B) List(1,2,3).foldLeft(0)(+)
- (C) List(1,2,3).filter( > 1)
- (D)  $\operatorname{def} \operatorname{add}(\operatorname{n:Int}) = (\operatorname{x:Int}) => \operatorname{n} + \operatorname{x;val} \operatorname{add}2 = \operatorname{add}(2)$
- (E) def add(n:Int)(x:Int) = n + x; val add2 = add (2)



- 4. Cho giá trị ban đầu của biến  $\mathbf{c}$  là 3, cho biết những giá trị **có thể có** của biến  $\mathbf{a}$  sau khi thực hiện phép gán sau trên ngôn ngữ lập trình C?  $\mathbf{a} = \mathbf{c} * (\mathbf{c} = \mathbf{5});$
- (A) 15, 25, 9
- (B) 15
- (C) 15, 25
- (D) 25
- (E) 9

Cho văn phạm của một biểu thức sau dùng cho các câu 5–7:

$$\exp \rightarrow \text{term '='} \exp \mid \text{term}$$

$$term \rightarrow term '+' fact \mid term '>' fact \mid fact$$

$$\mathrm{fact} \quad \rightarrow \quad \mathrm{ope} \ '*', \ \mathrm{ope} \mid \mathrm{ope}$$

ope 
$$\rightarrow$$
 '(' exp ')' | ID

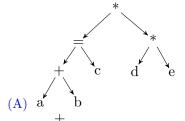
với ID là một danh hiệu.

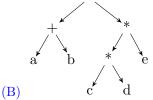
- 5. Chọn biểu thức được viết đúng văn phạm, có số lượng dấu '(' và ')' ít nhất mà tương đương với biểu thức sau: (a + (b = ((c \* d) \* e))) > f
- (A) (a + (b = (c \* d) \* e)) > f
- (B) (a + (b = ((c \* d) \* e))) > f

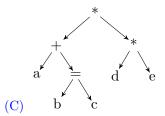
(C) a + (b = c \* d \* e) > f

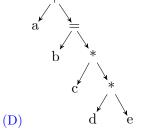
(D) a + b = c \* d \* e > f

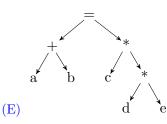
- (E) a + (b = (c \* d) \* e) > f
  - 6. Vẽ AST cho biểu thức sau được viết dựa vào văn phạm trên:  $\mathbf{a} + \mathbf{b} = \mathbf{c} * (\mathbf{d} * \mathbf{e})$











- 7. Cho cấu trúc AST được khai báo trên Scala như sau: trait Exp case class Bin(op:String.e1:Exp.e2:Exp) extends Exp
  - case class Bin(op:String,e1:Exp,e2:Exp) extends Exp case class Id(i:String) extends Exp

Chọn AST cho biểu thức sau:  $\mathbf{a} + \mathbf{b} > \mathbf{c} = \mathbf{d} * \mathbf{e} = \mathbf{f}$ 

- $(A) \ \operatorname{Bin}("=",\operatorname{Bin}(">",\operatorname{Bin}("+",\operatorname{Id}("a"),\operatorname{Id}("b")),\operatorname{Id}("c")),\operatorname{Bin}("*",\operatorname{Id}("d"),\operatorname{Bin}("=",\operatorname{Id}("e"),\operatorname{Id}("f"))))$
- (B) Bin("=",Bin("=",Bin(">",Bin("+",Id("a"),Id("b")),Id("c")),Bin("\*",Id("d"),Id("e"))),Id("f"))
- (C) Bin("=",Bin("=",Bin("+",Id("a"),Bin(">",Id("b"),Id("c"))),Bin("\*",Id("d"),Id("e"))),Id("f"))
- (D) Bin("=",Bin(">",Bin("+",Id("a"),Id("b")),Id("c")),Bin("=",Bin("\*",Id("d"),Id("e"))),Id("f"))
- (E) Bin("=",Bin("+",Id("a"),Bin(">",Id("b"),Id("c"))),Bin("=",Bin("\*",Id("d"),Id("e"))),Id("f"))



8. Cho lớp(class) A là lớp cha (superclass) của lớp B và B là lớp cha của hai lớp C và D. Trong lớp A, có khai báo phương thức thực thể (instance method) foo và phương thức này bị ghi đè(override) trên tất cả các lớp B, C và D. Cho biến x được khai báo như sau: B x;

Qui ước viết func<sub>T</sub> là phương thức func được khai báo trong lớp T, cho biết tập của phương thức đích của cuộc gọi sau: **x.foo()**;

- (B)  $\{foo_A, foo_B, foo_C, foo_D\}$  (C)  $\{foo_B, foo_C, foo_D\}$ (A)  $\{foo_C, foo_D\}$ (E)  $\{foo_A, foo_B\}$ (D)  $\{foo_B\}$ 
  - 9. Trình đinh thời (scheduled subprograms) thường được dùng trong
- (A) Lập trình song song (Parallel Programming)
- (B) Lập trình hướng đối tượng (Object-Oriented Programming)
- (C) Lập trình thời gian thực (Real-time Programming)
- (D) Lập trình hàm (Functional Programming)
- (E) Lập trình hướng sự kiện (Event-driven Programming)

Đoạn code sau, được viết trên ngôn ngữ dùng qui tắc tầm vực tĩnh (static-scope rule), áp dụng cho các câu 10–13

```
procedure main(){
    var a, b, c:integer; //1
    procedure sub1(a: integer) { //2
        procedure sub3();
        procedure sub2() {
            var a, c : real;//3
            sub3();
        procedure sub3() {
            a // use a
        }
        sub2();
    }
    sub1(3);
}
```

10. Môi trường tham khảo tĩnh (static referencing environment) của thủ tục sub3 KHÔNG chứa

- (A) a  $\dot{\sigma}$  //1
- (B) sub2
- (C) a  $\dot{\sigma}$  //2 (D) sub1
- (E) sub3
- 11. Giả sử chuỗi gọi là main  $\rightarrow$  sub1  $\rightarrow$  sub2  $\rightarrow$  sub3, tham khảo đến **a** trong **sub3** ứng với khai báo
- (A) Báo lỗi a chưa khai báo
- (B) a trong sub1 (//2)
- (C) a trong main (//1)

- (D) a trong sub2 (//3)
- (E) a trong sub3
- 12. Giả sử chuỗi goi là main  $\rightarrow$  sub1  $\rightarrow$  sub2  $\rightarrow$  sub3, tham khảo đến **a** trong sub3 **khi thực** thi ứng với kết hợp của a trong bản hoạt động của
- (A) main
- (B) sub1
- (C) Báo lỗi không tìm thấy a
- (D) sub2

(E) sub3



- 13. Nếu đoạn code trên được viết trên ngôn ngữ dùng qui tắc tầm vực động (dynamic-scope rule), và giả sử chuỗi gọi là main  $\rightarrow$  sub1  $\rightarrow$  sub2  $\rightarrow$  sub3, tham khảo đến a trong sub3 **khi thực thi** ứng với kết hợp của a trong bản hoạt động của
- (A) Báo lỗi không tìm thấy a
- (B) main
- (C) sub3
- (D) sub1

- (E) sub2
- 14. Kiểu *union* của ngôn ngữ C++ cho phép ghi vào ở thành phần này nhưng đọc lên ở thành phần khác như trong ví dụ sau:

```
union {
    int songuyen;
    char kytu[2];
} x;
x.songuyen = 12;
cout << x.kytu[0] << " " << x.kytu[1] << endl;</pre>
```

Đặc tính này của ngôn ngữ C++ gây ảnh hưởng XẤU NHẤT đối với

- (A) tính trực giao (orthonogality)
- (B) tính đơn giản (simplicity)

(C) chi phí (cost)

(D) tính dễ viết (writability)

- (E) tính tin cậy (reliability)
  - 15. Khi cần khai thác đặc tính của kiểu union(các thành phần loại trừ lẫn nhau), làm thế nào để hiện thực trên những ngôn ngữ lập trình hướng đối tượng như Java(không có kiểu union)? Cụ thể làm thế nào để hiện thực biến z kiểu union sau trên những ngôn ngữ này?

```
union {
    string loikhen;
    string loiche;
} z;
```

- (A) Định nghĩa lớp cha A và hai lớp con B (có thuộc tính lo<br/>ikhen) và C (có thuộc tính lo<br/>iche), và z có kiểu A
- (B) Định nghĩa một lớp A có 2 thuộc tính loikhen và loiche; và z có kiểu A
- (C) Định nghĩa lớp trừu tượng (abstract class) A và 2 lớp con B (có thuộc tính loikhen) và C (có thuộc tính loiche), và z có kiểu A
- (D) Định nghĩa lớp A có thuộc tính lo<br/>ikhen và lớp B(con của A) có thuộc tính lo<br/>iche, và z có kiểu A
- (E) Định nghĩa lớp A có thuộc tính lo<br/>ikhen và lớp B(con của A) có thuộc tính lo<br/>iche, và z có kiểu B



16. Để thực hiện tìm kiếm một chuỗi (String) trên nhiều loại danh sách khác nhau, hãy chọn giải pháp thích hợp nhất để điền vào chỗ trống trong định nghĩa hàm sau:

```
def lookup[T](x:String,lst:List[T],f:T=>String):Option[T] =
    lst match {
        case List() => None
        case h::t =>______
}

(A) if (h == x) Some(h) else lookup(x,t,f)
(B) if (f(h) == x) Some(h) else lookup(x,t,f)
(C) if (h == f(x)) Some(f(h)) else lookup(x,t,f)
(D) if (h == f(x) Some(h) else lookup(x,t,f)
```

- 17. Cơ chế gọi chương trình con nào mà điều khiển có thể chuyển vào một vị trí ở giữa chương trình con được gọi thay vì chuyển vào đầu chương trình con được gọi?
- (A) Trình định thời (Scheduled Subprogram)
- (B) Gọi trở về đơn giản (Simple Call Return)

(E) if (f(h) == x) Some(f(h)) else lookup(x,t,f)

- (C) Trình cộng hành (Coroutine)
- (D) Gọi đệ qui (Recursive call)
- (E) Biến cố Xử lý biến cố (Exception)

Đoạn mã sau được dùng cho các câu 18–23.

Cho một đoạn chương trình được viết trên một ngôn ngữ tựa C như sau:

```
int A[5] = \{1,3,5,7,9\}; // \ index \ of \ A \ starts \ from \ 0
int j = 0;
int n = 5;
int sumAndIncrease(int \ a, \ int \ i) \ \{
    int s = 0;
    for ( \ ; \ i < n; \ i = i + 1) \ \{
        s = s + a;
        A[j] = A[j] + 1;
    }
    return s;
}

void main()\{
    int s = sumAndIncrease(A[j], j);
    printf("a = %i\n",s); //1
        printf("%i %i %i %i %i\n",A[0],A[1],A[2],A[3],A[4]); //2
}
```

18. Nếu a và i được truyền bằng **trị-kết quả (passed by value-result)**, giá trị của biến s được in ra ở phát biểu //1 là bao nhiêu?

- (A) 5
- (B) 25
- (C) 10
- (D) 15
- (E) Khác
- 19. Nếu a và i được truyền bằng **trị-kết quả (passed by value-result)**, giá trị của biến A được in ra ở phát biểu //2 là bao nhiêu?
- (A) 1 3 5 7 9
- (B) 5 3 5 7 9
- (C) 2 4 6 8 10
- (D) 6 3 5 7 9
- (E) Khác



20. Nếu a và i được truyền bằng **tham khảo (passed by reference)**, giá trị của biến s được in ra ở phát biểu //1 là bao nhiêu?

- (A) 5
- (B) 25
- (C) 10
- (D) 15
- (E) Khác

21. Nếu a và i được truyền bằng **tham khảo (passed by reference)**, giá trị của biến A được in ra ở phát biểu //2 là bao nhiêu?

- (A) 1 3 5 7 9
- (B) 5 3 5 7 9
- (C) 2 4 6 8 10
- (D) 6 3 5 7 9
- (E) Khác

22. Nếu a và i được truyền bằng **tên (passed by name)**, giá trị của biến s được in ra ở phát biểu //1 là bao nhiêu?

- (A) 5
- (B) 25
- (C) 15
- (D) 20
- (E) Khác

23. Nếu a và i được truyền bằng **tên (passed by name)**, giá trị của biến A được in ra ở phát biểu //2 là bao nhiêu?

- (A) 1 3 5 7 9
- (B) 5 3 5 7 9
- (C) 2 4 6 8 10
- (D) 6 3 5 7 9
- (E) Khác

Cho văn phạm sau dùng cho các câu 24–25:

- $S \rightarrow b \mid BA$
- $A \rightarrow a A \mid \epsilon$
- $B \rightarrow C e \mid c B$
- $C \rightarrow dC \mid \epsilon$

24. Chuỗi nào KHÔNG thuộc ngôn ngữ được mô tả bởi văn phạm trên?

- (A) e
- (B) b
- (C) cccaaa
- (D) eaaa
- (E) cccddddeaa

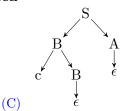
25. Chọn cây phân tích cú pháp (parse tree) cho chuỗi nhập: cea

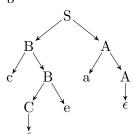
(E)



(D)

(B) e





B A A

26. Viết lại biểu thức trung tố (infix) sau sang dạng hậu tố (postfix) Polish? Giả sử là tất cả các phép toán đều có 2 toán hạng và có độ ưu tiên và tính kết hợp như trong ngôn ngữ lập trình C.

- a + b \* c d \* e + f
- (A) a + b c \* d e \* f +
- (B) a b + c \* d e \* f +
- (C) a b c \* + d e \* f +

- (D) a b c \* + d e f \* +
- (E) a b c \* + d e \* + f -

Đoạn code sau dùng cho các câu 27–29

int p;

int\* foo(int x) {



```
static int q;
      int *s = new int;
      switch (x) {
            case 1: return &p;
            case 2: return &q;
            case 3: return &x;
            case 4: return s;
            default: return foo(x-1);
        }
}
 27. Phát biểu nào sẽ gây ra lỗi tham chiếu treo (dangling reference) khi thực thi
(A) return &p
                         (B) return s
                                                  (C) return &q
                                                                          (D) return &x
(E) Không phát biểu nào gây ra lỗi trên khi thực thi
 28. Trong đoan mã trên, đối tương dữ liêu nào có thể trở thành rác (garbage)
                         (B) trỏ đến bởi s
                                                  (C) q
(E) Không có đối tượng nào có thể trở thành rác
```

- 20. Khi hàm **foo** được gọi đô quy (rocursiyo) th
  - 29. Khi hàm **foo** được gọi đệ quy (recursive) thì các bản hoạt động của **foo** dùng chung những đối tượng dữ liệu nào?
- $(A) \ q \ v\grave{a} \ s \qquad \qquad (B) \ p, \ q, \ x \ v\grave{a} \ s \qquad \qquad (C) \ x, \ q \ v\grave{a} \ s \qquad \qquad (D) \ p \qquad \qquad (E) \ p \ v\grave{a} \ q$ 
  - 30. Một biến trên ngôn ngữ Javascript có thể nhận những giá trị thuộc các kiểu khác nhau. Ví du:

```
var x = "def";
 x = 1:
```

Ràng buộc giữa một biến và kiểu của nó trên ngôn ngữ Javascript xảy ra trong thời gian

- (A) Hiện thực (implementation)
- (B) Chay (running)

(C) Khởi đông (loading)

(D) Lập trình (programming)

- (E) Dich (compiling)
- 31. Cơ chế gọi chương trình nào là cơ chế gọi chương trình con cơ bản của Lập trình hướng sự kiện (Event-driven Programming)
- (A) Biến cố Xử lý biến cố (Exception)
- (B) Gọi trở về đơn giản (Simple Call Return)
- (C) Trình định thời (Scheduled Subprogram)
- (D) Gọi đệ qui (Recursive call)
- (E) Trình cộng hành (Coroutine)
- 32. Chọn phát biểu ĐÚNG trong các phát biểu về dãy (array) sau:
- (A) Trên C, kích thước của dãy có thể tăng thêm sau khi dãy đã được tạo ra
- (B) Truy xuất một phần tử bên ngoài một dãy C không được phép, sẽ lập tức gây ra lỗi
- (C) Truy xuất một phần tử bên ngoài một dãy Java không được phép, sẽ lập tức gây ra lỗi
- (D) Trên C và C++, kích thước của một dãy được lưu trữ để kiểm tra động
- (E) Trên Java, kích thước của một dãy KHÔNG được lưu trữ để kiểm tra động

Phần trình bày sau dùng trong các câu hỏi 33–34:

Cho mã giả của phát biểu for var = expr1 to expr2 do body như sau:

Môn: Nguyên Lý Ngôn Ngữ Lập Trình



- 33. Nếu trong thân *body* có thực thi lệnh *continue* thì điều khiển sẽ chuyển đến thực thi phát biểu có nhãn là:
- (A) out (B) start
- (C) label2
- (D) loop
- (E) label1

(C) n = n + 1;

34. Với hiện thực của phát biểu *for* như trên, phát biểu nào đặt trong thân phát biểu **for** dưới đây sẽ làm cho phát biểu này bị lặp mãi mãi?

- (D) câu A và C đúng
- (E) Không phát biểu nào trong các câu A, B và C làm phát biểu for lặp mãi mãi

Phần hướng dẫn này áp dụng cho các câu 35–39

Hãy viết thêm các đoạn mã cần thiết vào phương thức dưới đây để thực hiện sinh mã cho phát biểu *repeat*. Nhắc lại phát biểu *repeat* kết thúc quá trình lặp khi điều kiện của phát biểu trở thành **true**. AST của phát biểu **repeat** được định nghĩa như sau: case class Repeat(val stmts:List[Stmt],val exp:Expr) extends Stmt

```
override def visitRepeat(ast: Repeat, o:Context)= {
        val ctxt = o.asInstanceOf[SubBody]
        ctxt.frame.enterLoop();
        val labelStart = ctxt.frame.getNewLabel()
        val labelBreak = ctxt.frame.getBreakLabel()
        val labelCont = ctxt.frame.getContinueLabel()
        val str1 = //1
        // sinh mã cho từng phát biểu trong thân của repeat
        val str2 = //2
        val str3 = //3
        // sinh mã cho biểu thức điều kiên
        val str4 = visit(ast.exp,...)
        val str5 = //4
        val str6 = //5
        frame().exitLoop();
        str1 + str2 + str3 + str4 + str5 + str6
   }
```

- 35. Đoạn mã nào trong số các đoạn mã dưới đây cần phải xuất hiện ở vị trí //1 trong đoạn mã trên
- (A) ""

- (B) ctxt.emit.emitLABEL(labelStart)
- (C) ctxt.emit.emitGOTO(labelStart)
- (D) ctxt.emit.emitLABEL(labelBreak)
- (E) ctxt.emit.emitLABEL(labelCont)



- 36. Đoạn mã nào trong số các đoạn mã dưới đây cần phải xuất hiện ở vị trí //2 trong đoạn mã trên
- (A) visit(ast.stmts,o)
- (B) ast.stmts.map(x = > visit(x, o))
- (C) ast.stmts.foldLeft("")((x,y)=>y + visit(x,o))
- (D) ast.stmts.filter(x = > visit(x, o))
- (E) ast.stmts.foldLeft("")((x,y)=>x + visit(y,o))
  - 37. Đoạn mã nào trong số các đoạn mã dưới đây cần phải xuất hiện ở vị trí //3 trong đoạn mã trên
- (A) ""

- (B) ctxt.emit.emitIFFALSE(labelStart)
- (C) ctxt.emit.emitGOTO(labelStart)
- (D) ctxt.emit.emitLABEL(labelBreak)
- (E) ctxt.emit.emitLABEL(labelCont)
  - 38. Đoạn mã nào trong số các đoạn mã dưới đây cần phải xuất hiện ở vị trí //4 trong đoạn mã trên
- (A) ctxt.emit.emitGOTO(labelBreak)
- (B) ctxt.emit.emitIFFALSE(labelStart)
- (C) ctxt.emit.emitGOTO(labelStart)
- (D) ctxt.emit.emitLABEL(labelBreak)
- (E) ctxt.emit.emitLABEL(labelCont)
- 39. Đoạn mã nào trong số các đoạn mã dưới đây cần phải xuất hiện ở vị trí //5 trong đoạn mã trên
- (A) ctxt.emit.emitGOTO(labelBreak))
- (B) ctxt.emit.emitIFFALSE(labelStart))
- (C) ctxt.emit.emitGOTO(labelStart))
- (D) ctxt.emit.emitLABEL(labelBreak))
- (E) ctxt.emit.emitLABEL(labelCont))
  - 40. Cho khai báo sau được viết trên ngôn ngữ lập trình Ada:

Cho kích thước của các kiểu *Integer*, *Float*, *Boolean và Enumeration* lần lượt là 2, 4, 1, và 2, và kích thước cho phần mô tả kiểu (type description) là 0. Cho biết kích thước của một đối tượng kiểu **Figure**?

- (A) 21
- (B) 15
- (C) 13
- (D) 23
- (E) Khác
- 41. Cho biết kích thước của đối tượng dữ liệu  ${\bf x}$  được khai báo như sau: x: set of 1..16
- (A) 4 bytes
- (B) 2 bytes
- (C) 16 bytes
- (D) 4 bits
- (E) 1 byte



42. Cho  $X = \{a,b\}$ . Chọn biểu thức chính quy (regular expression) mô tả ngôn ngữ chứa bất kỳ chuỗi nào được tạo bởi các ký tư trên tập X nhưng KHÔNG chứa chuỗi có 2 ký tư a liên tiếp.

```
(A) a|(abb^*)^*
```

(B) 
$$b*a?(bb*a)*b*$$
 (C)  $(b*ab*)*$ 

(D) 
$$b^*(abb^*)^*$$

43. Cho biết mã Jasmin của biểu thức viết bằng ngôn ngữ C như sau:

với a và b là 2 biến nguyên có chỉ số lần lượt là 0 và 1.

$$\begin{array}{c} iload\_0 \\ iload\_1 \end{array}$$

imul (A) iconst 2 isub

iconst 2 imul isub

iload 1 isub iconst 2

iload 1 (D) isub iconst 2

iconst 2 (E) isub imul

. . .

Đoạn mã sau được dùng trong các câu 44–45:

```
int *p = new int;
void foo(int * r) {
    delete r;
}
foo(p); //1
*p = 2; //2
```

44. Hiện tượng gì xảy ra khi p được truyền cho r ở phát biểu //1 trong đoạn mã trên:

- (A) Con trỏ chưa khai báo (undeclared pointer) (B) Bí danh (alias)

(C) Rác (garbage)

- (D) Khai báo trùng tên (redeclared)
- (E) Tham chiếu treo (dangling reference)

45. Hiện tượng gì xảy ra khi thực thi phát biểu //2 trong đoạn mã trên:

- (A) Tham chiếu treo
- (B) Bí danh

(C) Con trỏ chưa khai báo

- (D) Da hình (polymorphism)
- (E) Rác

46. Giả sử chương trình có một lỗi văn phạm nằm sau lệnh **print**. Khi em nhấn nút Run (để dịch và chạy chương trình) trên trình soạn thảo, kết quả được in ra bởi lệnh **print** trên trước khi lỗi văn phạm được báo. Hỏi chế độ dịch của trình soạn thảo là gì?

(A) trình liên kết (link editor)

(B) trình biên dich (compiler)

aload 1

iload 3

aload 1

- (C) trình biên dich đông(just-in-time compiler)
- (D) trình thông dich(interpreter)

- (E) trình hợp ngữ (assembler)
  - 47. Cho biết mã Jasmin của phát biểu gán sau:

$$a[i] = a[i] + 2$$

iastore

với i kiểu nguyên có chỉ số là 3 và a là kiểu dãy nguyên có chỉ số là 1.

aload 1 iadd iload 3 iaload iaload aload 1 (A) (B) iconst 2 iload 3 iadd dup iastore iconst 2

iastore aload 1 (C) iload 3 iconst 2 iadd

aload 1

iload 3

iload 3 (D) iaload iconst 2 iadd iastore

dup (E) iaload iconst 2 iadd

iastore

aload 1

iload 3

dup

. . .



## II. Phần câu hỏi tự luận: (dành cho tất cả sinh viên) (2 điểm)

48.	Trên những ngôn ngữ lập trình cho phép rút ngắn tính toán (short-circuit evaluation) biểu
	thức luận lý, biểu thức sau sẽ không gây ra lỗi khi thực thi. Biết hàm $length(a)$ trả về kích
	thước dãy a.

```
 ((x < length(a)) \&\& (a[x] >= 1))  Hãy dùng if then else để mô phỏng quá trình tính toán của biểu thức luận lý trên?
```

49. Hãy nêu những giải pháp đã được sử dụng để tránh tham chiếu treo (dangling reference) trong trường hợp trả về hàm như sau:

```
void->int F() {
    int x = 1;
    int g() {
        return x + 1;
    }
    return g;
}
void->int gg = F();
int z = gg();
```

50. Hãy viết lại biểu thức ở dạng trung tố (infix) sau sang dạng biểu thức tiền tố (prefix) Cambridge Polish:

```
a - b * c * d - e + f
```

Biết rằng độ ưu tiên và tính kết hợp của các phép toán trong biểu thức như thông thường (đều kết hợp trái và \* có ưu tiên cao hơn +, -). Trong biểu thức tiền tố nhiều toán hạng, thứ tự tính toán cũng từ trái sang phải.

Yêu cầu: Biểu thức dạng tiền tố Cambridge Polish phải thỏa các yêu cầu sau:



- Thứ tự xuất hiện các toán hạng trong biểu thức dạng tiềm tố phải có cùng thứ tự xuất hiện như trong biểu thức trung tố
- $\bullet$  Số ( và ) là ít nhất
- Có cùng thứ tự tính toán các phép toán với thứ tự đó trong biểu thức dạng trung tố


## III. Phần bài tập lớn: (dành cho tất cả sinh viên)

- 51. Viết hàm visitBlock(ast:Block,o:Context) để thực hiện kiểm tra kiểu. Hàm visitBlock sẽ thực hiện chức năng xây dựng bảng danh hiệu (symbol table) và truyền bảng này khi visit các phát biểu trong khối. Yêu cầu giải thích rõ về Context cụ thể được sử dụng trong hàm và ý tưởng mã của hàm. Nhắc lại AST của Block được khai báo như sau: case class Block(val decl:List[Decl],val stmts:List[Stmt]) extends Stmt
- 52. Viết hàm visitIf(ast:If,o:Context) để thực hiện sinh mã cho một phát biểu If. Nhắc lại AST của If được khai báo như sau: case class If(val expr:Expr, val thenStmt:Stmt, val elseStmt:Option[Stmt]) extends Stmt Yêu cầu giải thích rõ về Context cụ thể được sử dụng trong hàm và ý tưởng mã của hàm.

### IV. Phần KSTN: (dành cho sinh viên KSTN)

- 53. Viết hàm visitIf(ast:If,o:Context) để thực hiện kiểm tra lỗi FunctionNotReTurn? Yêu cầu giải thích ý tưởng thực hiện thể hiện qua mã của hàm?
- 54. Viết hàm visitBlock(ast:Block,o:Context) để thực hiện sinh mã cho một khối (block)? Yêu cầu giải thích ý tưởng thực hiện sinh mã cho các khai báo array trong một khối?

HÉT-	
Họ và tên:	Điểm:
MSSV:	





Chủ nhiệm bộ môn	Giảng viên ra đề
Chữ kí:	Chữ kí:
Họ tên:	Họ tên:



# KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH $\mbox{BỘ MÔN KHOA HỌC MÁY TÍNH}$

ĐÁP ÁN cho Đề thi cuối kỳ								
Môn thi: Nguyên Lý Ng Thời gian: 120 phút	gôn Ngữ Lập T	rình						
□ Sinh viên được phép sử ơ Mã đề: 1153	lụng tài liệu	⊠ Sin	oxtimes Sinh viên không được sử dụng tài liệu					
I. Phần câu hỏi trắc ng	chiệm:(8 điểm)	)						
<ol> <li>Liệt kê theo thứ tự ch chuỗi) của ngôn ngữ đo ab*(a b)? a</li> </ol>			_	•				
(A) aa, aba, aaa, abaa, aaba D aa, aba, aaa, abaa, abba			(C) aa, a	ba, aaa, abaa, aaaa				
<ol> <li>Gọi append là hàm dùn sẽ có kết quả là [1,2,3,4 bậc cao (high-order fun</li> </ol>	[4,5,6]. Hãy hiện t							
<ul> <li>(A) a.map(b)((x,y)=&gt;x::y)</li> <li>(C) b.foldRight(a)((x,y)=&gt;x</li> <li>(E) b.foldLeft(a)((x,y)=&gt;x:</li> </ul>			a.foldLeft(b)((x,ya.foldRight(b)((x					
3. Đoạn code viết bằng S niệm Hàm Currying	scala nào dưới đâ	y có thể dù	ng như một ví d	ụ minh họa cho khái				
(A) def add1(n:Int) = n + 1 (B) List(1,2,3).foldLeft(0)(_ (C) List(1,2,3).filter(_ > 1) (D) def add(n:Int) = (x:Int) E def add(n:Int)(x:Int) =	$+$ _) => n + x;val ac		)					
<ol> <li>Cho giá trị ban đầu củ thực hiện phép gán sau a = c * (c = 5);</li> </ol>		_	giá trị <b>có thể c</b>	<b>có</b> của biến <b>a</b> sau khi				
(A) 15, 25, 9 (B) 15	$oldsymbol{\mathbf{C}}$	15, 25	(D) 25	(E) 9				
Cho văn phạm của một biểu $\exp \rightarrow \operatorname{term} '=' \exp \mid \cdot \mid$ $\operatorname{term} \rightarrow \operatorname{term} '+' \operatorname{fact} \mid$ $\operatorname{fact} \rightarrow \operatorname{ope} '*' \operatorname{ope} \mid \operatorname{ope} $ $\operatorname{ope} \rightarrow '(' \exp ')' \mid \operatorname{ID}$	term term '>' fact   fa		<b>−7</b> :					

với ID là một danh hiệu.

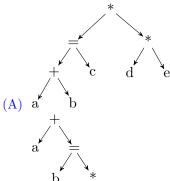


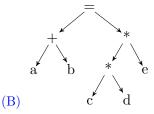
- 5. Chọn biểu thức được viết đúng văn phạm, có số lượng dấu '(' và ')' ít nhất mà tương đương với biểu thức sau: (a + (b = ((c \* d) \* e))) > f
- (A) (a + (b = (c \* d) \* e)) > f
- (B) (a + (b = ((c \* d) \* e))) > f

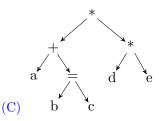
(C) a + (b = c \* d \* e) > f

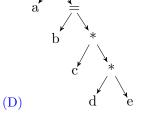
(D) a + b = c \* d \* e > f

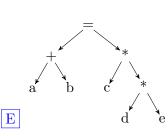
- E a + (b = (c \* d) \* e) > f
  - 6. Vẽ AST cho biểu thức sau được viết dựa vào văn phạm trên:  $\mathbf{a} + \mathbf{b} = \mathbf{c} * (\mathbf{d} * \mathbf{e})$











7. Cho cấu trúc AST được khai báo trên Scala như sau: trait Exp case class Bin(op:String,e1:Exp,e2:Exp) extends Exp case class Id(i:String) extends Exp

Chọn AST cho biểu thức sau:  $\mathbf{a} + \mathbf{b} > \mathbf{c} = \mathbf{d} * \mathbf{e} = \mathbf{f}$ 

- $(A) \ \operatorname{Bin}("=",\operatorname{Bin}(">",\operatorname{Bin}("+",\operatorname{Id}("a"),\operatorname{Id}("b")),\operatorname{Id}("c")),\operatorname{Bin}("*",\operatorname{Id}("d"),\operatorname{Bin}("=",\operatorname{Id}("e"),\operatorname{Id}("f"))))$
- $(B) \ \operatorname{Bin}("=",\operatorname{Bin}("=",\operatorname{Bin}(">",\operatorname{Bin}("+",\operatorname{Id}("a"),\operatorname{Id}("b")),\operatorname{Id}("c")),\operatorname{Bin}("*",\operatorname{Id}("d"),\operatorname{Id}("e"))),\operatorname{Id}("f"))$
- $\underbrace{(\texttt{C})} \; \operatorname{Bin}("=",\operatorname{Bin}("=",\operatorname{Bin}("+",\operatorname{Id}("a"),\operatorname{Bin}(">",\operatorname{Id}("b"),\operatorname{Id}("c"))),\operatorname{Bin}("*",\operatorname{Id}("d"),\operatorname{Id}("e"))),\operatorname{Id}("f"))$
- D Bin("=",Bin(">",Bin("+",Id("a"),Id("b")),Id("c")),Bin("=",Bin("\*",Id("d"),Id("e"))),Id("f"))
- $\overline{\text{(E)}} \ \operatorname{Bin}("=",\operatorname{Bin}("+",\operatorname{Id}("a"),\operatorname{Bin}(">",\operatorname{Id}("b"),\operatorname{Id}("c"))),\operatorname{Bin}("=",\operatorname{Bin}("*",\operatorname{Id}("d"),\operatorname{Id}("e"))),\operatorname{Id}("f"))$ 
  - 8. Cho lớp(class) A là lớp cha (superclass) của lớp B và B là lớp cha của hai lớp C và D. Trong lớp A, có khai báo phương thức thực thể (instance method) **foo** và phương thức này bị ghi đè(override) trên tất cả các lớp B, C và D. Cho biến x được khai báo như sau: **B** x;

Qui ước viết  $func_T$  là phương thức func được khai báo trong lớp T, cho biết tập của phương thức đích của cuộc gọi sau:  $\mathbf{x.foo}()$ ;

- (A)  $\{foo_C, foo_D\}$
- (B)  $\{foo_A, foo_B, foo_C, foo_D\}$   $\boxed{\mathbf{C}}$   $\{foo_B, foo_C, foo_D\}$

(D)  $\{foo_B\}$ 

(E)  $\{foo_A, foo_B\}$ 



- 9. Trình định thời (scheduled subprograms) thường được dùng trong
- (A) Lập trình song song (Parallel Programming)
- (B) Lập trình hướng đối tượng (Object-Oriented Programming)
- C Lập trình thời gian thực (Real-time Programming)
- (D) Lâp trình hàm (Functional Programming)
- (E) Lâp trình hướng sư kiên (Event-driven Programming)

Đoạn code sau, được viết trên ngôn ngữ dùng qui tắc tầm vực tĩnh (static-scope rule), áp dụng cho các câu 10-13

```
procedure main(){
    var a, b, c:integer; //1
    procedure sub1(a: integer) { //2
        procedure sub3();
        procedure sub2() {
            var a, c : real;//3
            sub3();
        }
        procedure sub3() {
            a // use a
        }
        sub2();
    }
    sub1(3);
}
```

10. Môi trường tham khảo tĩnh (static referencing environment) của thủ tục sub3 KHÔNG chứa

- A a ở //1
- (B) sub2
- (C) a ở //2
- (D) sub1
- (E) sub3
- 11. Giả sử chuỗi gọi là main  $\rightarrow$  sub1  $\rightarrow$  sub2  $\rightarrow$  sub3, tham khảo đến **a** trong **sub3** ứng với khai báo
- (A) Báo lỗi a chưa khai báo
- B a trong sub1 (//2) (C) a trong main (//1)

- (D) a trong sub2 (//3)
- (E) a trong sub3
- 12. Giả sử chuỗi gọi là main  $\rightarrow$  sub1  $\rightarrow$  sub2  $\rightarrow$  sub3, tham khảo đến  $\bf a$  trong sub3  $\bf khi$  thực thi ứng với kết hợp của a trong bản hoạt đông của
- (A) main
- B sub1
- (C) Báo lỗi không tìm thấy a
- (D) sub2

- (E) sub3
- 13. Nếu đoạn code trên được viết trên ngôn ngữ dùng qui tắc tầm vực động (dynamic-scope rule), và giả sử chuỗi gọi là main  $\rightarrow$  sub1  $\rightarrow$  sub2  $\rightarrow$  sub3, tham khảo đến a trong sub3 khi thực thi ứng với kết hợp của a trong bản hoạt động của
- (A) Báo lỗi không tìm thấy a
- (B) main
- (C) sub3
- (D) sub1

E sub2



14. Kiểu union của ngôn ngữ C++ cho phép ghi vào ở thành phần này nhưng đọc lên ở thành phần khác như trong ví dụ sau:

```
union {
    int songuyen;
    char kytu[2];
x.songuyen = 12;
cout << x.kytu[0] << " " << x.kytu[1] << endl;</pre>
Đặc tính này của ngôn ngữ C++ gây ảnh hưởng XẤU NHẤT đối với
```

- (A) tính trực giao (orthonogality)
- (B) tính đơn giản (simplicity)

(C) chi phí (cost)

(D) tính dễ viết (writability)

- E tính tin cây (reliability)
- 15. Khi cần khai thác đặc tính của kiểu union(các thành phần loại trừ lẫn nhau), làm thế nào để hiện thực trên những ngôn ngữ lập trình hướng đối tương như Java(không có kiểu union)? Cu thể làm thế nào để hiện thực biến z kiểu union sau trên những ngôn ngữ này?

```
union {
    string loikhen;
    string loiche;
} z;
```

- Định nghĩa lớp cha A và hai lớp con B (có thuộc tính loikhen) và C (có thuộc tính loiche), (A) và z có kiểu A
- Định nghĩa một lớp A có 2 thuộc tính loikhen và loiche; và z có kiểu A (B)
- Định nghĩa lớp trừu tượng (abstract class) A và 2 lớp con B (có thuộc tính loikhen) và C C (có thuộc tính loiche), và z có kiểu A
- Đinh nghĩa lớp A có thuộc tính loikhen và lớp B(con của A) có thuộc tính loiche, và z có (D)
- Định nghĩa lớp A có thuộc tính loikhen và lớp B(con của A) có thuộc tính loiche, và z có (E) kiểu B
- 16. Để thực hiện tìm kiếm một chuỗi (String) trên nhiều loại danh sách khác nhau, hãy chọn giải pháp thích hợp nhất để điền vào chỗ trống trong định nghĩa hàm sau:

```
def lookup[T](x:String,lst:List[T],f:T=>String):Option[T] =
          lst match {
             case List() => None
             case h::t =>_____
          }
(A) if (h == x) Some(h) else lookup(x,t,f)
B \mid \text{if } (f(h) == x) \text{ Some}(h) \text{ else lookup}(x,t,f)
```

- (C) if (h == f(x)) Some(f(h)) else lookup(x,t,f)
- (D) if (h == f(x) Some(h) else lookup(x,t,f)
- (E) if (f(h) == x) Some(f(h)) else lookup(x,t,f)



- 17. Cơ chế gọi chương trình con nào mà điều khiển có thể chuyển vào một vị trí ở giữa chương trình con được gọi thay vì chuyển vào đầu chương trình con được gọi?
- (A) Trình định thời (Scheduled Subprogram)
- (B) Gọi trở về đơn giản (Simple Call Return)
- C Trình công hành (Coroutine)
- (D) Goi đê qui (Recursive call)
- (E) Biến cố Xử lý biến cố (Exception)

Đoạn mã sau được dùng cho các câu 18–23.

Cho một đoan chương trình được viết trên một ngôn ngữ tưa C như sau:

```
int A[5] = \{1,3,5,7,9\}; // index of A starts from 0
int j = 0;
int n = 5;
int sumAndIncrease(int a, int i) {
     int s = 0;
     for ( ; i < n; i = i + 1)  {
        s = s + a;
        A[j] = A[j] + 1;
     return s;
}
void main(){
     int s = sumAndIncrease(A[j], j);
     printf("a = \%i \setminus n", s); //1
     printf(" %i %i %i %i %i %i \n",A[0],A[1],A[2],A[3],A[4]);//2
}
```

18. Nếu a và i được truyền bằng **trị-kết quả (passed by value-result)**, giá trị của biến sđược in ra ở phát biểu //1 là bao nhiệu?

- A 5
- (B) 25
- (C) 10
- (D) 15
- (E) Khác
- 19. Nếu a và i được truyền bằng **tri-kết quả (passed by value-result)**, giá trị của biến A được in ra ở phát biểu //2 là bao nhiêu?
- (A) 1 3 5 7 9
- (B) 5 3 5 7 9
- (C) 2 4 6 8 10
- D 63579
- 20. Nếu a và i được truyền bằng **tham khảo (passed by reference)**, giá trị của biến s được in ra ở phát biểu //1 là bao nhiêu?
- (A) 5
- (B) 25
- (C) 10
- (D) 15
- E Khác
- 21. Nếu a và i được truyền bằng **tham khảo (passed by reference)**, giá trị của biến Ađược in ra ở phát biểu //2 là bao nhiêu?
- (A) 1 3 5 7 9
- (B) 5 3 5 7 9
- C 2 4 6 8 10 (D) 6 3 5 7 9
- (E) Khác
- 22. Nếu a và i được truyền bằng **tên (passed by name)**, giá trị của biến s được in ra ở phát biểu //1 là bao nhiêu?
- (A) 5
- B 25
- (C) 15
- (D) 20
- (E) Khác



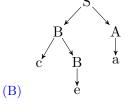
- 23. Nếu a và i được truyền bằng **tên (passed by name)**, giá trị của biến A được in ra ở phát biểu //2 là bao nhiêu?
- (A) 1 3 5 7 9
- (B) 5 3 5 7 9
- C 2 4 6 8 10
- (D) 6 3 5 7 9
- (E) Khác

Cho văn phạm sau dùng cho các câu 24–25:

- 24. Chuỗi nào KHÔNG thuộc ngôn ngữ được mô tả bởi văn phạm trên?
- (A) e
- (B) b
- C cccaaa
- (D) eaaa
- (E) cccddddeaa
- 25. Chọn cây phân tích cú pháp (parse tree) cho chuỗi nhập: cea

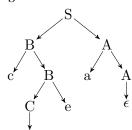


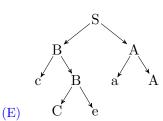












- D
- 26. Viết lại biểu thức trung tố (infix) sau sang dạng hậu tố (postfix) Polish? Giả sử là tất cả các phép toán đều có 2 toán hạng và có độ ưu tiên và tính kết hợp như trong ngôn ngữ lập trình C.

$$a + b * c - d * e + f$$

- (A) a + b c \* d e \* f +
- (B) a b + c \* d e \* f +
- C a b c \* + d e \* f +

- (D) a b c \* + d e f \* +
- (E) a b c \* + d e \* + f -

Đoạn code sau dùng cho các câu 27-29

```
int p;
int* foo(int x) {
      static int q;
      int *s = new int;
      switch (x) {
           case 1: return &p;
           case 2: return &q;
           case 3: return &x;
           case 4: return s;
           default: return foo(x-1);
       }
}
```

(C) return &q

D return &x



(A) return &p

(E) Khô	ng phát biể	ểu nào gây ra lỗi tư	ên khi thực	thi			
28. Tro	ong đoạn m	ã trên, đối tượng c	lữ liệu nào	có thể trở t	hành rác (ga	rbage)	
(A) p (E) Khô	ng có đối t	B trỏ đến ượng nào có thể tr		(C) q		(D) x	
	i hàm <b>foo</b> d tượng dữ l	được gọi đệ quy (re liệu nào?	cursive) thì	các bản ho	ạt động của <b>f</b>	oo dùng chung nh	ıững
(A) q và	S	(B) p, q, x và s	(C) x, q	và s	(D) p	E p và	q
Ví var x =	dụ: x = "def" = 1;	ngôn ngữ Javascr ; ữa một biến và kiểu					
(C) Khở	n thực (imp i động (loa n (compiling	/			(running) rình (progran	nming)	
	<u> </u>	ương trình nào là t-driven Programm	<u> </u>	hương trìn	h con cơ bản	của Lập trình hư	íớng
(B) Gọi (C) Trìn (D) Gọi	trở về đơn h định thờ đệ qui (Re	r biến cố (Exceptio giản (Simple Call i (Scheduled Subpr cursive call) th (Coroutine)	- Return)				
32. Ch	ọn phát biể	ểu ĐÚNG trong cá	c phát biểu	về dãy (ar	ray) sau:		
(B) Truy (C) Truy (D) Trên	y xuất một y xuất một n C và C+-	nước của dãy có the phần tử bên ngoà phần tử bên ngoà +, kích thước của r thước của một dã	i một dãy C i một dãy J nột dãy đượ	không đượ ava không ợc lưu trữ đ	ợc phép, sẽ lậ được phép, sẽ lể kiểm tra đơ	p tức gây ra lỗi g lập tức gây ra lỗ ộng	i
	_	dùng trong các câ $f$ thiểu $f$ or			như sau:		
start:	<pre>var = ex] etemp = e if (var : body</pre>	2"	ut				
label1: label2: out:	•	p					

27. Phát biểu nào sẽ gây ra lỗi tham chiếu treo (dangling reference) khi thực thi

(B) return s



33. Nếu trong thân *body* có thực thi lệnh *continue* thì điều khiển sẽ chuyển đến thực thi phát biểu có nhãn là:

- (A) out
- (B) start
- (C) label2
- (D) loop
- E label1

34. Với hiện thực của phát biểu *for* như trên, phát biểu nào đặt trong thân phát biểu **for** dưới đây sẽ làm cho phát biểu này bị lặp mãi mãi?

```
n = 10;s = 0;
for i = s to n do ______
```

A i = i - 1;

(B) s = s - 1;

(C) n = n + 1;

- (D) câu A và C đúng
- (E) Không phát biểu nào trong các câu A, B và C làm phát biểu for lặp mãi mãi

Phần hướng dẫn này áp dụng cho các câu 35–39

Hãy viết thêm các đoạn mã cần thiết vào phương thức dưới đây để thực hiện sinh mã cho phát biểu *repeat*. Nhắc lại phát biểu *repeat* kết thúc quá trình lặp khi điều kiện của phát biểu trở thành **true**. AST của phát biểu **repeat** được định nghĩa như sau: case class Repeat(val stmts:List[Stmt],val exp:Expr) extends Stmt

```
override def visitRepeat(ast: Repeat, o:Context)= {
        val ctxt = o.asInstanceOf[SubBody]
        ctxt.frame.enterLoop();
        val labelStart = ctxt.frame.getNewLabel()
        val labelBreak = ctxt.frame.getBreakLabel()
        val labelCont = ctxt.frame.getContinueLabel()
        val str1 = //1
        // sinh mã cho từng phát biểu trong thân của repeat
        val str2 = //2
        val str3 = //3
        // sinh mã cho biểu thức điều kiên
        val str4 = visit(ast.exp,...)
        val str5 = //4
        val str6 = //5
        frame().exitLoop();
        str1 + str2 + str3 + str4 + str5 + str6
   }
```

- 35. Đoạn mã nào trong số các đoạn mã dưới đây cần phải xuất hiện ở vị trí //1 trong đoạn mã trên
- (A) ""

- B ctxt.emit.emitLABEL(labelStart)
- (C) ctxt.emit.emitGOTO(labelStart)
- (D) ctxt.emit.emitLABEL(labelBreak)
- (E) ctxt.emit.emitLABEL(labelCont)
  - 36. Đoạn mã nào trong số các đoạn mã dưới đây cần phải xuất hiện ở vị trí //2 trong đoạn mã trên
- (A) visit(ast.stmts,o)
- (B) ast.stmts.map(x = > visit(x, o))
- (C) ast.stmts.foldLeft("")((x,y)=>y + visit(x,o))
- (D) ast.stmts.filter(x = > visit(x, o))
- E ast.stmts.foldLeft("")((x,y)=>x + visit(y,o))



37. Đoạn mã nào trong số các đoạn mã dưới đây cần phải xuất hiện ở vị trí //3 trong đoạn mã trên

(A) ""

(B) ctxt.emit.emitIFFALSE(labelStart)

(C) ctxt.emit.emitGOTO(labelStart)

(D) ctxt.emit.emitLABEL(labelBreak)

E ctxt.emit.emitLABEL(labelCont)

38. Đoạn mã nào trong số các đoạn mã dưới đây cần phải xuất hiện ở vị trí //4 trong đoạn mã trên

(A) ctxt.emit.emitGOTO(labelBreak)

B ctxt.emit.emitIFFALSE(labelStart)

(C) ctxt.emit.emitGOTO(labelStart)

(D) ctxt.emit.emitLABEL(labelBreak)

(E) ctxt.emit.emitLABEL(labelCont)

39. Đoạn mã nào trong số các đoạn mã dưới đây cần phải xuất hiện ở vị trí //5 trong đoạn mã trên

(A) ctxt.emit.emitGOTO(labelBreak))

(B) ctxt.emit.emitIFFALSE(labelStart))

(C) ctxt.emit.emitGOTO(labelStart))

D ctxt.emit.emitLABEL(labelBreak))

(E) ctxt.emit.emitLABEL(labelCont))

40. Cho khai báo sau được viết trên ngôn ngữ lập trình Ada:

```
type Shape is (Circle, Triangle, Rectangle);
type Colors is (Red, Green, Blue);
type Figure (Form: Shape) is record
        Filled: Boolean;
        Color: Colors;
        case Form is
                when Circle => Diameter: Float;
                when Triangle =>
                        Leftside, Rightside: Integer;
                        Angle: Float;
                when Rectangle => Side1, Side2: Integer;
        end case;
```

end record;

Cho kích thước của các kiểu Integer, Float, Boolean và Enumeration lần lượt là 2, 4, 1, và 2, và kích thước cho phần mô tả kiểu (type description) là 0. Cho biết kích thước của môt đối tương kiểu **Figure**?

(A) 21

(B) 15

C 13

(D) 23

(E) Khác

41. Cho biết kích thước của đối tượng dữ liệu **x** được khai báo như sau: x: set of 1..16

(A) 4 bytes

B 2 bytes

(C) 16 bytes

(D) 4 bits

(E) 1 byte

42. Cho X = {a,b}. Chọn biểu thức chính quy (regular expression) mô tả ngôn ngữ chứa bất kỳ chuỗi nào được tạo bởi các ký tự trên tập X nhưng KHÔNG chứa chuỗi có 2 ký tự a liên tiếp.

(A)  $a|(abb^*)^*$ 

B b\*a?(bb\*a)\*b\*

(C) (b\*ab\*)\*

(D)  $b^*(abb^*)^*$ 

(E) b\*ab\*ab\*



43. Cho biết mã Jasmin của biểu thức viết bằng ngôn ngữ C như sau:

a \* b - 2

với a và b là 2 biến nguyên có chỉ số lần lượt là 0 và 1.

iload\_0 iload\_1

iconst 2

 $iload_0$   $iload_1$  iconst 2

imul

 $\begin{array}{c} load\_0 \\ iload\_1 \\ isub \end{array}$ 

imul

iconst 2

 $\begin{array}{c} \text{imul} \\ \text{(D)} & \begin{array}{c} \text{iload} \\ \text{isub} \\ \text{iconst} \end{array} 2 \end{array}$ 

iload 0

(E)  $\begin{array}{c} \operatorname{iload}_{-1} \\ \operatorname{iconst}_{-2} \\ \operatorname{isub} \end{array}$ 

iload 0

isub ...

imul

A

isub ...

...

imul ...

Đoạn mã sau được dùng trong các câu 44–45:

(B)

```
int *p = new int;
void foo(int * r) {
    delete r;
}
foo(p); //1
*p = 2; //2
```

44. Hiện tượng gì xảy ra khi p được truyền cho r ở phát biểu //1 trong đoạn mã trên:

(A) Con trỏ chưa khai báo (undeclared pointer)

B Bí danh (alias)

(C) Rác (garbage)

(D) Khai báo trùng tên (redeclared)

(E) Tham chiếu treo (dangling reference)

45. Hiện tượng gì xảy ra khi thực thi phát biểu //2 trong đoạn mã trên:

A Tham chiếu treo

(B) Bí danh

(C) Con trỏ chưa khai báo

(D) Da hình (polymorphism)

(E) Rác

46. Giả sử chương trình có một lỗi văn phạm nằm sau lệnh **print**. Khi em nhấn nút Run (để dịch và chạy chương trình) trên trình soạn thảo, kết quả được in ra bởi lệnh **print** trên trước khi lỗi văn phạm được báo. Hỏi chế độ dịch của trình soạn thảo là gì?

(A) trình liên kết (link editor)

(B) trình biên dich (compiler)

aload 1

iload 3

(C) trình biên dịch động(just-in-time compiler)

D trình thông dich(interpreter)

(E) trình hợp ngữ (assembler)

47. Cho biết mã Jasmin của phát biểu gán sau:

a[i] = a[i] + 2

với i kiểu nguyên có chỉ số là 3 và a là kiểu dãy nguyên có chỉ số là 1.

 $\begin{array}{c} \text{iadd} \\ \text{iaload} \\ \text{(A)} \quad \begin{array}{c} \text{aload}\_1 \\ \text{iload}\_3 \\ \text{dup} \\ \text{iconst} \quad 2 \end{array}$ 

iastore

iload\_3
iaload
(B) iconst\_2
iadd
iastore

aload 1

 $\begin{array}{c} \text{iload}\_3\\ \text{iastore}\\ \text{(C)} \quad \begin{array}{c} \text{aload}\_1\\ \text{iload}\_3\\ \text{iconst}\_2\\ \text{iadd} \end{array}$ 

aload 1

aload\_1
iload\_3
iaload
iconst\_2
iadd
iastore

iload\_3
dup
(E) iaload
iconst\_2
iadd
iastore

aload 1

dup

...

II. Phần câu hỏi tự luận: (dành cho tất cả sinh viên) (2 điểm)



48. Trên những ngôn ngữ lập trình cho phép rút ngắn tính toán (short-circuit evaluation) biểu thức luận lý, biểu thức sau sẽ không gây ra lỗi khi thực thi. Biết hàm length(a) trả về kích thước dãy a.

```
\begin{array}{c} \text{((x < length(a)) \&\& (a[x] >= 1))} \\ \text{Hãy dùng if then else để mô phỏng quá trình tính toán của biểu thức luận lý trên?} \\ \\ \\ \end{array}
```

```
Lời giải. if (x < length(a)) then if (a[x] > 0) then ... Giải pháp đề xuất đúng: 2
```

49. Hãy nêu những giải pháp đã được sử dụng để tránh tham chiếu treo (dangling reference) trong trường hợp trả về hàm như sau:

```
void->int F() {
    int x = 1;
    int g() {
        return x + 1;
    }
    return g;
}
void->int gg = F();
int z = gg();
```

50. Hãy viết lại biểu thức ở dạng trung tố (infix) sau sang dạng biểu thức tiền tố (prefix) Cambridge Polish:

```
a - b * c * d - e + f
```

Biết rằng độ ưu tiên và tính kết hợp của các phép toán trong biểu thức như thông thường (đều kết hợp trái và \* có ưu tiên cao hơn +, -). Trong biểu thức tiền tố nhiều toán hạng, thứ tự tính toán cũng từ trái sang phải.



Yêu cầu: Biểu thức dạng tiền tố Cambridge Polish phải thỏa các yêu cầu sau:

- Thứ tự xuất hiện các toán hạng trong biểu thức dạng tiềm tố phải có cùng thứ tự xuất hiện như trong biểu thức trung tố
- Số ( và ) là ít nhất
- Có cùng thứ tự tính toán các phép toán với thứ tự đó trong biểu thức dạng trung tố

Lời giải. (+ (- a (\* b c d) e) f)

Nếu viết đúng biểu thức dạng tiền tố Cambridge Polish: 1 Nếu viết đúng và có số () ít nhất: 1

### III. Phần bài tập lớn: (dành cho tất cả sinh viên)

- 51. Viết hàm visitBlock(ast:Block,o:Context) để thực hiện kiểm tra kiểu. Hàm visitBlock sẽ thực hiện chức năng xây dựng bảng danh hiệu (symbol table) và truyền bảng này khi visit các phát biểu trong khối. Yêu cầu giải thích rõ về Context cụ thể được sử dụng trong hàm và ý tưởng mã của hàm. Nhắc lại AST của Block được khai báo như sau: case class Block(val decl:List[Decl],val stmts:List[Stmt]) extends Stmt
- 52. Viết hàm visitIf(ast:If,o:Context) để thực hiện sinh mã cho một phát biểu If. Nhắc lại AST của If được khai báo như sau: case class If(val expr:Expr, val thenStmt:Stmt, val elseStmt:Option[Stmt]) extends Stmt Yêu cầu giải thích rõ về Context cụ thể được sử dụng trong hàm và ý tưởng mã của hàm.

#### IV. Phần KSTN: (dành cho sinh viên KSTN)

- 53. Viết hàm visitIf(ast:If,o:Context) để thực hiện kiểm tra lỗi FunctionNotReTurn? Yêu cầu giải thích ý tưởng thực hiện thể hiện qua mã của hàm?
- 54. Viết hàm visitBlock(ast:Block,o:Context) để thực hiện sinh mã cho một khối (block)? Yêu cầu giải thích ý tưởng thực hiện sinh mã cho các khai báo array trong một khối?

Question Distribution

Content	Level 1	Level 2	Level 3	Level 4
Introduction		1		
Lexical	1	4	5	
Syntax	2	4	7	
AST		2		1
Total	3	11	12	1

-HÊT-





Chủ nhiệm bộ môn	Giảng viên ra đề
Chữ kí:	Chữ kí:
Họ tên:	Họ tên:



КНОА <b>КНОА</b>	HỌC VÀ	KŸ TH	ΙŲẬΤ	MÁY	TÍNH
BÔ MÔ	ÎN <b>KHOA</b>	HOC N	лáу т	ΊΝΗ	

Họ và tên:	
MSSV:	

# Đề thi cuối kỳ

Môn thi: Nguyên Lý Ngôn Ngữ Lập Trình

Thời gian: 120 phút Ngày thi: 23-12-2015

□ Sinh viên được phép sử dụng tài liệu □ Sinh viên không được sử dụng tài liệu

Mã đề: 1154

- Sinh viên phải ghi tên và mã số sinh viên trên đề thi (**trang đầu và trang cuối**), giấy làm bài trắc nghiệm (giấy đỏ) và giấy làm bài. Sinh viên phải tô phần mã đề thi và mã số sinh viên trên giấy làm bài trắc nghiệm. Khi nộp bài, sinh viên phải nộp cả **đề thi, giấy làm bài trắc nghiệm và giấy làm bài**.
- Phần trắc nghiệm sẽ được chấm TỰ ĐỘNG trên giấy làm bài trắc nghiệm. Do đó, phần trắc nghiêm nếu làm trên đề thi sẽ KHÔNG được chấm.
- Đối với các câu hỏi phần trắc nghiệm, sinh viên chỉ chọn MỘT phương án đúng nhất.
- Đối với phần câu hỏi tự luận (phần II), sinh viên làm ngay trên đề thi, ở phần dành riêng ngay dưới mỗi câu hỏi.
- Đối với câu hỏi bài tập lớn (phần III và IV), sinh viên làm trên giấy làm bài.
- Sinh viên lớp đại trà làm 3 phần (I, II và III). Sinh viên lớp tài năng làm cả 4 phần (I, II, III và IV). Chỉ có phần I và II được dùng để tính điểm cuối kỳ, các phần III và IV để tính điểm bài tập lớn 2 và 3.

### I. Phần câu hỏi trắc nghiệm:(8 điểm))

1. Cho biết mã Jasmin của biểu thức viết bằng ngôn ngữ C như sau:

```
a * b - 2
```

(A)

với a và b là 2 biến nguyên có chỉ số lần lượt là 0 và 1.

iload 0 iload 0 iload 0 imul iload 0 iload 1 iload 1 imul load 0 iload 1 imul iconst 2 iload 1 iload 1 iconst 2 (C) (D) (E) isub isub iconst 2 imul isub isub isub iconst 2 iconst 2 imul

Đoạn mã sau được dùng trong các câu 2–3:

```
int *p = new int;
void foo(int * r) {
    delete r;
}
foo(p); //1
*p = 2; //2
```



- 2. Hiện tượng gì xảy ra khi p được truyền cho r ở phát biểu //1 trong đoạn mã trên:
- (A) Con trỏ chưa khai báo (undeclared pointer) (B) Bí danh (alias)
- (C) Khai báo trùng tên (redeclared) (D) Rác (garbage)
- (E) Tham chiếu treo (dangling reference)
  - 3. Hiện tượng gì xảy ra khi thực thi phát biểu //2 trong đoạn mã trên:
- (A) Tham chiếu treo (B) Bí danh (C) Đa hình (polymorphism)
- (D) Con trỏ chưa khai báo (E) Rác
  - 4. Cho  $X = \{a,b\}$ . Chọn biểu thức chính quy (regular expression) mô tả ngôn ngữ chứa bất kỳ chuỗi nào được tạo bởi các ký tự trên tập X nhưng  $KH \hat{O}NG$  chứa chuỗi có 2 ký tự a liên tiếp.
- (A)  $a|(abb^*)^*$
- (B)  $b^*(abb^*)^*$
- (C) (b\*ab\*)\*
- (D) b\*a?(bb\*a)\*b\* (E) b\*ab\*ab\*
- 5. Cho biết kích thước của đối tượng dữ liệu  ${\bf x}$  được khai báo như sau: x: set of 1..16
- (A) 4 bytes
- (B) 2 bytes
- (C) 4 bits
- (D) 16 bytes
- (E) 1 byte
- 6. Chọn phát biểu ĐÚNG trong các phát biểu về dãy (array) sau:
- (A) Trên C, kích thước của dãy có thể tăng thêm sau khi dãy đã được tạo ra
- (B) Truy xuất một phần tử bên ngoài một dãy C không được phép, sẽ lập tức gây ra lỗi
- (C) Trên C và C++, kích thước của một dãy được lưu trữ để kiểm tra động
- (D) Truy xuất một phần tử bên ngoài một dãy Java không được phép, sẽ lập tức gây ra lỗi
- (E) Trên Java, kích thước của một dãy KHÔNG được lưu trữ để kiểm tra động
  - 7. Trình định thời (scheduled subprograms) thường được dùng trong
- (A) Lập trình song song (Parallel Programming)
- (B) Lập trình hướng đối tượng (Object-Oriented Programming)
- (C) Lâp trình hàm (Functional Programming)
- (D) Lâp trình thời gian thực (Real-time Programming)
- (E) Lâp trình hướng sư kiên (Event-driven Programming)
  - 8. Liệt kê theo thứ tự chiều dài tăng dần ít nhất 5 chuỗi ngắn nhất (nếu có nhiều hơn 5 chuỗi) của ngôn ngữ được mô tả bởi biểu thức chính quy (regular expression) sau:  $ab^*(a|b)$ ? a
- (A) aa, aba, aaa, abaa, aaba
- (B)  $\epsilon$ , a, aa, aba, abaa
- (C) aa, aba, aaa, abaa, abba

- (D) aa, aba, aaa, abaa, aaaa
- (E) a, aa, aba, aaa, aaba

Doạn code sau, được viết trên ngôn ngữ dùng **qui tắc tầm vực tĩnh (static-scope rule)**, áp dung cho các câu 9–12

```
procedure main(){
   var a, b, c:integer; //1
   procedure sub1(a: integer) { //2
      procedure sub3();
      procedure sub2() {
        var a, c : real;//3
        sub3();
   }
   procedure sub3() {
```



```
a // use a
}
sub2();
}
sub1(3);
}
```

9. Môi trường tham khảo tĩnh (static referencing environment) của thủ tục  ${f sub3}$  KHÔNG chứa

- (A) a ở //1
- (B) sub1
- (C) a ở //2
- (D) sub2
- (E) sub3

10. Giả sử chuỗi gọi là main  $\to$  sub<br/>1  $\to$  sub2  $\to$  sub3, tham khảo đến **a** trong **sub3** ứng với khai báo

- (A) Báo lỗi a chưa khai báo
- (B) a trong sub1 (//2)
- (C) a trong sub2 (//3)

- (D) a trong main (//1)
- (E) a trong sub3
- 11. Giả sử chuỗi gọi là main  $\rightarrow$  sub1  $\rightarrow$  sub2  $\rightarrow$  sub3 , tham khảo đến  $\bf a$  trong sub3  $\bf khi$  thực thi ứng với kết hợp của  $\bf a$  trong bản hoạt động của
- (A) main
- (B) sub1
- (C) sub2
- (D) Báo lỗi không tìm thấy a

- (E) sub3
  - 12. Nếu đoạn code trên được viết trên ngôn ngữ dùng qui tắc tầm vực động (dynamic-scope rule), và giả sử chuỗi gọi là main  $\rightarrow$  sub1  $\rightarrow$  sub2  $\rightarrow$  sub3, tham khảo đến a trong sub3 **khi thực thi** ứng với kết hợp của a trong bản hoạt động của
- (A) Báo lỗi không tìm thấy a
- (B) main
- (C) sub1
- (D) sub3

- (E) sub2
- 13. Đoạn code viết bằng Scala nào dưới đây có thể dùng như một ví dụ minh họa cho khái niệm Hàm Currying
- (A) def add1(n:Int) = n + 1
- (B)  $\operatorname{List}(1,2,3).\operatorname{foldLeft}(0)(\underline{\ }+\underline{\ })$
- (C)  $\operatorname{def} \operatorname{add}(\operatorname{n:Int}) = (\operatorname{x:Int}) => \operatorname{n} + \operatorname{x;val} \operatorname{add} 2 = \operatorname{add}(2)$
- (D) List(1,2,3).filter( > 1)
- (E)  $\operatorname{def} \operatorname{add}(n:\operatorname{Int})(x:\operatorname{Int}) = n + x$ ;  $\operatorname{val} \operatorname{add}2 = \operatorname{add}(2)$ 
  - 14. Gọi append là hàm dùng để nối 2 danh sách thành 1 danh sách. Ví dụ append([1,2,3],[4,5,6]) sẽ có kết quả là [1,2,3,4,5,6]. Hãy hiện thực hàm append(a:List[Int],b:List[Int]) dùng hàm bậc cao (high-order function)?
- (A) a.map(b)((x,y)=>x::y)

(B) a.foldLeft(b)((x,y)=>x::y)

(C) a.foldRight(b)((x,y)=>x::y)

(D) b.foldRight(a)((x,y)=>x::y)

- (E) b.foldLeft(a)((x,y)=>x::y)
  - 15. Một biến trên ngôn ngữ Javascript có thể nhận những giá trị thuộc các kiểu khác nhau. Ví dụ:

```
var x = "def";
```

x = 1;

Ràng buộc giữa một biến và kiểu của nó trên ngôn ngữ Javascript xảy ra trong thời gian

- (A) Hiện thực (implementation)
- (B) Chay (running)

(C) Lập trình (programming)

(D) Khởi động (loading)

(E) Dich (compiling)



16. Kiểu *union* của ngôn ngữ C++ cho phép ghi vào ở thành phần này nhưng đọc lên ở thành phần khác như trong ví dụ sau:

```
union {
    int songuyen;
    char kytu[2];
} x;
x.songuyen = 12;
cout << x.kytu[0] << " " << x.kytu[1] << endl;</pre>
```

Đặc tính này của ngôn ngữ C++ gây ảnh hưởng XẤU NHẤT đối với

- (A) tính trực giao (orthonogality)
- (B) tính đơn giản (simplicity)

(C) tính dễ viết (writability)

(D) chi phí (cost)

(E) tính tin cậy (reliability)

Phần trình bày sau dùng trong các câu hỏi 17– 18:

Cho mã giả của phát biểu  $for \ var = expr1 \ to \ expr2 \ do \ body$  như sau:

- 17. Nếu trong thân *body* có thực thi lệnh *continue* thì điều khiển sẽ chuyển đến thực thi phát biểu có nhãn là:
- (A) out
- (B) start
- (C) loop
- (D) label2
- (E) label1
- 18. Với hiện thực của phát biểu *for* như trên, phát biểu nào đặt trong thân phát biểu **for** dưới đây sẽ làm cho phát biểu này bị lặp mãi mãi?

```
n = 10;s = 0;
for i = s to n do ______
```

(A) i = i - 1;

(B) s = s - 1;

(C) n = n + 1;

- (D) câu A và C đúng
- (E) Không phát biểu nào trong các câu A, B và C làm phát biểu for lặp mãi mãi

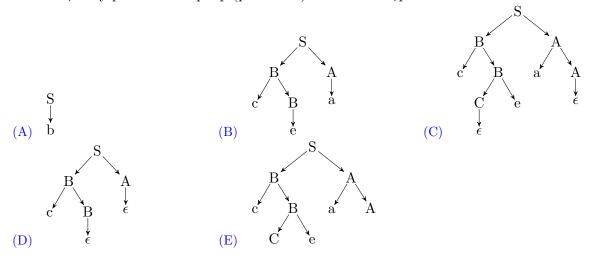
Cho văn phạm sau dùng cho các câu 19-20:

19. Chuỗi nào KHÔNG thuộc ngôn ngữ được mô tả bởi văn phạm trên?

- (A) e
- (B) b
- (C) eaaa
- (D) cccaaa
- (E) cccddddeaa



20. Chọn cây phân tích cú pháp (parse tree) cho chuỗi nhập: cea



- 21. Cơ chế gọi chương trình nào là cơ chế gọi chương trình con cơ bản của Lập trình hướng sự kiện (Event-driven Programming)
- (A) Biến cố Xử lý biến cố (Exception)
- (B) Gọi trở về đơn giản (Simple Call Return)
- (C) Gọi đệ qui (Recursive call)
- (D) Trình định thời (Scheduled Subprogram)
- (E) Trình cộng hành (Coroutine)
  - 22. Khi cần khai thác đặc tính của kiểu union(các thành phần loại trừ lẫn nhau), làm thế nào để hiện thực trên những ngôn ngữ lập trình hướng đối tượng như Java(không có kiểu union)? Cụ thể làm thế nào để hiện thực biến z kiểu union sau trên những ngôn ngữ này?

```
union {
    string loikhen;
    string loiche;
} z;
```

- (A)  $\stackrel{\text{Dịnh nghĩa lớp cha A và hai lớp con B (có thuộc tính loikhen) và C (có thuộc tính loiche), và z có kiểu A$
- (B) Định nghĩa một lớp A có 2 thuộc tính loikhen và loiche; và z có kiểu A
- (C) Định nghĩa lớp A có thuộc tính loikhen và lớp B(con của A) có thuộc tính loiche, và z có kiểu A
- (D) Định nghĩa lớp trừu tượng (abstract class) A và 2 lớp con B (có thuộc tính loikhen) và C (có thuộc tính loiche), và z có kiểu A
- (E) Định nghĩa lớp A có thuộc tính lo<br/>ikhen và lớp B(con của A) có thuộc tính loiche, và z có kiểu B
- 23. Giả sử chương trình có một lỗi văn phạm nằm sau lệnh **print**. Khi em nhấn nút Run (để dịch và chạy chương trình) trên trình soạn thảo, kết quả được in ra bởi lệnh **print** trên trước khi lỗi văn phạm được báo. Hỏi chế độ dịch của trình soạn thảo là gì?
- (A) trình liên kết (link editor)

(B) trình biên dịch (compiler)

- (C) trình thông dich(interpreter)
- (D) trình biên dịch động(just-in-time compiler)
- (E) trình hợp ngữ (assembler)



24. Cho lớp(class) A là lớp cha (superclass) của lớp B và B là lớp cha của hai lớp C và D. Trong lớp A, có khai báo phương thức thực thể (instance method) foo và phương thức này bị ghi đè(override) trên tất cả các lớp B, C và D. Cho biến x được khai báo như sau: B x;

Qui ước viết func<sub>T</sub> là phương thức func được khai báo trong lớp T, cho biết tập của phương thức đích của cuộc gọi sau: **x.foo()**;

- (A)  $\{foo_C, foo_D\}$
- (B)  $\{foo_A, foo_B, foo_C, foo_D\}$  (C)  $\{foo_B\}$

- (D)  $\{foo_B, foo_C, foo_D\}$
- $(E) \{foo_A, foo_B\}$
- 25. Cơ chế gọi chương trình con nào mà điều khiển có thể chuyển vào một vị trí ở giữa chương trình con được gọi thay vì chuyển vào đầu chương trình con được gọi?
- (A) Trình định thời (Scheduled Subprogram)
- (B) Gọi trở về đơn giản (Simple Call Return)
- (C) Gọi đệ qui (Recursive call)
- (D) Trình cộng hành (Coroutine)
- (E) Biến cố Xử lý biến cố (Exception)
- 26. Cho biết mã Jasmin của phát biểu gán sau:

$$a[i] = a[i] + 2$$

với i kiểu nguyên có chỉ số là 3 và a là kiểu dãy nguyên có chỉ số là 1.

(A)	iastore iadd iaload aload_1 iload_3 dup iconst_2	aload_1 iload_3 iaload B) iconst_2 iadd iastore	(C)	aload_1 iload_3 aload_1 iload_3 iaload iconst_2 iadd iastore	(D)	aload_1 iload_3 iastore aload_1 iload_3 iconst_2 iadd	(E)	aload_1 dup iload_3 dup iaload iconst_2 iadd iastore
-----	--	---	-----	--	-----	---	-----	--

27. Cho giá trị ban đầu của biến c là 3, cho biết những giá trị có thể có của biến a sau khi thực hiện phép gán sau trên ngôn ngữ lập trình C? a = c \* (c = 5);

- (A) 15, 25, 9
- (B) 15
- (C) 25
- (D) 15, 25
- (E) 9
- 28. Viết lại biểu thức trung tố (infix) sau sang dạng hậu tố (postfix) Polish? Giả sử là tất cả các phép toán đều có 2 toán hang và có đô ưu tiên và tính kết hợp như trong ngôn ngữ lập trình C.

$$a + b * c - d * e + f$$

- (A) a + b c \* d e \* f + (B) a b + c \* d e \* f + (C) a b c \* + d e f \* +

- (D) a b c \* + d e \* f +
- (E) a b c \* + d e \* + f -

Cho văn pham của một biểu thức sau dùng cho các câu 29-31:

```
\rightarrow term '=' exp | term
  exp
        \rightarrow term '+' fact | term '>' fact | fact
 term
         \rightarrow ope '*' ope | ope
        \rightarrow '(' exp ')' | ID
với ID là một danh hiệu.
```

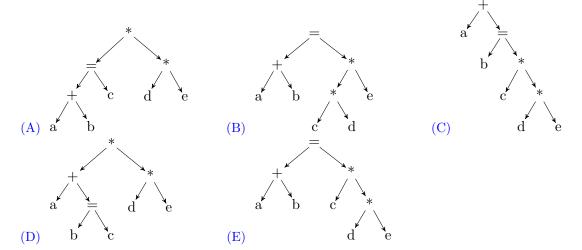


- 29. Chọn biểu thức được viết đúng văn phạm, có số lượng dấu '(' và ')' ít nhất mà tương đương với biểu thức sau: (a + (b = ((c \* d) \* e))) > f
- (A) (a + (b = (c \* d) \* e)) > f
- (B) (a + (b = ((c \* d) \* e))) > f

(C) a + b = c \* d \* e > f

(D) a + (b = c \* d \* e) > f

- (E) a + (b = (c \* d) \* e) > f
- 30. Vẽ AST cho biểu thức sau được viết dựa vào văn phạm trên:  $\mathbf{a} + \mathbf{b} = \mathbf{c} * (\mathbf{d} * \mathbf{e})$



31. Cho cấu trúc AST được khai báo trên Scala như sau: trait Exp case class Bin(op:String,e1:Exp,e2:Exp) extends Exp

case class Bin(op:String,e1:Exp,e2:Exp) extends Exp case class Id(i:String) extends Exp

Chọn AST cho biểu thức sau:  $\mathbf{a} + \mathbf{b} > \mathbf{c} = \mathbf{d} * \mathbf{e} = \mathbf{f}$ 

- (A) Bin("=",Bin(">",Bin("+",Id("a"),Id("b")),Id("c")),Bin("\*",Id("d"),Bin("=",Id("e"),Id("f")))
- (B) Bin("=",Bin("=",Bin(">",Bin("+",Id("a"),Id("b")),Id("c")),Bin("\*",Id("d"),Id("e"))),Id("f"))
- (C) Bin("=",Bin(">",Bin("+",Id("a"),Id("b")),Id("c")),Bin("=",Bin("\*",Id("d"),Id("e"))),Id("f"))
- (D) Bin("=",Bin("=",Bin("+",Id("a"),Bin(">",Id("b"),Id("c"))),Bin("\*",Id("d"),Id("e"))),Id("f"))
- (E) Bin("=",Bin("+",Id("a"),Bin(">",Id("b"),Id("c"))),Bin("=",Bin("\*",Id("d"),Id("e"))),Id("f"))



32. Cho khai báo sau được viết trên ngôn ngữ lập trình Ada:

Cho kích thước của các kiểu *Integer, Float, Boolean và Enumeration* lần lượt là 2, 4, 1, và 2, và kích thước cho phần mô tả kiểu (type description) là 0. Cho biết kích thước của một đối tượng kiểu **Figure**?

- (A) 21
- (B) 23
- (C) 13
- (D) 15
- (E) Khác

Đoạn code sau dùng cho các câu 33-35

```
int p;
int* foo(int x) {
    static int q;
    int *s = new int;
    switch (x) {
        case 1: return &p;
        case 2: return &q;
        case 3: return &x;
        case 4: return s;
        default: return foo(x-1);
    }
}
```

- 33. Phát biểu nào sẽ gây ra lỗi tham chiếu treo (dangling reference) khi thực thi
- (A) return &p
- (B) return &x
- (C) return &q
- (D) return s

- (E) Không phát biểu nào gây ra lỗi trên khi thực thi
- 34. Trong đoạn mã trên, đối tượng dữ liệu nào có thể trở thành rác (garbage)
- (A) r

(B) x

(C) q

(D) trỏ đến bởi s

- (E) Không có đối tượng nào có thể trở thành rác
- 35. Khi hàm **foo** được gọi đệ quy (recursive) thì các bản hoạt động của **foo** dùng chung những đối tượng dữ liệu nào?
- (A) q và s
- (B) p, q, x và s
- (C) p
- (D) x, q và s
- (E) p và q

Phần hướng dẫn này áp dụng cho các câu 36–40

Hãy viết thêm các đoạn mã cần thiết vào phương thức dưới đây để thực hiện sinh mã cho phát biểu *repeat*. Nhắc lại phát biểu *repeat* kết thúc quá trình lặp khi điều kiện của phát biểu trở thành **true**. AST của phát biểu **repeat** được định nghĩa như sau: case class Repeat(val stmts:List[Stmt],val exp:Expr) extends Stmt



```
override def visitRepeat(ast: Repeat, o:Context)= {
        val ctxt = o.asInstanceOf[SubBody]
        ctxt.frame.enterLoop();
        val labelStart = ctxt.frame.getNewLabel()
        val labelBreak = ctxt.frame.getBreakLabel()
        val labelCont = ctxt.frame.getContinueLabel()
        val str1 = //1
        // sinh mã cho từng phát biểu trong thân của repeat
        val str2 = //2
        val str3 = //3
        // sinh mã cho biểu thức điều kiện
        val str4 = visit(ast.exp,....)
        val str5 = //4
        val str6 = //5
        frame().exitLoop();
        str1 + str2 + str3 + str4 + str5 + str6
    }
 36. Đoạn mã nào trong số các đoạn mã dưới đây cần phải xuất hiện ở vi trí //1 trong đoạn
     mã trên
(A) ""
                                              (B) ctxt.emit.emitLABEL(labelStart)
(C) ctxt.emit.emitLABEL(labelBreak)
                                              (D) ctxt.emit.emitGOTO(labelStart)
(E) ctxt.emit.emitLABEL(labelCont)
 37. Đoạn mã nào trong số các đoạn mã dưới đây cần phải xuất hiện ở vị trí //2 trong đoạn
     mã trên
(A) visit(ast.stmts,o)
(B) ast.stmts.map(x = > visit(x, o))
(C) ast.stmts.filter(x = > visit(x, o))
(D) ast.stmts.foldLeft("")((x,y)=>y + visit(x,o))
(E) ast.stmts.foldLeft("")((x,y)=>x + visit(y,o))
 38. Đoạn mã nào trong số các đoạn mã dưới đây cần phải xuất hiện ở vị trí //3 trong đoạn
     mã trên
(A) ""
                                              (B) ctxt.emit.emitIFFALSE(labelStart)
(C) ctxt.emit.emitLABEL(labelBreak)
                                              (D) ctxt.emit.emitGOTO(labelStart)
(E) ctxt.emit.emitLABEL(labelCont)
 39. Đoạn mã nào trong số các đoạn mã dưới đây cần phải xuất hiện ở vị trí //4 trong đoạn
     mã trên
(A) ctxt.emit.emitGOTO(labelBreak)
                                              (B) ctxt.emit.emitIFFALSE(labelStart)
(C) ctxt.emit.emitLABEL(labelBreak)
                                              (D) ctxt.emit.emitGOTO(labelStart)
(E) ctxt.emit.emitLABEL(labelCont)
 40. Đoạn mã nào trong số các đoạn mã dưới đây cần phải xuất hiện ở vị trí //5 trong đoạn
     mã trên
```

(B) ctxt.emit.emitIFFALSE(labelStart))

(D) ctxt.emit.emitGOTO(labelStart))

(A) ctxt.emit.emitGOTO(labelBreak))

(C) ctxt.emit.emitLABEL(labelBreak))

(E) ctxt.emit.emitLABEL(labelCont))



41. Để thực hiện tìm kiếm một chuỗi (String) trên nhiều loại danh sách khác nhau, hãy chọn giải pháp thích hợp nhất để điền vào chỗ trống trong định nghĩa hàm sau:

```
def lookup[T](x:String,lst:List[T],f:T=>String):Option[T] =
    lst match {
        case List() => None
        case h::t =>______
}

(A) if (h == x) Some(h) else lookup(x,t,f)
(B) if (f(h) == x) Some(h) else lookup(x,t,f)
(C) if (h == f(x) Some(h) else lookup(x,t,f)
(D) if (h == f(x)) Some(f(h)) else lookup(x,t,f)
(E) if (f(h) == x) Some(f(h)) else lookup(x,t,f)
```

Đoạn mã sau được dùng cho các câu 42–47.

Cho một đoạn chương trình được viết trên một ngôn ngữ tựa C như sau:

```
int A[5] = {1,3,5,7,9}; // index of A starts from 0
int j = 0;
int n = 5;
int sumAndIncrease(int a, int i) {
    int s = 0;
    for (; i < n; i = i + 1) {
        s = s + a;
        A[j] = A[j] + 1;
    }
    return s;
}

void main() {
    int s = sumAndIncrease(A[j],j);
    printf("a = %i\n",s); //1
    printf("%i %i %i %i %i\n",A[0],A[1],A[2],A[3],A[4]); //2
}</pre>
```

42. Nếu a và i được truyền bằng **trị-kết quả (passed by value-result)**, giá trị của biến s được in ra ở phát biểu //1 là bao nhiêu?

- (A) 5
- (B) 15
- (C) 10
- (D) 25
- (E) Khác
- 43. Nếu *a* và *i* được truyền bằng **trị-kết quả (passed by value-result)**, giá trị của biến *A* được in ra ở phát biểu //2 là bao nhiêu?
- (A) 1 3 5 7 9
- (B) 6 3 5 7 9
- (C) 2 4 6 8 10
- (D) 5 3 5 7 9
- (E) Khác

44. Nếu a và i được truyền bằng **tham khảo (passed by reference)**, giá trị của biến s được in ra ở phát biểu //1 là bao nhiêu?

- (A) 5
- (B) 15
- (C) 10
- (D) 25
- (E) Khác

45. Nếu a và i được truyền bằng **tham khảo (passed by reference)**, giá trị của biến A được in ra ở phát biểu //2 là bao nhiêu?

- (A) 1 3 5 7 9
- (B) 6 3 5 7 9
- (C) 2 4 6 8 10
- (D) 5 3 5 7 9
- (E) Khác



46.	Nếu $a$ và $i$ được truyền bằng <b>tên</b>	(passed by	name),	giá trị của	a biến $s$ c	được in ra	ở phá
	biểu //1 là bao nhiêu?						

(A) 5

- (B) 20
- (C) 15
- (D) 25
- (E) Khác
- 47. Nếu a và i được truyền bằng **tên (passed by name)**, giá trị của biến A được in ra ở phát biểu //2 là bao nhiêu?
- (A) 1 3 5 7 9
- (B) 6 3 5 7 9
- (C) 2 4 6 8 10
- (D) 5 3 5 7 9
- (E) Khác

II. Phần câu hỏi tự luận: (dành cho tất cả sinh viên) (2 điểm)

48. Trên những ngôn ngữ lập trình cho phép rút ngắn tính toán (short-circuit evaluation) biểu thức luận lý, biểu thức sau sẽ không gây ra lỗi khi thực thi. Biết hàm sqrt(a) trả về  $\sqrt{a}$  ((x < 0) || (sqrt(x) > 4))

```
Hãy dùng if then else để mô phỏng quá trình tính toán của biểu thức luận lý trên?
```

49. Hãy viết lại biểu thức ở dạng trung tố (infix) sau sang dạng biểu thức tiền tố (prefix) Cambridge Polish:

```
a - b * c - d * e + f
```

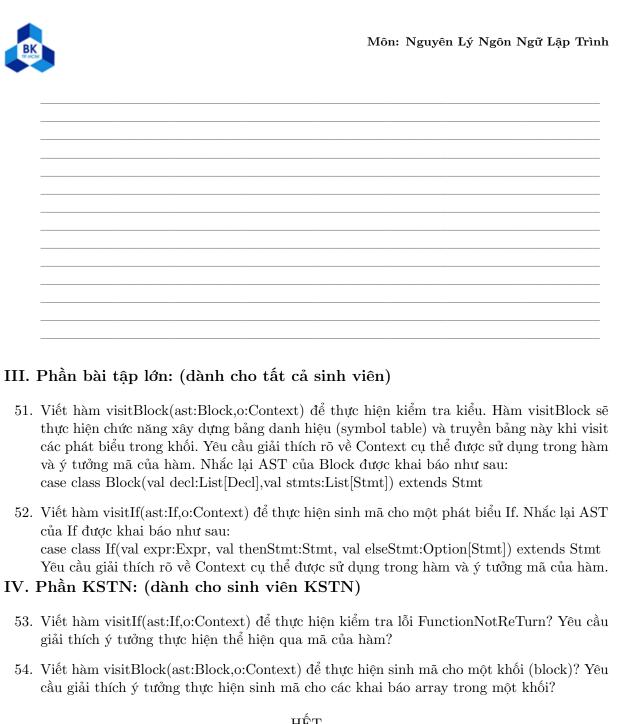
Biết rằng độ ưu tiên và tính kết hợp của các phép toán trong biểu thức như thông thường (đều kết hợp trái và \* có ưu tiên cao hơn +, -). Trong biểu thức tiền tố nhiều toán hạng, thứ tự tính toán cũng từ trái sang phải.

Yêu cầu: Biểu thức dạng tiền tố Cambridge Polish phải thỏa các yêu cầu sau:

- Thứ tự xuất hiện các toán hạng trong biểu thức dạng tiềm tố phải có cùng thứ tự xuất hiện như trong biểu thức trung tố
- Số ( và ) là ít nhất
- Có cùng thứ tự tính toán các phép toán với thứ tự đó trong biểu thức dạng trung tố

50. Hãy nêu những giải pháp đã được sử dụng để tránh tham chiếu treo (dangling reference) trong trường hợp trả về hàm như sau:

```
void->int F() {
    int x = 1;
    int g() {
        return x + 1;
    }
    return g;
}
void->int gg = F();
int z = gg();
```



94.	viet nam visit block (ast. block, o. Context) de thậc mện sinh mà cho một khôi (block): Te
	cầu giải thích ý tưởng thực hiện sinh mã cho các khai báo array trong một khối?
_	HÊT

Họ và tên:	Điểm:
MSSV:	





Chủ nhiệm bộ môn	Giảng viên ra đề
Chữ kí:	Chữ kí:
Họ tên:	Họ tên:



# KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH $\mbox{BỘ MÔN KHOA HỌC MÁY TÍNH} \label{eq:homographical}$

# ĐÁP ÁN cho Đề thi cuối kỳ

Mã	Sinh viên được ph . đề: 1154	ep sư dụ	ng tài l	iệu 		Sinh vi	ên không đượ	ợc sử d	lụng tài liệu
I. P	hần câu hỏi tr	rắc ngh	iệm:(8	điểi	m))				
1.	Cho biết mã Jas a * b - 2 với a và b là 2 b						C như sau:		
A	iload_0 iload_1 imul iconst_2 isub	iload iload icons imul isub 	_1	(C)	iload_0 imul iload_1 isub iconst_2	(D)	imul load_0 iload_1 isub iconst_2	(E)	iload_0 iload_1 iconst_2 isub imul
Đoạ	n mã sau được dù	ng trong	; các câi	u 2– 3	3:				
void } foo(	*p = new int; d foo(int * r) delete r; (p); //1 = 2; //2	{							
2.	Hiện tượng gì xã	ảy ra khi	p được	truyé	ền cho r ở ph	nát biể	ı //1 trong đ	toạn m	ã trên:
В	Con trỏ chưa kha Bí danh (alias) Rác (garbage)	u báo (u	ndeclare	ed poi	(C) I		io trùng tên hiếu treo (da	`	/
3.	Hiện tượng gì xã	ảy ra khi	thực th	ni phá	t biểu $//2$ t	rong đơ	oạn mã trên:		
	Tham chiếu treo		(B) E	3í dar Rác	ıh		(C) Đa hìn	nh (po	lymorphism)

(C) (b\*ab\*)\*

D b\*a?(bb\*a)\*b\*

(B) b\*(abb\*)\*

 $(A) a|(abb^*)^*$ 

(E) b\*ab\*ab\*



- 5. Cho biết kích thước của đối tượng dữ liệu  ${\bf x}$  được khai báo như sau:  ${\bf x}$ : set of 1..16
- (A) 4 bytes
- B 2 bytes
- (C) 4 bits
- (D) 16 bytes
- (E) 1 byte
- 6. Chọn phát biểu ĐÚNG trong các phát biểu về dãy (array) sau:
- (A) Trên C, kích thước của dãy có thể tăng thêm sau khi dãy đã được tạo ra
- (B) Truy xuất một phần tử bên ngoài một dãy C không được phép, sẽ lập tức gây ra lỗi
- (C) Trên C và C++, kích thước của một dãy được lưu trữ để kiểm tra động
- D Truy xuất một phần tử bên ngoài một dãy Java không được phép, sẽ lập tức gây ra lỗi
- (E) Trên Java, kích thước của một dãy KHÔNG được lưu trữ để kiểm tra động
  - 7. Trình định thời (scheduled subprograms) thường được dùng trong
- (A) Lập trình song song (Parallel Programming)
- (B) Lập trình hướng đối tượng (Object-Oriented Programming)
- (C) Lập trình hàm (Functional Programming)
- D Lập trình thời gian thực (Real-time Programming)
- (E) Lập trình hướng sự kiện (Event-driven Programming)
  - 8. Liệt kê theo thứ tự chiều dài tăng dần ít nhất 5 chuỗi ngắn nhất (nếu có nhiều hơn 5 chuỗi) của ngôn ngữ được mô tả bởi biểu thức chính quy (regular expression) sau: ab\*(a|b)? a
- (A) aa, aba, aaa, abaa, aaba
- (B)  $\epsilon$ , a, aa, aba, abaa
- C aa, aba, aaa, abaa, abba

- (D) aa, aba, aaa, abaa, aaaa
- (E) a, aa, aba, aaa, aaba

Đoạn code sau, được viết trên ngôn ngữ dùng **qui tắc tầm vực tĩnh (static-scope rule)**, áp dụng cho các câu 9–12

9. Môi trường tham khảo tĩnh (static referencing environment) của thủ tục **sub3** KHÔNG chứa

- A a ở //1
- (B) sub1
- (C) a ở //2
- (D) sub2
- (E) sub3



10. Giả sử chuỗi gọi là main khai báo	$\rightarrow$ sub1 $\rightarrow$ sub2 $\rightarrow$ sub3,	tham khảo đến <b>a</b>	trong <b>sub3</b> ứng vớ
A) Báo lỗi a chưa khai báo D) a trong main (//1)	B a trong sub1 (//2) (E) a trong sub3	(C) a trong	g sub2 $(//3)$
11. Giả sử chuỗi gọi là main thi ứng với kết hợp của a	$ ightarrow \operatorname{sub1}  ightarrow \operatorname{sub2}  ightarrow \operatorname{sub3}$ , a trong bản hoạt động của		rong sub3 <b>khi thực</b>
A) main E) sub1	(C) sub2	(D) Báo lỗi kh	nông tìm thấy a
12. Nếu đoạn code trên được rule), và giả sử chuỗi gọi <b>khi thực thi</b> ứng với kết	là main $\rightarrow$ sub1 $\rightarrow$ sub2	ightarrow sub3 , tham khå	- , -
A) Báo lỗi không tìm thấy a E sub2	(B) main	(C) sub1	(D) sub3
13. Đoạn code viết bằng Sca niệm Hàm Currying	la nào dưới đây có thể di	ùng như một ví dụ	minh họa cho khá
A) def add1(n:Int) = n + 1 B) List(1,2,3).foldLeft(0)(_ + C) def add(n:Int) = (x:Int) = D) List(1,2,3).filter(_ > 1) E def add(n:Int)(x:Int) = n	> n + x;val add2 = add(		
14. Gọi append là hàm dùng c sẽ có kết quả là [1,2,3,4,5 bậc cao (high-order funct	,6]. Hãy hiện thực hàm ap		
A) a.map(b)((x,y)=>x::y) C a.foldRight(b)((x,y)=>x::y E) b.foldLeft(a)((x,y)=>x::y)		a.foldLeft(b)((x,y) b.foldRight(a)((x,y)	•
15. Một biến trên ngôn ngữ $V$ í dụ: $ var                                  $	Javascript có thể nhận nh và kiểu của nó trên ngôn		
<ul><li>A) Hiện thực (implementation</li><li>C) Lập trình (programming)</li><li>E) Dịch (compiling)</li></ul>	-	hạy (running) thởi động (loading)	



16. Kiểu union của ngôn ngữ C++ cho phép ghi vào ở thành phần này nhưng đọc lên ở thành phần khác như trong ví dụ sau:

```
union {
    int songuyen;
    char kytu[2];
} x;
x.songuyen = 12;
cout << x.kytu[0] << " " << x.kytu[1] << endl;</pre>
```

Đặc tính này của ngôn ngữ C++ gây ảnh hưởng  $X \hat{A} U NH \hat{A} T$  đối với

- (A) tính trực giao (orthonogality)
- (B) tính đơn giản (simplicity)

(C) tính dễ viết (writability)

(D) chi phí (cost)

E tính tin cậy (reliability)

Phần trình bày sau dùng trong các câu hỏi 17– 18:

Cho mã giả của phát biểu for var = expr1 to expr2 do body như sau:

- 17. Nếu trong thân body có thực thi lệnh continue thì điều khiển sẽ chuyển đến thực thi phát biểu có nhãn là:
- (A) out
- (B) start
- (C) loop
- (D) label2
- E label1

18. Với hiện thực của phát biểu *for* như trên, phát biểu nào đặt trong thân phát biểu **for** dưới đây sẽ làm cho phát biểu này bị lặp mãi mãi?

```
n = 10;s = 0;
for i = s to n do _____
```

|A| i = i - 1;

(B) s = s - 1;

(C) n = n + 1;

- (D) câu A và C đúng
- (E) Không phát biểu nào trong các câu A, B và C làm phát biểu for lặp mãi mãi

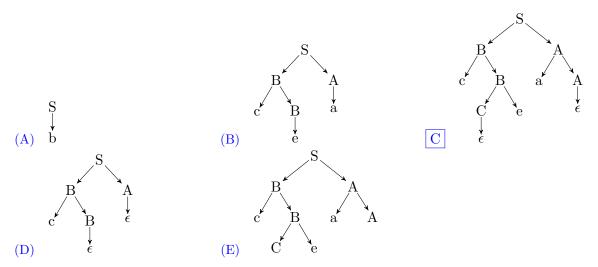
Cho văn phạm sau dùng cho các câu 19–20:

19. Chuỗi nào KHÔNG thuộc ngôn ngữ được mô tả bởi văn phạm trên?

- (A) e
- (B) b
- (C) eaaa
- D cccaaa
- (E) cccddddeaa



20. Chọn cây phân tích cú pháp (parse tree) cho chuỗi nhập: cea



- 21. Cơ chế gọi chương trình nào là cơ chế gọi chương trình con cơ bản của Lập trình hướng sự kiện (Event-driven Programming)
- A Biến cố Xử lý biến cố (Exception)
- (B) Gọi trở về đơn giản (Simple Call Return)
- (C) Gọi đệ qui (Recursive call)
- (D) Trình định thời (Scheduled Subprogram)
- (E) Trình cộng hành (Coroutine)
  - 22. Khi cần khai thác đặc tính của kiểu union(các thành phần loại trừ lẫn nhau), làm thế nào để hiện thực trên những ngôn ngữ lập trình hướng đối tượng như Java(không có kiểu union)? Cụ thể làm thế nào để hiện thực biến z kiểu union sau trên những ngôn ngữ này?

```
union {
    string loikhen;
    string loiche;
} z;
```

- (A)  $\stackrel{\text{Dịnh nghĩa lớp cha A và hai lớp con B (có thuộc tính loikhen) và C (có thuộc tính loiche), và z có kiểu A$
- (B) Đinh nghĩa một lớp A có 2 thuộc tính loikhen và loiche; và z có kiểu A
- (C) Định nghĩa lớp A có thuộc tính lo<br/>ikhen và lớp B(con của A) có thuộc tính loiche, và z có kiểu A
- Dịnh nghĩa lớp trừu tượng (abstract class) A và 2 lớp con B (có thuộc tính loikhen) và C (có thuộc tính loiche), và z có kiểu A
- (E)  $\lim_{x \to \infty} \frac{1}{x} \operatorname{B}(x) = 0$  Dịnh nghĩa lớp A có thuộc tính lo<br/>iche, và z có kiểu B
- 23. Giả sử chương trình có một lỗi văn phạm nằm sau lệnh **print**. Khi em nhấn nút Run (để dịch và chạy chương trình) trên trình soạn thảo, kết quả được in ra bởi lệnh **print** trên trước khi lỗi văn phạm được báo. Hỏi chế độ dịch của trình soạn thảo là gì?
- (A) trình liên kết (link editor)

(B) trình biên dịch (compiler)

- c trình thông dịch(interpreter)
- (D) trình biên dịch động(just-in-time compiler)
- (E) trình hợp ngữ (assembler)



24. Cho lớp(class) A là lớp cha (superclass) của lớp B và B là lớp cha của hai lớp C và D. Trong lớp A, có khai báo phương thức thực thể (instance method) foo và phương thức này bị ghi đè(override) trên tất cả các lớp B, C và D. Cho biến x được khai báo như sau: B x;

Qui ước viết func<sub>T</sub> là phương thức func được khai báo trong lớp T, cho biết tập của phương thức đích của cuộc gọi sau: **x.foo()**;

- (A)  $\{foo_C, foo_D\}$
- (B)  $\{foo_A, foo_B, foo_C, foo_D\}$  (C)  $\{foo_B\}$
- $D \mid \{foo_B, foo_C, foo_D\}$
- (E)  $\{foo_A, foo_B\}$
- 25. Cơ chế gọi chương trình con nào mà điều khiển có thể chuyển vào một vị trí ở giữa chương trình con được gọi thay vì chuyển vào đầu chương trình con được gọi?
- (A) Trình định thời (Scheduled Subprogram)
- (B) Gọi trở về đơn giản (Simple Call Return)
- (C) Gọi đệ qui (Recursive call)
- D Trình cộng hành (Coroutine)
- (E) Biến cố Xử lý biến cố (Exception)
  - 26. Cho biết mã Jasmin của phát biểu gán sau:

$$a[i] = a[i] + 2$$

với i kiểu nguyên có chỉ số là 3 và a là kiểu dãy nguyên có chỉ số là 1.

(A)	iastore iadd iaload aload_1 iload_3 dup iconst_2	(B)	aload_1 iload_3 iaload iconst_2 iadd iastore	C	aload_1 iload_3 aload_1 iload_3 iaload iconst_2 iadd iastore	(D)	aload_1 iload_3 iastore aload_1 iload_3 iconst_2 iadd	(E)	aload_1 dup iload_3 dup iaload iconst_2 iadd iastore
-----	--	-----	--	---	--	-----	---	-----	--

27. Cho giá trị ban đầu của biến **c** là 3, cho biết những giá trị **có thể có** của biến **a** sau khi thực hiện phép gán sau trên ngôn ngữ lập trình C? a = c \* (c = 5);

- (A) 15, 25, 9
- (B) 15
- (C) 25
- D 15, 25
- (E) 9

28. Viết lại biểu thức trung tố (infix) sau sang dạng hậu tố (postfix) Polish? Giả sử là tất cả các phép toán đều có 2 toán hang và có đô ưu tiên và tính kết hợp như trong ngôn ngữ lập trình C.

$$a + b * c - d * e + f$$

- (A) a + b c \* d e \* f + (B) a b + c \* d e \* f +
- (C) a b c \* + d e f \* +

- D a b c \* + d e \* f +
- (E) a b c \* + d e \* + f -

Cho văn pham của một biểu thức sau dùng cho các câu 29-31:

```
\rightarrow term '=' exp | term
           term '+' fact | term '>' fact | fact
            ope '*' ope | ope
fact
            '(' exp ')' | ID
ope
```



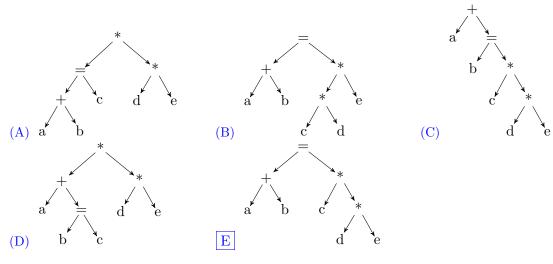
với ID là một danh hiệu.

- 29. Chọn biểu thức được viết đúng văn phạm, có số lượng dấu '(' và ')' ít nhất mà tương đương với biểu thức sau: (a + (b = ((c \* d) \* e))) > f
- (A) (a + (b = (c \* d) \* e)) > f
- (B) (a + (b = ((c \* d) \* e))) > f

(C) a + b = c \* d \* e > f

(D) a + (b = c \* d \* e) > f

- E = a + (b = (c \* d) \* e) > f
- 30. Vẽ AST cho biểu thức sau được viết dựa vào văn phạm trên:  $\mathbf{a} + \mathbf{b} = \mathbf{c} * (\mathbf{d} * \mathbf{e})$



31. Cho cấu trúc AST được khai báo trên Scala như sau: trait Exp case class Bin(op:String,e1:Exp,e2:Exp) extends Exp case class Id(i:String) extends Exp

Chọn AST cho biểu thức sau:  $\mathbf{a} + \mathbf{b} > \mathbf{c} = \mathbf{d} * \mathbf{e} = \mathbf{f}$ 

- (A) Bin("=",Bin(">",Bin("+",Id("a"),Id("b")),Id("c")),Bin("\*",Id("d"),Bin("=",Id("e"),Id("f"))))
- (B) Bin("=",Bin("=",Bin(">",Bin("+",Id("a"),Id("b")),Id("c")),Bin("\*",Id("d"),Id("e"))),Id("f"))
- (D) Bin("=",Bin("=",Bin("+",Id("a"),Bin(">",Id("b"),Id("c"))),Bin("\*",Id("d"),Id("e"))),Id("f"))
- (E) Bin("=",Bin("+",Id("a"),Bin(">",Id("b"),Id("c"))),Bin("=",Bin("\*",Id("d"),Id("e"))),Id("f"))



32. Cho khai báo sau được viết trên ngôn ngữ lập trình Ada:

Cho kích thước của các kiểu *Integer, Float, Boolean và Enumeration* lần lượt là 2, 4, 1, và 2, và kích thước cho phần mô tả kiểu (type description) là 0. Cho biết kích thước của một đối tượng kiểu **Figure**?

(A) 21

(B) 23



(D) 15

(E) Khác

Đoạn code sau dùng cho các câu 33-35

```
int p;
int* foo(int x) {
    static int q;
    int *s = new int;
    switch (x) {
        case 1: return &p;
        case 2: return &q;
        case 3: return &x;
        case 4: return s;
        default: return foo(x-1);
    }
}
```

33. Phát biểu nào sẽ gây ra lỗi tham chiếu treo (dangling reference) khi thực thi

- (A) return &p
- B return &x
- (C) return &q
- (D) return s

- (E) Không phát biểu nào gây ra lỗi trên khi thực thi
- 34. Trong đoạn mã trên, đối tượng dữ liệu nào có thể trở thành rác (garbage)
- (A) p

(B) x

(C) q

- D trỏ đến bởi s
- (E) Không có đối tượng nào có thể trở thành rác

35. Khi hàm **foo** được gọi đệ quy (recursive) thì các bản hoạt động của **foo** dùng chung những đối tượng dữ liệu nào?

- (A) q và s
- (B) p, q, x và s
- (C) p
- (D) x, q và s
- E p và q

Phần hướng dẫn này áp dụng cho các câu 36–40



Hãy viết thêm các đoạn mã cần thiết vào phương thức dưới đây để thực hiện sinh mã cho phát biểu *repeat*. Nhắc lại phát biểu *repeat* kết thúc quá trình lặp khi điều kiện của phát biểu trở thành **true**. AST của phát biểu **repeat** được định nghĩa như sau: case class Repeat(val stmts:List[Stmt],val exp:Expr) extends Stmt

```
override def visitRepeat(ast: Repeat, o:Context)= {
        val ctxt = o.asInstanceOf[SubBody]
        ctxt.frame.enterLoop();
        val labelStart = ctxt.frame.getNewLabel()
        val labelBreak = ctxt.frame.getBreakLabel()
        val labelCont = ctxt.frame.getContinueLabel()
        val str1 = //1
        // sinh mã cho từng phát biểu trong thân của repeat
        val str2 = //2
        val str3 = //3
        // sinh mã cho biểu thức điều kiện
        val str4 = visit(ast.exp,....)
        val str5 = //4
        val str6 = \frac{1}{5}
        frame().exitLoop();
        str1 + str2 + str3 + str4 + str5 + str6
   }
```

- 36. Đoạn mã nào trong số các đoạn mã dưới đây cần phải xuất hiện ở vị trí //1 trong đoạn mã trên
- (A) ""
- B ctxt.emit.emitLABEL(labelStart)
- (C) ctxt.emit.emitLABEL(labelBreak) (D) ctxt.emit.emitGOTO(labelStart)
- (E) ctxt.emit.emitLABEL(labelCont)
  - 37. Đoạn mã nào trong số các đoạn mã dưới đây cần phải xuất hiện ở vị trí //2 trong đoạn mã trên
- (A) visit(ast.stmts,o)
- (B) ast.stmts.map(x = > visit(x, o))
- (C) ast.stmts.filter(x = > visit(x, o))
- (D) ast.stmts.foldLeft("")((x,y)=>y + visit(x,o))
- E ast.stmts.foldLeft("")((x,y)=>x + visit(y,o))
- 38. Đoạn mã nào trong số các đoạn mã dưới đây cần phải xuất hiện ở vị trí //3 trong đoạn mã trên
- (A) ""
  (C) at the amit amit I ART (label Prock)
- (B) ctxt.emit.emitIFFALSE(labelStart)
- (C) ctxt.emit.emitLABEL(labelBreak)
- (D) ctxt.emit.emitGOTO(labelStart)
- E ctxt.emit.emitLABEL(labelCont)
- 39. Đoạn mã nào trong số các đoạn mã dưới đây cần phải xuất hiện ở vị trí //4 trong đoạn mã trên
- (A) ctxt.emit.emitGOTO(labelBreak)
- B ctxt.emit.emitIFFALSE(labelStart)
- (C) ctxt.emit.emitLABEL(labelBreak)
- (D) ctxt.emit.emitGOTO(labelStart)
- (E) ctxt.emit.emitLABEL(labelCont)



- 40. Đoạn mã nào trong số các đoạn mã dưới đây cần phải xuất hiện ở vị trí //5 trong đoạn mã trên
- (A) ctxt.emit.emitGOTO(labelBreak)) (B) ctxt.emit.emitIFFALSE(labelStart))
- C ctxt.emit.emitLABEL(labelBreak)) (D) ctxt.emit.emitGOTO(labelStart))
- (E) ctxt.emit.emitLABEL(labelCont))
  - 41. Để thực hiện tìm kiếm một chuỗi (String) trên nhiều loại danh sách khác nhau, hãy chọn giải pháp thích hợp nhất để điền vào chỗ trống trong định nghĩa hàm sau:

```
def lookup[T](x:String,lst:List[T],f:T=>String):Option[T] =
   lst match {
     case List() => None
     case h::t =>______
}
```

- (A) if (h == x) Some(h) else lookup(x,t,f)
- B if (f(h) == x) Some(h) else lookup(x,t,f)
- (C) if (h == f(x) Some(h) else lookup(x,t,f)
- (D) if (h == f(x)) Some(f(h)) else lookup(x,t,f)
- (E) if (f(h) == x) Some(f(h)) else lookup(x,t,f)

Đoạn mã sau được dùng cho các câu 42–47.

Cho một đoạn chương trình được viết trên một ngôn ngữ tựa C như sau:

```
int A[5] = {1,3,5,7,9}; // index of A starts from 0
int j = 0;
int n = 5;
int sumAndIncrease(int a, int i) {
    int s = 0;
    for ( ; i < n; i = i + 1) {
        s = s + a;
        A[j] = A[j] + 1;
    }
    return s;
}

void main(){
    int s = sumAndIncrease(A[j],j);
    printf("a = %i\n",s); //1
    printf("%i %i %i %i %i\n",A[0],A[1],A[2],A[3],A[4]); //2
}</pre>
```

42. Nếu a và i được truyền bằng **trị-kết quả (passed by value-result)**, giá trị của biến s được in ra ở phát biểu //1 là bao nhiêu?

- A 5
- (B) 15
- (C) 10
- (D) 25
- (E) Khác
- 43. Nếu *a* và *i* được truyền bằng **trị-kết quả (passed by value-result)**, giá trị của biến *A* được in ra ở phát biểu //2 là bao nhiêu?
- (A) 1 3 5 7 9
- B 63579
- (C) 2 4 6 8 10
- (D) 5 3 5 7 9
- (E) Khác



44.		ợc truyền bằng <b>than</b> iểu //1 là bao nhiêu	ν=	reference), giá t	rị của biến $s$ được			
(A)	5	(B) 15	(C) 10	(D) 25	E Khác			
45.		ược truyền bằng <b>th</b> a shát biểu //2 là bao	-	by reference), g	giá trị của biến $A$			
(A)	1 3 5 7 9	(B) 6 3 5 7 9	C 2 4 6 8 10	(D) 5 3 5 7 9	(E) Khác			
46.	Nếu $a$ và $i$ đư biểu $//1$ là ba	ợc truyền bằng <b>tên (</b> ao nhiêu?	(passed by name),	, giá trị của biến $s$	được in ra ở phát			
(A)	5	(B) 20	(C) 15	D 25	(E) Khác			
47.	Nếu $a$ và $i$ đư phát biểu $//2$	ược truyền bằng <b>tê</b> r là bao nhiêu?	n (passed by name	e), giá trị của biế	en $A$ được in ra ở			
(A)	1 3 5 7 9	(B) 6 3 5 7 9	C 2 4 6 8 10	(D) 5 3 5 7 9	(E) Khác			
II. I	Phần câu hỏ	i tự luận: (dành	cho tất cả sinh v	viên) (2 điểm)				
		hen else để mô phả			Iuạn ly tren?			
		$(\mathrm{x} < 0)$ then else i p đề xuất đúng: 2	f(sqrt(x) > 8) then	n				
49.	Hãy viết lại biểu thức ở dạng trung tố (infix) sau sang dạng biểu thức tiền tố (prefix) Cambridge Polish:							
	a - b * c - d * e + f							
	Biết rằng độ ưu tiên và tính kết hợp của các phép toán trong biểu thức như thông thường (đều kết hợp trái và $*$ có ưu tiên cao hơn $+$ , $-$ ). Trong biểu thức tiền tố nhiều toán hạng, thứ tự tính toán cũng từ trái sang phải.							
	Yêu cầu: Biểu thức dạng tiền tố Cambridge Polish phải thỏa các yêu cầu sau:							
	<ul> <li>Thứ tự xuất hiện các toán hạng trong biểu thức dạng tiềm tố phải có cùng thứ tự xuất hiện như trong biểu thức trung tố</li> <li>Số ( và ) là ít nhất</li> </ul>							
	<ul> <li>Số (và ) là 11 mat</li> <li>Có cùng thứ tự tính toán các phép toán với thứ tự đó trong biểu thức dạng trung tế</li> </ul>							
			r or other					



```
Lời giải. (+ (- a (* b c) (* d e) f)
Nếu viết đúng biểu thức dạng tiền tố Cambridge Polish: 1
Nếu viết đúng và có số () ít nhất: 1
```

50. Hãy nêu những giải pháp đã được sử dụng để tránh tham chiếu treo (dangling reference) trong trường hợp trả về hàm như sau:

```
void->int F() {
    int x = 1;
    int g() {
        return x + 1;
    }
    return g;
}
void->int gg = F();
int z = gg();
```

### III. Phần bài tập lớn: (dành cho tất cả sinh viên)

- 51. Viết hàm visitBlock(ast:Block,o:Context) để thực hiện kiểm tra kiểu. Hàm visitBlock sẽ thực hiện chức năng xây dựng bảng danh hiệu (symbol table) và truyền bảng này khi visit các phát biểu trong khối. Yêu cầu giải thích rõ về Context cụ thể được sử dụng trong hàm và ý tưởng mã của hàm. Nhắc lại AST của Block được khai báo như sau: case class Block(val decl:List[Decl],val stmts:List[Stmt]) extends Stmt
- 52. Viết hàm visitIf(ast:If,o:Context) để thực hiện sinh mã cho một phát biểu If. Nhắc lại AST của If được khai báo như sau: case class If(val expr:Expr, val thenStmt:Stmt, val elseStmt:Option[Stmt]) extends Stmt Yêu cầu giải thích rõ về Context cụ thể được sử dụng trong hàm và ý tưởng mã của hàm.

### IV. Phần KSTN: (dành cho sinh viên KSTN)

53. Viết hàm visitIf(ast:If,o:Context) để thực hiện kiểm tra lỗi FunctionNotReTurn? Yêu cầu giải thích ý tưởng thực hiện thể hiện qua mã của hàm?



54. Viết hàm visitBlock(ast:Block,o:Context) để thực hiện sinh mã cho một khối (block)? Yêu cầu giải thích ý tưởng thực hiện sinh mã cho các khai báo array trong một khối?

Question Distribution

Content	Level 1	Level 2	Level 3	Level 4	
Introduction		1			
Lexical	1	4	5		
Syntax	2	4	7		
AST		2		1	
Total	3	11	12	1	

-HÊT-

Chủ nhiệm bộ môn	Giảng viên ra đề
Chữ kí:	Chữ kí:
Họ tên:	Họ tên:



KHOA	KHOA	Học VÀ	ΚŶ	THUẬT	$\mathbf{M}\mathbf{\acute{A}}\mathbf{Y}$	TÍNH
	BÔ MÔ	N KHO	, но	C MÁY T	ΓÍΝΗ	

Họ và tên:
MSSV:

# Đề thi cuối kỳ

Môn thi: Nguyên Lý Ngôn Ngữ Lập Trình

Thời gian: 120 phút Ngày thi: 23-12-2015

□ Sinh viên được phép sử dụng tài liêu □ Sinh viên không được sử dụng tài liêu

Mã đề: 1155

- Sinh viên phải ghi tên và mã số sinh viên trên đề thi (**trang đầu và trang cuối**), giấy làm bài trắc nghiệm (giấy đỏ) và giấy làm bài. Sinh viên phải tô phần mã đề thi và mã số sinh viên trên giấy làm bài trắc nghiệm. Khi nộp bài, sinh viên phải nộp cả **đề thi, giấy làm bài trắc nghiệm và giấy làm bài**.
- Phần trắc nghiệm sẽ được chấm TỰ ĐỘNG trên giấy làm bài trắc nghiệm. Do đó, phần trắc nghiệm nếu làm trên đề thi sẽ KHÔNG được chấm.
- Đối với các câu hỏi phần trắc nghiệm, sinh viên chỉ chon MÔT phương án đúng nhất.
- Đối với phần câu hỏi tự luận (phần II), sinh viên làm ngay trên đề thi, ở phần dành riêng ngay dưới mỗi câu hỏi.
- Đối với câu hỏi bài tập lớn (phần III và IV), sinh viên làm trên giấy làm bài.
- Sinh viên lớp đại trà làm 3 phần (I, II và III). Sinh viên lớp tài năng làm cả 4 phần (I, II, III và IV). Chỉ có phần I và II được dùng để tính điểm cuối kỳ, các phần III và IV để tính điểm bài tập lớn 2 và 3.

### I. Phần câu hỏi trắc nghiệm:(8 điểm))

1. Cho biết mã Jasmin của phát biểu gán sau:

$$a[i] = a[i] + 2$$

với i kiểu nguyên có chỉ số là 3 và a là kiểu dãy nguyên có chỉ số là 1.

Đoạn code sau, được viết trên ngôn ngữ dùng **qui tắc tầm vực tĩnh (static-scope rule)**, áp dụng cho các câu 2–5



2. Môi trường tham khảo tĩnh (static referencing environment) của thủ tục **sub3** KHÔNG chứa

- (A) sub2
- (B) a ở //1
  - (C) sub1
- (D) a ở //2
- (E) sub3
- 3. Giả sử chuỗi gọi là main  $\to$  sub<br/>1  $\to$  sub2  $\to$  sub3, tham khảo đến **a** trong **sub3** ứng với khai báo
- (A) a trong sub1 (//2)
- (B) Báo lỗi a chưa khai báo
- (C) a trong sub3

- (D) a trong sub2 (//3)
- (E) a trong main (//1)
- 4. Giả sử chuỗi gọi là main  $\to$  sub1  $\to$  sub2  $\to$  sub3 , tham khảo đến  $\bf a$  trong sub3  $\bf khi$  thực thi ứng với kết hợp của  $\bf a$  trong bản hoạt động của
- (A) sub1
- (B) main
- (C) sub3
- (D) sub2

- (E) Báo lỗi không tìm thấy a
  - 5. Nếu đoạn code trên được viết trên ngôn ngữ dùng qui tắc tầm vực động (dynamic-scope rule), và giả sử chuỗi gọi là main  $\rightarrow$  sub1  $\rightarrow$  sub2  $\rightarrow$  sub3, tham khảo đến a trong sub3 khi thực thi ứng với kết hợp của a trong bản hoạt động của
- (A) main
- (B) Báo lỗi không tìm thấy a
- (C) sub2
- (D) sub1

- (E) sub3
  - 6. Trình định thời (scheduled subprograms) thường được dùng trong
- (A) Lập trình hướng đối tượng (Object-Oriented Programming)
- (B) Lập trình song song (Parallel Programming)
- (C) Lập trình hướng sự kiện (Event-driven Programming)
- (D) Lập trình hàm (Functional Programming)
- (E) Lâp trình thời gian thực (Real-time Programming)
  - 7. Liệt kê theo thứ tự chiều dài tăng dần ít nhất 5 chuỗi ngắn nhất (nếu có nhiều hơn 5 chuỗi) của ngôn ngữ được mô tả bởi biểu thức chính quy (regular expression) sau: ab\*(a|b)? a
- (A)  $\epsilon$ , a, aa, aba, abaa
- (B) aa, aba, aaa, abaa, aaba
- (C) a, aa, aba, aaa, aaba

- (D) aa, aba, aaa, abaa, abba
- (E) aa, aba, aaa, abaa, aaaa



- 8. Giả sử chương trình có một lỗi văn phạm nằm sau lệnh **print**. Khi em nhấn nút Run (để dịch và chạy chương trình) trên trình soạn thảo, kết quả được in ra bởi lệnh **print** trên trước khi lỗi văn phạm được báo. Hỏi chế độ dịch của trình soạn thảo là gì?
- (A) trình biên dich (compiler)

(B) trình liên kết (link editor)

(C) trình hợp ngữ (assembler)

- (D) trình thông dich(interpreter)
- (E) trình biên dich đông(just-in-time compiler)

Đoạn mã sau được dùng trong các câu 9–10:

```
int *p = new int;
void foo(int * r) {
    delete r;
}
foo(p); //1
*p = 2; //2
```

- 9. Hiện tượng gì xảy ra khi p được truyền cho r ở phát biểu //1 trong đoạn mã trên:
- (A) Bí danh (alias)
- (B) Con trỏ chưa khai báo (undeclared pointer) (C) Tham chiếu treo (dangling reference)
- (D) Khai báo trùng tên (redeclared)
- (E) Rác (garbage)
- 10. Hiện tượng gì xảy ra khi thực thi phát biểu //2 trong đoạn mã trên:
- (A) Bí danh

- (B) Tham chiếu treo
- (C) Rác

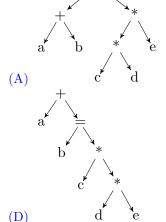
- (D) Da hình (polymorphism)
- (E) Con trỏ chưa khai báo

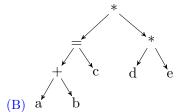
Cho văn phạm của một biểu thức sau dùng cho các câu 11–13:

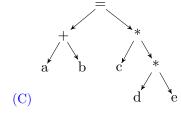
```
term '=' exp | term
 _{\text{term}}
               term '+' fact | term '>' fact | fact
               ope '*' ope | ope
  fact
               '(' exp ')' | ID
  ope
         \rightarrow
với ID là một danh hiệu.
```

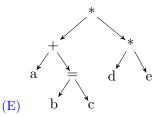
- 11. Chọn biểu thức được viết đúng văn phạm, có số lượng dấu '(' và ')' ít nhất mà tương đương với biểu thức sau:  $(\mathbf{a} + (\mathbf{b} = ((\mathbf{c} * \mathbf{d}) * \mathbf{e}))) > \mathbf{f}$
- (A) (a + (b = ((c \* d) \* e))) > f
- (B) (a + (b = (c \* d) \* e)) > f(D) a + b = c \* d \* e > f
- (C) a + (b = (c \* d) \* e) > f

- (E) a + (b = c \* d \* e) > f
  - 12. Vẽ AST cho biểu thức sau được viết dựa vào văn phạm trên:  $\mathbf{a} + \mathbf{b} = \mathbf{c} * (\mathbf{d} * \mathbf{e})$











13. Cho cấu trúc AST được khai báo trên Scala như sau:

trait Exp

case class Bin(op:String,e1:Exp,e2:Exp) extends Exp case class Id(i:String) extends Exp

Chon AST cho biểu thức sau:  $\mathbf{a} + \mathbf{b} > \mathbf{c} = \mathbf{d} * \mathbf{e} = \mathbf{f}$ 

- (A) Bin("=",Bin("=",Bin("+",Id("a"),Id("b")),Id("c")),Bin("\*",Id("d"),Id("e"))),Id("f"))
- (B) Bin("=",Bin(">",Bin("+",Id("a"),Id("b")),Id("c")),Bin("\*",Id("d"),Bin("=",Id("e"),Id("f"))))
- (C)  $\operatorname{Bin}("=",\operatorname{Bin}("+",\operatorname{Id}("a"),\operatorname{Bin}(">",\operatorname{Id}("b"),\operatorname{Id}("c"))),\operatorname{Bin}("=",\operatorname{Bin}("*",\operatorname{Id}("d"),\operatorname{Id}("e"))),\operatorname{Id}("f"))$
- (D) Bin("=",Bin(">",Bin("+",Id("a"),Id("b")),Id("c")),Bin("=",Bin("\*",Id("d"),Id("e"))),Id("f"))
- $(E) \ \operatorname{Bin}("=",\operatorname{Bin}("=",\operatorname{Bin}("+",\operatorname{Id}("a"),\operatorname{Bin}(">",\operatorname{Id}("b"),\operatorname{Id}("c"))),\operatorname{Bin}("*",\operatorname{Id}("d"),\operatorname{Id}("e"))),\operatorname{Id}("f"))$ 
  - 14. Cho giá trị ban đầu của biến **c** là 3, cho biết những giá trị **có thể có** của biến **a** sau khi thực hiện phép gán sau trên ngôn ngữ lập trình C? a = c \* (c = 5);
- (A) 15
- (B) 15, 25, 9
- (C) 9
- (D) 25
- (E) 15, 25

15. Viết lai biểu thức trung tố (infix) sau sang dang hâu tố (postfix) Polish? Giả sử là tất cả các phép toán đều có 2 toán hạng và có độ ưu tiên và tính kết hợp như trong ngôn ngữ lập trình C.

$$a + b * c - d * e + f$$

- (A) a b + c \* d e \* f +
- (B) a + b c \* d e \* f +
- (C) a b c \* + d e \* + f -

- (D) a b c \* + d e f \* +
- (E) a b c \* + d e \* f +

Cho văn phạm sau dùng cho các câu 16–17:

- S BA
- Α a A  $\epsilon$
- Се cB
- $\mathbf{C}$ d C

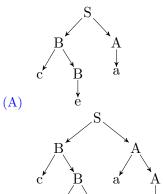
16. Chuỗi nào KHÔNG thuộc ngôn ngữ được mô tả bởi văn phạm trên?

(A) b

(D)

- (B) e
- (C) cccddddeaa
- (D) eaaa
- (E) cccaaa

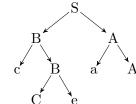
17. Chọn cây phân tích cú pháp (parse tree) cho chuỗi nhập: cea

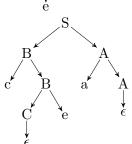


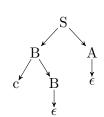
(B)

(E)









. . .



(A)

...

18. Cho biết mã Jasmin của biểu thức viết bằng ngôn ngữ C như sau:

```
a * b - 2
với a và b là 2 biến nguyên có chỉ số lần lượt là 0 và 1.
iload 0
                   iload 0
                                      iload 0
                                                          iload 0
                                                                             imul
                                      iload 1
iload 1
                   iload 1
                                                          imul
                                                                             load 0
iconst 2
                   imul
                                      iconst 2
                                                          iload 1
                                                                             iload 1
                                                    (D)
imul
                   iconst 2
                                      isub
                                                          isub
                                                                             isub
isub
                   isub
                                      imul
                                                          iconst 2
                                                                             iconst 2
```

19. Cho biết kích thước của đối tượng dữ liệu  ${\bf x}$  được khai báo như sau: x: set of 1..16

- (A) 2 bytes (B) 4 bytes (C) 1 byte (D) 4 bits (E) 16 bytes
- 20. Để thực hiện tìm kiếm một chuỗi (String) trên nhiều loại danh sách khác nhau, hãy chọn giải pháp thích hợp nhất để điền vào chỗ trống trong định nghĩa hàm sau:

```
def lookup[T](x:String,lst:List[T],f:T=>String):Option[T] =
   lst match {
     case List() => None
     case h::t =>______
}
```

- (A) if (f(h) == x) Some(h) else lookup(x,t,f) (B) if (h == x) Some(h) else lookup(x,t,f)
- (C) if (f(h) == x) Some(f(h)) else lookup(x,t,f)
- (D) if (h == f(x) Some(h) else lookup(x,t,f)
- (E) if (h == f(x)) Some(f(h)) else lookup(x,t,f)
- 21. Chọn phát biểu ĐÚNG trong các phát biểu về dãy (array) sau:
- (A) Truy xuất một phần tử bên ngoài một dãy C không được phép, sẽ lập tức gây ra lỗi
- (B) Trên C, kích thước của dãy có thể tăng thêm sau khi dãy đã được tạo ra
- (C) Trên Java, kích thước của một dãy KHÔNG được lưu trữ để kiểm tra động
- (D) Trên C và C++, kích thước của một dãy được lưu trữ để kiểm tra động
- (E) Truy xuất một phần tử bên ngoài một dãy Java không được phép, sẽ lập tức gây ra lỗi
- 22. Cơ chế gọi chương trình con nào mà điều khiển có thể chuyển vào một vị trí ở giữa chương trình con được gọi thay vì chuyển vào đầu chương trình con được gọi?
- (A) Gọi trở về đơn giản (Simple Call Return)
- (B) Trình định thời (Scheduled Subprogram)
- (C) Biến cố Xử lý biến cố (Exception)
- (D) Gọi đệ qui (Recursive call)
- (E) Trình cộng hành (Coroutine)

Phần hướng dẫn này áp dụng cho các câu 23–27

Hãy viết thêm các đoạn mã cần thiết vào phương thức dưới đây để thực hiện sinh mã cho phát biểu *repeat*. Nhắc lại phát biểu *repeat* kết thúc quá trình lặp khi điều kiện của phát biểu trở thành **true**. AST của phát biểu **repeat** được định nghĩa như sau: case class Repeat(val stmts:List[Stmt],val exp:Expr) extends Stmt

```
override def visitRepeat(ast: Repeat, o:Context)= {
    val ctxt = o.asInstanceOf[SubBody]
```



```
ctxt.frame.enterLoop();
    val labelStart = ctxt.frame.getNewLabel()
    val labelBreak = ctxt.frame.getBreakLabel()
    val labelCont = ctxt.frame.getContinueLabel()
    val str1 = //1
    // sinh mã cho từng phát biểu trong thân của repeat
    val str2 = //2
    val str3 = //3
    // sinh mã cho biểu thức điều kiện
    val str4 = visit(ast.exp,...)
    val str5 = //4
    val str6 = //5
    frame().exitLoop();
    str1 + str2 + str3 + str4 + str5 + str6
}
```

- 23. Đoạn mã nào trong số các đoạn mã dưới đây cần phải xuất hiện ở vị trí //1 trong đoạn mã trên
- (A) ctxt.emit.emitLABEL(labelStart)
- (B) ""
- (C) ctxt.emit.emitLABEL(labelCont)
- (D) ctxt.emit.emitLABEL(labelBreak)
- (E) ctxt.emit.emitGOTO(labelStart)
- 24. Đoạn mã nào trong số các đoạn mã dưới đây cần phải xuất hiện ở vị trí //2 trong đoạn mã trên
- (A) ast.stmts.map(x = > visit(x, o))
- (B) visit(ast.stmts,o)
- (C) ast.stmts.foldLeft("")((x,y)=>x + visit(y,o))
- (D) ast.stmts.filter(x = > visit(x, o))
- (E) ast.stmts.foldLeft("")((x,y)=>y + visit(x,o))
- 25. Đoạn mã nào trong số các đoạn mã dưới đây cần phải xuất hiện ở vi trí //3 trong đoạn mã trên
- (A) ctxt.emit.emitIFFALSE(labelStart)
- (B) ""
- (C) ctxt.emit.emitLABEL(labelCont)
- (D) ctxt.emit.emitLABEL(labelBreak)
- (E) ctxt.emit.emitGOTO(labelStart)
  - 26. Đoạn mã nào trong số các đoạn mã dưới đây cần phải xuất hiện ở vị trí //4 trong đoạn mã trên
- (A) ctxt.emit.emitIFFALSE(labelStart)
- (B) ctxt.emit.emitGOTO(labelBreak)
- (C) ctxt.emit.emitLABEL(labelCont)
- (D) ctxt.emit.emitLABEL(labelBreak)
- (E) ctxt.emit.emitGOTO(labelStart)
- 27. Đoạn mã nào trong số các đoạn mã dưới đây cần phải xuất hiện ở vị trí //5 trong đoạn
- (A) ctxt.emit.emitIFFALSE(labelStart))
- (B) ctxt.emit.emitGOTO(labelBreak))
- (C) ctxt.emit.emitLABEL(labelCont))
- (D) ctxt.emit.emitLABEL(labelBreak))
- (E) ctxt.emit.emitGOTO(labelStart))
- 28. Cho  $X = \{a,b\}$ . Chọn biểu thức chính quy (regular expression) mô tả ngôn ngữ chứa bất kỳ chuỗi nào được tạo bởi các ký tự trên tập X nhưng KHÔNG chứa chuỗi có 2 ký tự a liên tiếp.
- (A) b\*a?(bb\*a)\*b\* (B) a|(abb\*)\*
- (C)  $b^*(abb^*)^*$
- (D) (b\*ab\*)\*
- (E) b\*ab\*ab\*



Đoạn code sau dùng cho các câu 29–31

```
int p;
int* foo(int x) {
    static int q;
    int *s = new int;
    switch (x) {
        case 1: return &p;
        case 2: return &q;
        case 3: return &x;
        case 4: return s;
        default: return foo(x-1);
    }
}
```

- 29. Phát biểu nào sẽ gây ra lỗi tham chiếu treo (dangling reference) khi thực thi
- (A) return s
- (B) return &p
- (C) return &x
- (D) return &q

- (E) Không phát biểu nào gây ra lỗi trên khi thực thi
- 30. Trong đoạn mã trên, đối tượng dữ liệu nào có thể trở thành rác (garbage)
- (A) trỏ đến bởi s
- (B) p

(C) x

(D) q

- (E) Không có đối tượng nào có thể trở thành rác
- 31. Khi hàm **foo** được gọi đệ quy (recursive) thì các bản hoạt động của **foo** dùng chung những đối tượng dữ liệu nào?
- (A) p, q, x và s
- (B) q và s
- (C) p và q
- (D) p
- (E) x, q và s

32. Cho khai báo sau được viết trên ngôn ngữ lập trình Ada:

Cho kích thước của các kiểu *Integer, Float, Boolean và Enumeration* lần lượt là 2, 4, 1, và 2, và kích thước cho phần mô tả kiểu (type description) là 0. Cho biết kích thước của một đối tượng kiểu **Figure**?

- (A) 15
- (B) 21
- (C) 23
- (D) 13
- (E) Khác

Đoạn mã sau được dùng cho các câu 33–38.

Cho một đoạn chương trình được viết trên một ngôn ngữ tựa C như sau:



```
int A[5] = {1,3,5,7,9}; // index of A starts from 0
int j = 0;
int n = 5;
int sumAndIncrease(int a, int i) {
    int s = 0;
    for ( ; i < n; i = i + 1) {
        s = s + a;
        A[j] = A[j] + 1;
    }
    return s;
}

void main(){
    int s = sumAndIncrease(A[j],j);
    printf("a = %i\n",s); //1
    printf("%i %i %i %i %i\n",A[0],A[1],A[2],A[3],A[4]); //2
}</pre>
```

- 33. Nếu a và i được truyền bằng **trị-kết quả (passed by value-result)**, giá trị của biến s được in ra ở phát biểu //1 là bao nhiêu?
- (A) 25
- (B) 5
- (C) 15
- (D) 10
- (E) Khác
- 34. Nếu *a* và *i* được truyền bằng **trị-kết quả (passed by value-result)**, giá trị của biến *A* được in ra ở phát biểu //2 là bao nhiêu?
- (A) 5 3 5 7 9
- (B) 1 3 5 7 9
- (C) 6 3 5 7 9
- (D) 2 4 6 8 10
- (E) Khác
- 35. Nếu a và i được truyền bằng **tham khảo (passed by reference)**, giá trị của biến s được in ra ở phát biểu //1 là bao nhiêu?
- (A) 25
- (B) 5
- (C) 15
- (D) 10
- (E) Khác
- 36. Nếu a và i được truyền bằng **tham khảo (passed by reference)**, giá trị của biến A được in ra ở phát biểu //2 là bao nhiêu?
- (A) 5 3 5 7 9
- (B) 1 3 5 7 9
- (C) 6 3 5 7 9
- (D) 2 4 6 8 10
- (E) Khác
- 37. Nếu a và i được truyền bằng **tên (passed by name)**, giá trị của biến s được in ra ở phát biểu //1 là bao nhiêu?
- (A) 25
- (B) 5
- (C) 20
- (D) 15
- (E) Khác
- 38. Nếu a và i được truyền bằng **tên (passed by name)**, giá trị của biến A được in ra ở phát biểu //2 là bao nhiêu?
- (A) 5 3 5 7 9
- (B) 1 3 5 7 9
- (C) 6 3 5 7 9
- (D) 2 4 6 8 10
- (E) Khác
- 39. Cho lớp(class) A là lớp cha (superclass) của lớp B và B là lớp cha của hai lớp C và D. Trong lớp A, có khai báo phương thức thực thể (instance method) **foo** và phương thức này bị ghi đè(override) trên tất cả các lớp B, C và D. Cho biến x được khai báo như sau: **B x**;

Qui ước viết  $func_T$  là phương thức func được khai báo trong lớp T, cho biết tập của phương thức đích của cuộc gọi sau:  $\mathbf{x.foo}$ ();

- (A)  $\{foo_A, foo_B, foo_C, foo_D\}$
- (B)  $\{foo_C, foo_D\}$
- (C)  $\{foo_A, foo_B\}$

(D)  $\{foo_B\}$ 

(E)  $\{foo_B, foo_C, foo_D\}$ 



- 40. Cơ chế gọi chương trình nào là cơ chế gọi chương trình con cơ bản của Lập trình hướng sự kiện (Event-driven Programming)
- (A) Gọi trở về đơn giản (Simple Call Return)
- (B) Biến cố Xử lý biến cố (Exception)
- (C) Trình cộng hành (Coroutine)
- (D) Gọi đệ qui (Recursive call)
- (E) Trình định thời (Scheduled Subprogram)
  - 41. Một biến trên ngôn ngữ Javascript có thể nhận những giá trị thuộc các kiểu khác nhau. Ví du:

```
var x = "def";
```

x = 1;

Ràng buộc giữa một biến và kiểu của nó trên ngôn ngữ Javascript xảy ra trong thời gian

(A) Chạy (running)

(B) Hiện thực (implementation)

(C) Dich (compiling)

(D) Lập trình (programming)

- (E) Khởi động (loading)
  - 42. Gọi append là hàm dùng để nối 2 danh sách thành 1 danh sách. Ví dụ append([1,2,3],[4,5,6]) sẽ có kết quả là [1,2,3,4,5,6]. Hãy hiện thực hàm append(a:List[Int],b:List[Int]) dùng hàm bậc cao (high-order function)?
- (A) a.foldLeft(b)((x,y)=>x::y)

(B) a.map(b)((x,y)=>x::y)

(C) b.foldLeft(a)((x,y)=>x::y)

(D) a.foldRight(b)((x,y)=>x::y)

(E) b.foldRight(a)((x,y)=>x::y)

Phần trình bày sau dùng trong các câu hỏi 43–44:

Cho mã giả của phát biểu for var = expr1 to expr2 do body như sau:

start: var = expr1
 etemp = expr2

loop: if (var >= etemp) goto out

body

label1: var++
label2: goto loop

out:

- 43. Nếu trong thân *body* có thực thi lệnh *continue* thì điều khiển sẽ chuyển đến thực thi phát biểu có nhãn là:
- (A) start
- (B) out
- (C) label1
- (D) loop
- (E) label2
- 44. Với hiện thực của phát biểu *for* như trên, phát biểu nào đặt trong thân phát biểu **for** dưới đây sẽ làm cho phát biểu này bi lặp mãi mãi?

(A) i = i - 1;

(B) s = s - 1;

(C) n = n + 1;

- (D) câu A và C đúng
- (E) Không phát biểu nào trong các câu A, B và C làm phát biểu for lặp mãi mãi



45. Khi cần khai thác đặc tính của kiểu union(các thành phần loại trừ lẫn nhau), làm thế nào để hiện thực trên những ngôn ngữ lập trình hướng đối tượng như Java(không có kiểu union)? Cụ thể làm thế nào để hiện thực biến z kiểu union sau trên những ngôn ngữ này?

```
union {
    string loikhen;
    string loiche;
} z;
```

- (A) Định nghĩa một lớp A có 2 thuộc tính loikhen và loiche; và z có kiểu A
- (B)  $\frac{\rm Dịnh}$  nghĩa lớp cha A và hai lớp con B (có thuộc tính lo<br/>ikhen) và C (có thuộc tính lo<br/>iche), và z có kiểu A
- (C) Định nghĩa lớp A có thuộc tính loikhen và lớp B(con của A) có thuộc tính loiche, và z có kiểu B
- (D) Định nghĩa lớp A có thuộc tính lo<br/>ikhen và lớp B(con của A) có thuộc tính loiche, và z có kiểu A
- (E) Định nghĩa lớp trừu tượng (abstract class) A và 2 lớp con B (có thuộc tính loikhen) và C (có thuộc tính loiche), và z có kiểu A
- 46. Đoạn code viết bằng Scala nào dưới đây có thể dùng như một ví dụ minh họa cho khái niệm Hàm Currying
- (A) List(1,2,3).foldLeft(0)(+)
- (B) def add1(n:Int) = n + 1
- (C) def add(n:Int)(x:Int) = n + x; val add2 = add (2)
- (D)  $\operatorname{def} \operatorname{add}(\operatorname{n:Int}) = (\operatorname{x:Int}) = n + \operatorname{x;val} \operatorname{add}2 = \operatorname{add}(2)$
- (E) List(1,2,3).filter( > 1)
  - 47. Kiểu *union* của ngôn ngữ C++ cho phép ghi vào ở thành phần này nhưng đọc lên ở thành phần khác như trong ví dụ sau:

```
union {
    int songuyen;
    char kytu[2];
} x;
x.songuyen = 12;
cout << x.kytu[0] << " " << x.kytu[1] << endl;</pre>
```

Đặc tính này của ngôn ngữ C++ gây ảnh hưởng XẤU NHẤT đối với

(A) tính đơn giản (simplicity)

(B) tính trực giao (orthonogality)

(C) tính tin cậy (reliability)

(D) tính dễ viết (writability)

- (E) chi phí (cost)
- II. Phần câu hỏi tự luận: (dành cho tất cả sinh viên) (2 điểm)
  - 48. Hãy viết lại biểu thức ở dạng trung tố (infix) sau sang dạng biểu thức tiền tố (prefix) Cambridge Polish:

```
a + b * c * d - e - f
```

Biết rằng độ ưu tiên và tính kết hợp của các phép toán trong biểu thức như thông thường (đều kết hợp trái và \* có ưu tiên cao hơn +, -). Trong biểu thức tiền tố nhiều toán hạng, thứ tự tính toán cũng từ trái sang phải.

Yêu cầu: Biểu thức dạng tiền tố Cambridge Polish phải thỏa các yêu cầu sau:



- Thứ tự xuất hiện các toán hạng trong biểu thức dạng tiềm tố phải có cùng thứ tự xuất hiện như trong biểu thức trung tố
- Số ( và ) là ít nhất
- Có cùng thứ tự tính toán các phép toán với thứ tự đó trong biểu thức dạng trung tố

49. Hãy nêu những giải pháp đã được sử dụng để tránh tham chiếu treo (dangling reference) trong trường hợp trả về hàm như sau:

```
void->int F() {
    int x = 1;
    int g() {
        return x + 1;
    }
    return g;
}
void->int gg = F();
int z = gg();
```

50. Trên những ngôn ngữ lập trình cho phép rút ngắn tính toán (short-circuit evaluation) biểu thức luận lý, biểu thức sau sẽ không gây ra lỗi khi thực thi.

$$((a != 0) \&\& ((b / a) > 5))$$

Hãy dùng **if then else** để mô phỏng quá trình tính toán của biểu thức luận lý trên?

III. Phần bài tập lớn: (dành cho tất cả sinh viên)



- 51. Viết hàm visitBlock(ast:Block,o:Context) để thực hiện kiểm tra kiểu. Hàm visitBlock sẽ thực hiện chức năng xây dựng bảng danh hiệu (symbol table) và truyền bảng này khi visit các phát biểu trong khối. Yêu cầu giải thích rõ về Context cụ thể được sử dụng trong hàm và ý tưởng mã của hàm. Nhắc lại AST của Block được khai báo như sau: case class Block(val decl:List[Decl],val stmts:List[Stmt]) extends Stmt
- 52. Viết hàm visitIf(ast:If,o:Context) để thực hiện sinh mã cho một phát biểu If. Nhắc lại AST của If được khai báo như sau: case class If(val expr:Expr, val thenStmt:Stmt, val elseStmt:Option[Stmt]) extends Stmt Yêu cầu giải thích rõ về Context cụ thể được sử dụng trong hàm và ý tưởng mã của hàm.

#### IV. Phần KSTN: (dành cho sinh viên KSTN)

- 53. Viết hàm visitIf(ast:If,o:Context) để thực hiện kiểm tra lỗi FunctionNotReTurn? Yêu cầu giải thích ý tưởng thực hiện thể hiện qua mã của hàm?





Chủ nhiệm bộ môn	Giảng viên ra đề
Chữ kí:	Chữ kí:
Họ tên:	Họ tên:



#### KHOA **KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH** BÔ MÔN **KHOA HOC MÁY TÍNH**

# ĐÁP ÁN cho Đề thi cuối kỳ

Môn thi	Nguyên	Lý N	Vgôn	Ngữ	Lập	Trình
---------	--------	------	------	-----	-----	-------

Thời gian: 120 phút

 $\square$  Sinh viên được phép sử dụng tài liệu

⊠ Sinh viên không được sử dụng tài liệu

Mã đề: 1155

#### I. Phần câu hỏi trắc nghiệm:(8 điểm))

1. Cho biết mã Jasmin của phát biểu gán sau:  $\mathbf{a}[\mathbf{i}] = \mathbf{a}[\mathbf{i}] + 2$ 

với i kiểu nguyên có chỉ số là 3 và a là kiểu dãy nguyên có chỉ số là 1.

```
aload 1
                                                              aload 1
                                                                                 aload 1
                        iastore
     aload 1
                                                              iload 3
                                           dup
                        iadd
                                                                                 iload 3
     iload 3
                                           iload 3
                                                              aload 1
                        iaload
                                                                                 iastore
     iaload
                                           dup
                                                              iload 3
                        aload 1
                                                                                 aload 1
(A) iconst 2
                                           iaload
                                                              iaload
                                                                            (E)
                        iload 3
                                                                                 iload 3
     iadd
                                           iconst 2
                                                              iconst 2
                        dup
                                                                                 iconst 2
     iastore
                                           iadd
                                                              iadd
                                                                                 iadd
                        iconst 2
                                           iastore
                                                              iastore
                                                                                  ...
                                           ...
                                                               ...
```

Đoạn code sau, được viết trên ngôn ngữ dùng **qui tắc tầm vực tĩnh (static-scope rule)**, áp dụng cho các câu 2–5

- 2. Môi trường tham khảo tĩnh (static referencing environment) của thủ tục **sub3** KHÔNG chứa
- (A) sub2
- B a ở //1
- (C) sub1
- (D) a  $\dot{\sigma} / / 2$
- (E) sub3



3.	Giả sử chuỗi gọi là main $\rightarrow$	$\mathrm{sub1} \to$	sub2 -	$\rightarrow$ sub3,	tham	khảo	đến	a trong	sub3	ứng vớ	ji
	khai báo										

- $\mathbf{A}$  a trong sub1 (//2)
- (B) Báo lỗi a chưa khai báo
- (C) a trong sub3

- (D) a trong sub2 (//3)
- (E) a trong main (//1)
- 4. Giả sử chuỗi gọi là main  $\to$  sub1  $\to$  sub2  $\to$  sub3 , tham khảo đến  $\bf a$  trong sub3  $\bf khi$  thực thi ứng với kết hợp của  $\bf a$  trong bản hoạt động của
- A sub1
- (B) main
- (C) sub3
- (D) sub2

- (E) Báo lỗi không tìm thấy a
  - 5. Nếu đoạn code trên được viết trên ngôn ngữ dùng qui tắc tầm vực động (dynamic-scope rule), và giả sử chuỗi gọi là main  $\rightarrow$  sub1  $\rightarrow$  sub2  $\rightarrow$  sub3, tham khảo đến a trong sub3 **khi thực thi** ứng với kết hợp của a trong bản hoạt động của
- (A) main
- (B) Báo lỗi không tìm thấy a
- C sub2
- (D) sub1

- (E) sub3
  - 6. Trình đinh thời (scheduled subprograms) thường được dùng trong
- (A) Lập trình hướng đối tượng (Object-Oriented Programming)
- (B) Lâp trình song song (Parallel Programming)
- (C) Lập trình hướng sự kiện (Event-driven Programming)
- (D) Lập trình hàm (Functional Programming)
- E Lập trình thời gian thực (Real-time Programming)
  - 7. Liệt kê theo thứ tự chiều dài tăng dần ít nhất 5 chuỗi ngắn nhất (nếu có nhiều hơn 5 chuỗi) của ngôn ngữ được mô tả bởi biểu thức chính quy (regular expression) sau: ab\*(a|b)? a
- (A)  $\epsilon$ , a, aa, aba, abaa
- (B) aa, aba, aaa, abaa, aaba
- (C) a, aa, aba, aaa, aaba

- D aa, aba, aaa, abaa, abba
- (E) aa, aba, aaa, abaa, aaaa
- 8. Giả sử chương trình có một lỗi văn phạm nằm sau lệnh **print**. Khi em nhấn nút Run (để dịch và chạy chương trình) trên trình soạn thảo, kết quả được in ra bởi lệnh **print** trên trước khi lỗi văn phạm được báo. Hỏi chế độ dịch của trình soạn thảo là gì?
- (A) trình biên dich (compiler)

(B) trình liên kết (link editor)

(C) trình hợp ngữ (assembler)

- D trình thông dich(interpreter)
- (E) trình biên dịch động(just-in-time compiler)

Đoạn mã sau được dùng trong các câu 9–10:

```
int *p = new int;
void foo(int * r) {
    delete r;
}
foo(p); //1
*p = 2; //2
```

- 9. Hiện tượng gì xảy ra khi p được truyền cho r ở phát biểu //1 trong đoạn mã trên:
- A Bí danh (alias)
- (B) Con trỏ chưa khai báo (undeclared pointer) (C) Tham chiếu treo (dangling reference)
- (D) Khai báo trùng tên (redeclared)
- (E) Rác (garbage)



10. Hiện tượng gì xảy ra khi thực thi phát biểu //2 trong đoạn mã trên:

(A) Bí danh

- B Tham chiếu treo
- (C) Rác

- (D) Da hình (polymorphism)
- (E) Con trỏ chưa khai báo

Cho văn phạm của một biểu thức sau dùng cho các câu 11–13:

```
\exp \rightarrow \text{term '='} \exp \mid \text{term}
```

$$term \rightarrow term '+' fact | term '>' fact | fact$$

fact 
$$\rightarrow$$
 ope '\*' ope | ope

ope 
$$\rightarrow$$
 '(' exp ')' | ID

với ID là một danh hiệu.

11. Chọn biểu thức được viết đúng văn phạm, có số lượng dấu '(' và ')' ít nhất mà tương đương với biểu thức sau: (a + (b = ((c \* d) \* e))) > f

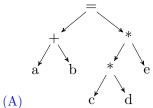
- (A) (a + (b = ((c \* d) \* e))) > f
- (B) (a + (b = (c \* d) \* e)) > f

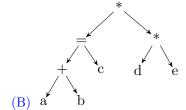
 $C \mid a + (b = (c * d) * e) > f$ 

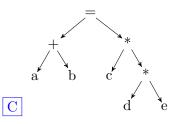
(D) a + b = c \* d \* e > f

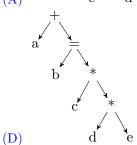
(E) a + (b = c \* d \* e) > f

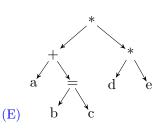
12. Vẽ AST cho biểu thức sau được viết dựa vào văn phạm trên:  $\mathbf{a} + \mathbf{b} = \mathbf{c} * (\mathbf{d} * \mathbf{e})$ 











13. Cho cấu trúc AST được khai báo trên Scala như sau: trait Exp case class Bin(op:String,e1:Exp,e2:Exp) extends Exp case class Id(i:String) extends Exp

Chọn AST cho biểu thức sau:  $\mathbf{a} + \mathbf{b} > \mathbf{c} = \mathbf{d} * \mathbf{e} = \mathbf{f}$ 

- (A) Bin("=",Bin("=",Bin(">",Bin("+",Id("a"),Id("b")),Id("c")),Bin("\*",Id("d"),Id("e"))),Id("f"))
- (B) Bin("=",Bin(">",Bin("+",Id("a"),Id("b")),Id("c")),Bin("\*",Id("d"),Bin("=",Id("e"),Id("f"))))
- (C) Bin("=",Bin("+",Id("a"),Bin(">",Id("b"),Id("c"))),Bin("=",Bin("\*",Id("d"),Id("e"))),Id("f"))
- D Bin("=",Bin(">",Bin("+",Id("a"),Id("b")),Id("c")),Bin("=",Bin("\*",Id("d"),Id("e"))),Id("f"))
- $\overline{\text{(E)}} \ \operatorname{Bin}("=",\operatorname{Bin}("=",\operatorname{Bin}("+",\operatorname{Id}("a"),\operatorname{Bin}(">",\operatorname{Id}("b"),\operatorname{Id}("c"))),\operatorname{Bin}("*",\operatorname{Id}("d"),\operatorname{Id}("e"))),\operatorname{Id}("f"))$



- 14. Cho giá trị ban đầu của biến c là 3, cho biết những giá trị có thể có của biến a sau khi thực hiện phép gán sau trên ngôn ngữ lập trình C? a = c \* (c = 5);
- (A) 15
- (B) 15, 25, 9
- (C) 9
- (D) 25
- E 15, 25
- 15. Viết lại biểu thức trung tố (infix) sau sang dạng hậu tố (postfix) Polish? Giả sử là tất cả các phép toán đều có 2 toán hạng và có độ ưu tiên và tính kết hợp như trong ngôn ngữ lâp trình C.
  - a + b \* c d \* e + f
- (A) a b + c \* d e \* f +
- (B) a + b c \* d e \* f +
- (C) a b c \* + d e \* + f -

- (D) a b c \* + d e f \* +
- E a b c \* + d e \* f +

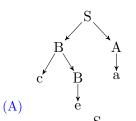
Cho văn phạm sau dùng cho các câu 16–17: ВА

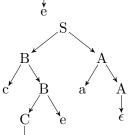
- S
- Α a A
- В Се  $c\ B$
- d C  $\epsilon$
- 16. Chuỗi nào KHÔNG thuộc ngôn ngữ được mô tả bởi văn pham trên?
- (A) b

D

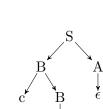
(A)

- (B) e
- (C) cccddddeaa
- (D) eaaa
- E cccaaa
- 17. Chọn cây phân tích cú pháp (parse tree) cho chuỗi nhập: cea





(B)



(E)

(C)

18. Cho biết mã Jasmin của biểu thức viết bằng ngôn ngữ C như sau:

với a và b là 2 biến nguyên có chỉ số lần lượt là 0 và 1.

- iload 0 iload 1
- iconst 2
- В
- iload 0 iload 1

iconst 2

imul

isub

- iload 0 iload 1

iconst 2

- (D) isub
- iload 0 imul iload 1
- (E)
- iconst 2
- iload 1 isub iconst 2

imul

load 0

isub

imul

- imul ...

Trang 4/12

isub



- 19. Cho biết kích thước của đối tượng dữ liệu  ${\bf x}$  được khai báo như sau:  ${\bf x}$ : set of 1..16
- A 2 bytes (B) 4 bytes
- (C) 1 byte
- (D) 4 bits

- (E) 16 bytes
- 20. Để thực hiện tìm kiếm một chuỗi (String) trên nhiều loại danh sách khác nhau, hãy chọn giải pháp thích hợp nhất để điền vào chỗ trống trong định nghĩa hàm sau:

```
def lookup[T](x:String,lst:List[T],f:T=>String):Option[T] =
   lst match {
     case List() => None
     case h::t =>______
}
```

- A if (f(h) == x) Some(h) else lookup(x,t,f)
- (B) if (h == x) Some(h) else lookup(x,t,f)
- (C) if (f(h) == x) Some(f(h)) else lookup(x,t,f)
- (D) if (h == f(x) Some(h) else lookup(x,t,f)
- (E) if (h == f(x)) Some(f(h)) else lookup(x,t,f)
- 21. Chọn phát biểu ĐÚNG trong các phát biểu về dãy (array) sau:
- (A) Truy xuất một phần tử bên ngoài một dãy C không được phép, sẽ lập tức gây ra lỗi
- (B) Trên C, kích thước của dãy có thể tăng thêm sau khi dãy đã được tạo ra
- (C) Trên Java, kích thước của một dãy KHÔNG được lưu trữ để kiểm tra động
- (D) Trên C và C++, kích thước của một dãy được lưu trữ để kiểm tra động
- E Truy xuất một phần tử bên ngoài một dãy Java không được phép, sẽ lập tức gây ra lỗi
- 22. Cơ chế gọi chương trình con nào mà điều khiển có thể chuyển vào một vị trí ở giữa chương trình con được gọi thay vì chuyển vào đầu chương trình con được gọi?
- (A) Gọi trở về đơn giản (Simple Call Return)
- (B) Trình định thời (Scheduled Subprogram)
- (C) Biến cố Xử lý biến cố (Exception)
- (D) Gọi đệ qui (Recursive call)
- E Trình cộng hành (Coroutine)

Phần hướng dẫn này áp dụng cho các câu 23–27

Hãy viết thêm các đoạn mã cần thiết vào phương thức dưới đây để thực hiện sinh mã cho phát biểu *repeat*. Nhắc lại phát biểu *repeat* kết thúc quá trình lặp khi điều kiện của phát biểu trở thành **true**. AST của phát biểu **repeat** được định nghĩa như sau: case class Repeat(val stmts:List[Stmt],val exp:Expr) extends Stmt

```
override def visitRepeat(ast: Repeat, o:Context)= {
    val ctxt = o.asInstanceOf[SubBody]
    ctxt.frame.enterLoop();
    val labelStart = ctxt.frame.getNewLabel()
    val labelBreak = ctxt.frame.getBreakLabel()
    val labelCont = ctxt.frame.getContinueLabel()
    val str1 = //1
    // sinh mã cho từng phát biểu trong thân của repeat
```



```
val str2 = //2
val str3 = //3
// sinh mã cho biểu thức điều kiện
val str4 = visit(ast.exp,...)
val str5 = //4
val str6 = //5
frame().exitLoop();
str1 + str2 + str3 + str4 + str5 + str6
}
```

- 23. Đoạn mã nào trong số các đoạn mã dưới đây cần phải xuất hiện ở vị trí //1 trong đoạn mã trên
- A ctxt.emit.emitLABEL(labelStart)
- (B) ""
- (C) ctxt.emit.emitLABEL(labelCont)
- (D) ctxt.emit.emitLABEL(labelBreak)
- (E) ctxt.emit.emitGOTO(labelStart)
  - 24. Đoạn mã nào trong số các đoạn mã dưới đây cần phải xuất hiện ở vị trí //2 trong đoạn mã trên
- (A) ast.stmts.map(x = > visit(x, o))
- (B) visit(ast.stmts,o)
- |C| ast.stmts.foldLeft("")((x,y)=>x + visit(y,o))
- $\overline{(D)}$  ast.stmts.filter(x=>visit(x,o))
- (E) ast.stmts.foldLeft("")((x,y)=>y + visit(x,o))
- 25. Đoạn mã nào trong số các đoạn mã dưới đây cần phải xuất hiện ở vị trí //3 trong đoạn mã trên
- (A) ctxt.emit.emitIFFALSE(labelStart)
- (B) ""
- C ctxt.emit.emitLABEL(labelCont)
- (D) ctxt.emit.emitLABEL(labelBreak)
- (E) ctxt.emit.emitGOTO(labelStart)
- 26. Đoạn mã nào trong số các đoạn mã dưới đây cần phải xuất hiện ở vị trí //4 trong đoạn mã trên
- A ctxt.emit.emitIFFALSE(labelStart)
- (B) ctxt.emit.emitGOTO(labelBreak)
- (C) ctxt.emit.emitLABEL(labelCont)
- (D) ctxt.emit.emitLABEL(labelBreak)
- (E) ctxt.emit.emitGOTO(labelStart)
  - 27. Đoạn mã nào trong số các đoạn mã dưới đây cần phải xuất hiện ở vị trí //5 trong đoạn mã trên
- (A) ctxt.emit.emitIFFALSE(labelStart))
- (B) ctxt.emit.emitGOTO(labelBreak))
- (C) ctxt.emit.emitLABEL(labelCont))
- D ctxt.emit.emitLABEL(labelBreak))
- (E) ctxt.emit.emitGOTO(labelStart))
- E) coxo.emic.emicoo i o(iabeibiai o))
- 28. Cho  $X = \{a,b\}$ . Chọn biểu thức chính quy (regular expression) mô tả ngôn ngữ chứa bất kỳ chuỗi nào được tạo bởi các ký tự trên tập X nhưng  $KH\^ONG$  chứa chuỗi có 2 ký tự a liên tiếp.
- A b\*a?(bb\*a)\*b\*

- (B)  $a|(abb^*)^*$
- (C) b\*(abb\*)\*

- (D) (b\*ab\*)\*
- (E) b\*ab\*ab\*



Đoạn code sau dùng cho các câu 29-31

```
int p;
int* foo(int x) {
    static int q;
    int *s = new int;
    switch (x) {
        case 1: return &p;
        case 2: return &q;
        case 3: return &x;
        case 4: return s;
        default: return foo(x-1);
    }
}
```

- 29. Phát biểu nào sẽ gây ra lỗi tham chiếu treo (dangling reference) khi thực thi
- (A) return s
- (B) return &p
- C return &x
- (D) return &q

- (E) Không phát biểu nào gây ra lỗi trên khi thực thi
- 30. Trong đoạn mã trên, đối tượng dữ liệu nào có thể trở thành rác (garbage)
- A trỏ đến bởi s
- (B) p

(C) x

(D) q

- (E) Không có đối tượng nào có thể trở thành rác
- 31. Khi hàm **foo** được gọi đệ quy (recursive) thì các bản hoạt động của **foo** dùng chung những đối tượng dữ liệu nào?
- (A) p, q, x và s
- (B) q và s
- C p và q
- (D) p

- (E) x, q và s
  - 32. Cho khai báo sau được viết trên ngôn ngữ lập trình Ada:

Cho kích thước của các kiểu *Integer, Float, Boolean và Enumeration* lần lượt là 2, 4, 1, và 2, và kích thước cho phần mô tả kiểu (type description) là 0. Cho biết kích thước của một đối tượng kiểu **Figure**?

- (A) 15
- (B) 21
- (C) 23
- D 13
- (E) Khác

Đoạn mã sau được dùng cho các câu 33–38.

Cho một đoạn chương trình được viết trên một ngôn ngữ tựa C như sau:



```
int A[5] = {1,3,5,7,9}; // index of A starts from 0
int j = 0;
int n = 5;
int sumAndIncrease(int a, int i) {
    int s = 0;
    for ( ; i < n; i = i + 1) {
        s = s + a;
        A[j] = A[j] + 1;
    }
    return s;
}

void main(){
    int s = sumAndIncrease(A[j],j);
    printf("a = %i\n",s); //1
    printf("%i %i %i %i %i\n",A[0],A[1],A[2],A[3],A[4]); //2
}</pre>
```

- 33. Nếu *a* và *i* được truyền bằng **trị-kết quả (passed by value-result)**, giá trị của biến *s* được in ra ở phát biểu //1 là bao nhiêu?
- (A) 25 B 5 (C) 15 (D) 10 (E) Khác
  - 34. Nếu a và i được truyền bằng **trị-kết quả (passed by value-result)**, giá trị của biến A được in ra ở phát biểu //2 là bao nhiêu?
- (A) 5 3 5 7 9 (B) 1 3 5 7 9 (C) 6 3 5 7 9 (D) 2 4 6 8 10 (E) Khác
  - 35. Nếu a và i được truyền bằng **tham khảo (passed by reference)**, giá trị của biến s được in ra ở phát biểu //1 là bao nhiêu?
- (A) 25 (B) 5 (C) 15 (D) 10
- 36. Nếu a và i được truyền bằng **tham khảo (passed by reference)**, giá trị của biến A được in ra ở phát biểu //2 là bao nhiêu?
- 37. Nếu a và i được truyền bằng **tên (passed by name)**, giá trị của biến s được in ra ở phát biểu //1 là bao nhiêu?
- A 25 (B) 5 (C) 20 (D) 15 (E) Khác
- 38. Nếu a và i được truyền bằng **tên (passed by name)**, giá trị của biến A được in ra ở phát biểu //2 là bao nhiêu?
- (A) 5 3 5 7 9 (B) 1 3 5 7 9 (C) 6 3 5 7 9 D 2 4 6 8 10 (E) Khác



39. Cho lớp(class) A là lớp cha (superclass) của lớp B và B là lớp cha của hai lớp C và D. Trong lớp A, có khai báo phương thức thực thể (instance method) foo và phương thức này bị ghi đè(override) trên tất cả các lớp B, C và D. Cho biến x được khai báo như sau: B x;

Qui ước viết func<sub>T</sub> là phương thức func được khai báo trong lớp T, cho biết tập của phương thức đích của cuộc gọi sau: **x.foo()**;

- $\begin{array}{cccc} \text{(A)} & \{foo_A, foo_B, foo_C, foo_D\} \\ \text{(D)} & \{foo_B\} \end{array} \qquad \begin{array}{cccc} \text{(B)} & \{foo_C, foo_D\} \\ \text{E} & \{foo_B, foo_C, foo_D\} \end{array}$
- (C)  $\{foo_A, foo_B\}$

- 40. Cơ chế gọi chương trình nào là cơ chế gọi chương trình con cơ bản của Lập trình hướng sự kiện (Event-driven Programming)
- (A) Gọi trở về đơn giản (Simple Call Return)
- B Biến cố Xử lý biến cố (Exception)
- (C) Trình công hành (Coroutine)
- (D) Goi đê qui (Recursive call)
- (E) Trình định thời (Scheduled Subprogram)
- 41. Một biến trên ngôn ngữ Javascript có thể nhận những giá trị thuộc các kiểu khác nhau. Ví du:

```
var x = "def";
```

x = 1:

Ràng buộc giữa một biến và kiểu của nó trên ngôn ngữ Javascript xảy ra trong thời gian

A Chay (running)

(B) Hiện thực (implementation)

(C) Dich (compiling)

(D) Lập trình (programming)

- (E) Khởi động (loading)
- 42. Gọi append là hàm dùng để nối 2 danh sách thành 1 danh sách. Ví dụ append ([1,2,3],[4,5,6])sẽ có kết quả là [1,2,3,4,5,6]. Hãy hiện thực hàm append(a:List[Int],b:List[Int]) dùng hàm bậc cao (high-order function)?
- (A) a.foldLeft(b)((x,y)=>x::y)

(B) a.map(b)((x,y)=>x::y)

(C) b.foldLeft(a)((x,y)=>x::y)

D a.foldRight(b)((x,y)=>x::y)

(E) b.foldRight(a)((x,y)=>x::y)

Phần trình bày sau dùng trong các câu hỏi 43–44:

Cho mã giả của phát biểu for var = expr1 to expr2 do body như sau:

start: var = expr1

etemp = expr2

loop: if (var >= etemp) goto out

label1: var++

label2: goto loop

out:

- 43. Nếu trong thân body có thực thi lênh continue thì điều khiển sẽ chuyển đến thực thi phát biểu có nhãn là:
- (A) start
- (B) out
- C label1
- (D) loop
- (E) label2



44. Với hiện thực của phát biểu *for* như trên, phát biểu nào đặt trong thân phát biểu **for** dưới đây sẽ làm cho phát biểu này bị lặp mãi mãi?

```
n = 10;s = 0;
for i = s to n do ______
```

- $\boxed{A} \quad i = i 1;$
- (B) s = s 1;

(C) n = n + 1;

- (D) câu A và C đúng
- (E) Không phát biểu nào trong các câu A, B và C làm phát biểu for lặp mãi mãi
  - 45. Khi cần khai thác đặc tính của kiểu union(các thành phần loại trừ lẫn nhau), làm thế nào để hiện thực trên những ngôn ngữ lập trình hướng đối tượng như Java(không có kiểu union)? Cụ thể làm thế nào để hiện thực biến z kiểu union sau trên những ngôn ngữ này?

```
union {
    string loikhen;
    string loiche;
} z;
```

- (A) Định nghĩa một lớp A có 2 thuộc tính loikhen và loiche; và z có kiểu A
- (B) Định nghĩa lớp cha A và hai lớp con B (có thuộc tính lo<br/>ikhen) và C (có thuộc tính lo<br/>iche), và z có kiểu A
- (C) Định nghĩa lớp A có thuộc tính lo<br/>ikhen và lớp B(con của A) có thuộc tính loiche, và z có kiểu B
- (D) Định nghĩa lớp A có thuộc tính lo<br/>ikhen và lớp B(con của A) có thuộc tính loiche, và z có kiểu A
- Định nghĩa lớp trừu tượng (abstract class) A và 2 lớp con B (có thuộc tính loikhen) và C (có thuộc tính loiche), và z có kiểu A
- 46. Đoạn code viết bằng Scala nào dưới đây có thể dùng như một ví dụ minh họa cho khái niêm Hàm Currying
- (A) List(1,2,3).foldLeft $(0)(_ + _ )$
- (B) def add1(n:Int) = n + 1
- C def add(n:Int)(x:Int) = n + x; val add2 = add (2)
- (D)  $\operatorname{def} \operatorname{add}(\operatorname{n:Int}) = (\operatorname{x:Int}) => \operatorname{n} + \operatorname{x;val} \operatorname{add} 2 = \operatorname{add}(2)$
- (E) List(1,2,3).filter( > 1)



47. Kiểu *union* của ngôn ngữ C++ cho phép ghi vào ở thành phần này nhưng đọc lên ở thành phần khác như trong ví dụ sau:

```
union {
    int songuyen;
    char kytu[2];
} x;
x.songuyen = 12;
cout << x.kytu[0] << " " << x.kytu[1] << endl;</pre>
```

Đặc tính này của ngôn ngữ C++ gây ảnh hưởng  $X \hat{A} U NH \hat{A} T$  đối với

(A) tính đơn giản (simplicity)

(B) tính trực giao (orthonogality)

tính tin cây (reliability)

(D) tính dễ viết (writability)

(E) chi phí (cost)

### II. Phần câu hỏi tự luận: (dành cho tất cả sinh viên) (2 điểm)

48. Hãy viết lại biểu thức ở dạng trung tố (infix) sau sang dạng biểu thức tiền tố (prefix) Cambridge Polish:

```
a + b * c * d - e - f
```

Biết rằng độ ưu tiên và tính kết hợp của các phép toán trong biểu thức như thông thường (đều kết hợp trái và \* có ưu tiên cao hơn +, -). Trong biểu thức tiền tố nhiều toán hạng, thứ tự tính toán cũng từ trái sang phải.

Yêu cầu: Biểu thức dạng tiền tố Cambridge Polish phải thỏa các yêu cầu sau:

- Thứ tự xuất hiện các toán hạng trong biểu thức dạng tiềm tố phải có cùng thứ tự xuất hiện như trong biểu thức trung tố
- Số ( và ) là ít nhất
- Có cùng thứ tự tính toán các phép toán với thứ tự đó trong biểu thức dạng trung tố

\_\_\_\_\_

```
Lời giải. (+ a (- (* b c d) e f))
Nếu viết đúng biểu thức dạng tiền tố Cambridge Polish: 1
Nếu viết đúng và có số () ít nhất: 1
```

49. Hãy nêu những giải pháp đã được sử dụng để tránh tham chiếu treo (dangling reference) trong trường hợp trả về hàm như sau:

```
void->int F() {
    int x = 1;
    int g() {
        return x + 1;
    }
    return g;
}
void->int gg = F();
int z = gg();
```



	ên những ngôn ngữ lập trìnl ức luận lý, biểu thức sau sẽ		_	,	uit evaluation) b
11	a != 0) && ((b / a) > 5))				
((a		1. 2	inh tính toá	n của biểu thứ	. 1 . 1/ 0
	ấy dùng <b>if then else</b> để mô	pnong qua tr	ınn tının toal	ii caa bica iiia	c luận ly trên?
	íy dùng <b>if then else</b> để mô	pnong qua tr	ınn tinn toa		c luận ly trên?
	íy dùng <b>if then else</b> để mô	pnong qua tr	mn tinn toa.	n caa bica tha	c luạn ly tren?
	íy dùng <b>if then else</b> để mô	p pnong qua tr	mn tum toa	ir cua bicu viru	c luạn ly tren?
	íy dùng <b>if then else</b> để mô	p phong qua tr	mn tinn toa	ir cua bicu viru	c luạn ly tren?

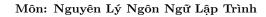
**Lời giải.** if ((a != 0) then if ((b / a) > 8) then ..Giải pháp đề xuất đúng: 2

## III. Phần bài tập lớn: (dành cho tất cả sinh viên)

- 51. Viết hàm visitBlock(ast:Block,o:Context) để thực hiện kiểm tra kiểu. Hàm visitBlock sẽ thực hiện chức năng xây dựng bảng danh hiệu (symbol table) và truyền bảng này khi visit các phát biểu trong khối. Yêu cầu giải thích rõ về Context cụ thể được sử dụng trong hàm và ý tưởng mã của hàm. Nhắc lại AST của Block được khai báo như sau: case class Block(val decl:List[Decl],val stmts:List[Stmt]) extends Stmt
- 52. Viết hàm visitIf(ast:If,o:Context) để thực hiện sinh mã cho một phát biểu If. Nhắc lại AST của If được khai báo như sau: case class If(val expr:Expr, val thenStmt:Stmt, val elseStmt:Option[Stmt]) extends Stmt Yêu cầu giải thích rõ về Context cụ thể được sử dụng trong hàm và ý tưởng mã của hàm.

#### IV. Phần KSTN: (dành cho sinh viên KSTN)

- 53. Viết hàm visitIf(ast:If,o:Context) để thực hiện kiểm tra lỗi FunctionNotReTurn? Yêu cầu giải thích ý tưởng thực hiện thể hiện qua mã của hàm?
- 54. Viết hàm visitBlock(ast:Block,o:Context) để thực hiện sinh mã cho một khối (block)? Yêu cầu giải thích ý tưởng thực hiện sinh mã cho các khai báo array trong một khối?





Question Distribution

Content	Level 1	Level 2	Level 3	Level 4
Introduction		1		
Lexical	1	4	5	
Syntax	2	4	7	
AST		2		1
Total	3	11	12	1

−HÊT

Chủ nhiệm bộ môn	Giảng viên ra đề
Chữ kí:	Chữ kí:
Họ tên:	Họ tên: