Review the Final Exam

1. Ngôn ngữ lập trình cho phép
   A. kết quả trả về của một hàm là một hàm
   B. truyền thông số là một hàm vào cho một hàm khác
   C. xem một hàm như là dữ liệu
   D. có thể gọi một hàm n thông số chỉ với $n-1$ thông số
   E. tất các câu trên đều đúng

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

   ```
   a[i] = 1
   ```

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

   A.
   ```
   iastore
   aload_1
   iload_3
   iconst_1
   ...
   ```

   B.
   ```
   aload_1
   iload_3
   iastore
   iconst_1
   ...
   ```

   C.
   ```
   aload_1
   iload_3
   iconst_1
   iastore
   ...
   ```

   D.
   ```
   aload_1
   iconst_1
   iload_3
   iastore
   ...
   ```

   E.
   ```
   aload_1
   iastore
   iload_3
   iconst_1
   ...
   ```

3. Trong ngôn ngữ lập trình nào, một biến toàn cục có thể được dùng trong một hàm nhưng không thể ghi vào biến toàn cục trong thân hàm nếu không khai báo lại biến đó là toàn cục. Ví dụ(mượn văn phạm C để thể hiện)

   ```
   int x;
   void foo {
       int t;
       t = x; //x được dùng trong foo
       x = 1; // dòng này sẽ gây ra lỗi vì không thể ghi vào x trong foo
       global x; // khai báo x là global
       x = 2; // được phép gán x vào trong foo
   }
   ```

   A. PHP
   B. Java
   C. Scala

D. Javascript

E. Python

4. Cho tập hợp nguyên được định nghĩa như sau trên Scala: `text Set = Int => Boolean`, và cho hàm `forall(s:Set, p:Int => Boolean):Boolean` trả về |True| khi mọi phần tử của tập hợp `s` thỏa hàm `b`. Định nghĩa nào dưới đây dùng để định nghĩa hàm `subSet` xác định `s1` có là tập hợp con của `s2` hay không?

   A. `def subSet(s1:Set, s2:Set) = !forall(s1,s2)`

   B. `def subSet(s1:Set, s2:Set) = forall(s1,x => !s2(x))`

   C. `def subSet(s1:Set, s2:Set) = !forall(s1,x => !s2(x))`

   D. `def subSet(s1:Set, s2:Set) = forall(s1,s2)`

   E. Không có định nghĩa nào phù hợp với yêu cầu

5. Nếu một chương trình con Java có khai báo cái biến cục bộ như sau:

```
float a;
int[] b = new int[10];
long c;
int d;
```

   Chỉ số biến `d` ở trong dãy biến cục bộ (local variable array) trên mà Jasmin là bao nhiêu nếu biến `a` có chỉ số là 1:

   A. 4

   B. 13

   C. 14

   D. 5

   E. Một giá trị khác.

6. Cho biết mã Jasmin của biểu thức viết bằng ngôn nhữ C như sau:

```
a + b * 3
```

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

   A.
```
iload_0
iload_1
iconst_3
imul
iadd
...
```

   B.
```
iload_0
iadd
iload_1
imul
iconst_3
...
```

   C.
```
iload_0
iload_1
iconst_3
iadd
imul
...
```

D.
```
iadd
imul
iload_0
iload_1
iconst_3
...
```

E.
```
iload_0
iload_1
iadd
iconst_3
imul
...
```

Đoạn code sau dùng cho các câu 7-8:

```
int p[10];
int* foo(int x){
    static int q[10];
    int r[10];
    int *s = new int[10];
    switch (x) {
        case 1: return p;
        case 2: return q;
        case 3: return r;
        case 4: return s;
        default: foo(x+1);
    }
}
```

7. Phát biểu nào sẽ gây ra lỗi khi thực thi

   A. `return p`

   B. `return q`

   C. `return r`

   D. `return s`

   E. Không phát biểu nào gây ra lỗi thi thực thi

8. Dãy nào có thể trở thành rác (garbage)

   A. `p`

   B. `q`

   C. `r`

   D. `s`

   E. Không có dãy nào có thể trở thành rác

Đoạn code sau, được viết trên ngôn ngữ dùng **qui tắc tầm vực tĩnh (static-scope rules)**, áp dụng cho các câu 9-14

```
var x; //1
procedure sub1() {
var x; //2
    procedure sub3() {
        var x; //3
        call sub2();
    }
    call sub3();
}
```

```
procedure sub2() {
    x // use x
}
main() {
    call sub1();
}
```

9. Giả sử chuỗi gọi là `main` → `sub1` → `sub2`, tham khảo đến `x` trong `sub2` ứng với khai báo
   A. Báo lỗi danh hiệu `x` chưa khai báo
   B. `x` trong `main` (//1)
   C. `x` trong `sub1` (//2)
   D. `x` trong `sub3` (//3)
   E. `x` trong `sub2`

10. Giả sử chuỗi gọi là `main` → `sub1` → `sub3` → `sub2`, chọn phát biểu ĐÚNG nhất trong các phát biểu sau
    A. IP của bản hoạt động `sub2` chứa địa chỉ nền của bản hoạt động `sub3`
    B. EP của bản hoạt động `sub2` chứa địa chỉ nền của bản hoạt động `sub3`
    C. SCP của bản hoạt động `sub2` chứa địa chỉ nền của bản hoạt động `sub3`
    D. Cả 3 câu trên đều đúng
    E. Cả 4 câu trên đều sai

11. Để `sub2` có thể truy xuất (đọc và ghi) biến `x` khai báo trong `sub1` thì
    A. không thể thực hiện được
    B. để `sub1` gọi trực tiếp `sub2`
    C. thông qua việc truyền thông số bằng tham khảo
    D. Khai báo lại biến `x` trong `sub2`
    E. thông qua việc truyền thông số bằng trị

12. Nếu sửa khai báo `sub3` thành `sub1` thì
    A. dịch và chạy bình thường
    B. để lỗi biên dịch vì trùng tên `sub1`
    C. lỗi biên dịch vì không thể gọi đệ quy
    D. lỗi thực thi vì không thể gọi đệ quy
    E. lỗi thực thi vì trùng tên

13. Nếu đoạn code trên dùng **qui tắc tầm vực động (dynamic-scope rules)**, và giả sử chuỗi gọi là `main` → `sub1` → `sub3` → `sub2`, tham khảo đến `x` trong `sub2` khi thực thi ứng với kết hợp của `x` trong bảng hoạt động của
    A. Báo lỗi không tìm thấy `x`
    B. `main`
    C. `sub1`
    D. `sub2`
    E. `sub3`

14. Ràng buộc giữa một biến và kiểu của nó trên ngôn ngữ Javascript xảy ra trong thời gian
    A. Hiện thực (implementation)

B. Lập trình (programming)

C. Dịch (compiling)

D. Khởi động (loading)

E. Chạy (running)

Phần hướng dẫn này áp dụng cho các câu 16-17
Hãy viết thêm các đoạn mã cần thiết vào phương thức dưới đây để hiện thực sinh mã cho phát biểu *if*

```
def visit(ast:If, emit:Emitter, ctx:Context):Type = {
    val label1 = emit.getFrame().getNewLabel()
    val label2 = emit.getFrame().getNewLabel()
    ast.expr.visit(this,emit,ctx)
    emit.printout(emit.emitIFFALSE(label1))
    ast.thenStmt.visit(this,emit,ctx)
    ast.elseStmt match {
        case Some(s) => {//1
                        //2
                        s.visit(this,emit,ctx)
                        //3
                        }
        case None => //4
    }
    VoidType
}
```

16. Chọn phát biểu ĐÚNG nhất trong các phát biểu sau cho việc sinh ra mã cho phát biểu *If*

A. `emit.printout(emit.emitGOTO(label2))` **ở** `//1`

B. `emit.printout(emit.emitGOTO(label2))` ở `//2`

C. `emit.printout(emit.emitLabel(label1))` ở `//1`

D. `emit.printout(emit.emitLabel(label2))` ở `//2`

E. Câu B và C đúng

17. Chọn phát biểu ĐÚNG nhất trong các phát biểu sau cho việc sinh ra mã cho phát biểu *If*

A. `emit.printout(emit.emitLabel(label2))` **ở** `//3` **và** `emit.printout(emit.emitLabel(label1))` **ở** `//4`

B. `emit.printout(emit.emitGOTO(label2))` ở `//2`

C. `emit.printout(emit.emitLabel(label1))` ở `//3` và `//4`

D. `emit.printout(emit.emitLabel(label1))` ở `//3` và `emit.printout(emit.emitLabel(label2))` ở `//4`

E. `emit.printout(emit.emitLabel(label2))` ở `//3` và `//4`

Doạn code sau được dùng cho các câu hỏi 18-20.
Cho hàm `sum` được viết như sau:

```
int sum(int a, int b, int n) {
    int s = 0;
    while (i < n) {s = s + a; i = i + 1;}
    return s;
}

void main() {
    int A[5] = {2,3,4,5,6}; // chỉ số của dãy A từ 0 đến 4
    int i = 0;
```

```
    int s = sum(A[i],i,5);
    printf("a = %i\n",s); //1
}
```

18. Trong trường hợp `a` và `i` được **truyền bằng trị - kết quả**, `n` được truyền bằng trị, giá trị của `s` khi được in ra tại phát biểu `//1` là:

    A. 30

    B. 10

    C. 15

    D. 20

    E. 25

19. Trong trường hợp `a` và `i` được **truyền tham khảo**, `n` được truyền bằng trị, giá trị của `s` khi được in ra tại phát biểu `//1` là:

    A. 30

    B. 10

    C. 15

    D. 20

    E. 25

20. Trong trường hợp `a` và `i` được **truyền bằng tên**, `n` được truyền bằng trị, giá trị của `s` khi được in ra tại phát biểu `//1` là:

    A. 30

    B. 10

    C. 15

    D. 20

    E. 25

21. Trên ngôn ngữ lập trình hàm thuần khiết (pure functional programming language),

    A. không có kiểu dữ liệu

    B. không có lệnh lặp

    C. không có lệnh gọi hàm

    D. không có lệnh rẽ nhánh (if)

    E. tất cả các phép toán có cùng độ ưu tiên

22. Thời gian xảy ra ràng buộc giữa biến textbfcount trong khai báo C dưới đây và địa chỉ thực của nó là

```
static int count;
```

    A. Hiện thực (implementation)

    B. Lập trình (programming)

    C. Dịch (compiling)

    D. Khởi động (loading)

    E. Chạy (running)

23. Cho mã giả của phát biểu `for (i = 1;i < n;i++) body` như sau:

```
i = 1
while (i < n) {
    body
    i++
}
```

Nếu trong thân `body` có thực thi lệnh `continue` thì điều khiển sẽ chuyển đến thực thi

A. lệnh sau phát biểu `for`

B. biểu thức điều kiện `i < n`

C. `body`

D. `i++`

E. `i = 1`

24. Phương pháp truyền thông số nào KHÔNG dùng để vừa nhập vừa xuất dữ liệu

A. Truyền bằng trị - kết quả (pass by value-result)

B. Truyền tham khảo (pass by reference)

C. Truyền bằng tên (pass by name)

D. Truyền bằng trị (pass by value)

E. Tất cả đều có thể dùng để vừa nhập vừa xuất dữ liệu

25. Hiện tượng gì xảy ra khi thực thi phát biểu `//1` trong đoạn code sau:

```
int *p = new int;
int *q = p;
delete q;
*p = 1; //1
```

A. Con trỏ chưa khai báo (undeclared pointer)

B. Bí danh (alias)

C. Đa hình (polymorphism)

D. Tham chiếu treo (dangling reference)

E. Rác (garbage)

26. Đoạn code nào được viết theo phong cách lập trình hàm

A. `int factorial(int n) {int t; if (n==0) t = 1; else t  = n * factorial(n - 1); return t;}`

B. `int factorial(int n) {int t; int k = 1; for (i = 1;i <= n;i++) k = k * i; return k;}`

C. `int factorial(int n) {if (n == 0) return 1; else return n * factorial(n - 1);}`

D. Cả 3 đoạn code trên đều theo phong cách lập trình hàm

E. Không có đoạn code nào theo phong cách lập trình hàm

Phần hướng dẫn này áp dụng cho các câu 27-30. Hãy viết thêm các đoạn mã cần thiết vào phương thức dưới đây để thực hiện sinh mã cho phát biểu /textitrepeat. Nhắc lại phát biểu *repeat* kết thúc quá trình lặp khi điều kiện của phát biểu trở thành **true**

```
def visit(ast: Repeat, emit:Emitter, ctx:Context): Type = {
    emit.getFrame().enterLoop();
    val labelStart = emit.getFrame.getNewLabel()
    val labelBreak = emit.getFrame.getBreakLabel()
    val labelCont = emit.getFrame.getContinueLabel()
    //1
    ast.loop.foreach(_.visit(this, emit, ctx)) //sinh mã cho từng phát biểu trong thân của repeat
    //2
    ast.expr.visit(this, emit, ctx)) // sinh mã cho từng phát biểu thức điều kiện
    //3
    //4
    emit.getFrame().exitLoop();
    VoidType
}
```

27. **Data loss**

28. **Data loss**

29. **Data loss**

30. **Data loss**

31. Trong các phương pháp truyền thông số dưới đây, phương pháp nào làm thay đổi thông số thực ngay khi thông số hình thức bị thay đổi

    A. Truyền bằng trị - kết quả (pass by value-result)

    B. Truyền tham khảo (pass by reference)

    C. Truyền bằng tên (pass by name)

    D. Cả 3 phương pháp trên

    E. Truyền tham khảo và truyền bằng tên

Phần hướng dẫn sau cho các câu 32-34. Trện những ngôn ngữ lập trình cho phép rút ngắn tính toán (short-circuit evaluation) biểu thức luận lý, phát biểu sau là hợp lệ:

```
if ((a != 0) && ((b / a) > 8)) then ...
```

32. Nhưng trên những ngôn ngữ không rút ngắn tính toán biểu thức luận lý, biểu thức trên sẽ có thể gây ra lỗi gì?

33. Nếu ngôn ngữ không hõ trợ rút ngắn tính toán biểu thức luận lý thì cần viết lại biểu thức trên như thế nào?

34. Nếu biểu thức trên xuất hiện trong phát biểu **while**, tức `while ((a != 0 && ((b / a) > 8)) do {body}` thì cần viết lại như thế nào?

35. Giả sử giá trị ban đầu của biến `c` là 3, cho biết giá trị của biến `a` trong phát biểu sau trên C có thể là những giá trị nào?

    ```
    a = c * (c = 4) + c;
    ```

36. Viết phương thức `visitIf(ast:If, ctx:Context)` để sinh ra mã cho môt phát biểu *If*. Cho biết lớp của phát biểu *If* trên AST được định nghĩa như sau:

    ```
    case class If(val expr:Exp, val thenStmt:Stmt, val elseStmt:Option[Stmt]) extends Stmt
    ```

    Một số phương thức của Emitter có thể sử dụng:
    `emitIFTRUE(label:Int, frame:Frame)`, `emitIFFALSE(label:Int, frame:Frame)`,
    `emitGOTO(label:Int, frame:Frame)emitLABEL(label:Int, frame:Frame)`,`printout(str:String)`.
    Một số phương thức của Frame có thể sử dụng:
    `getNewLabel()`,`getBreakLabel()`,`getStartLabel()`,`getEndLabel()`.

37. Giả sử chỉ có các phép toán AND(&&) trên biểu thức nhị phân, hãy viết hàm `visitBinaryOp(ast:BinaryOp, o:Context)` để sinh ra mã cho phép toán AND. Chú ý phải sinh ra mã cho phép rút ngắn tính toán (short-circuit) thì mới được trọn điểm câu này. Nhắc lại khai báo lớp `BinaryOp` trên cây AST như sau:
`case class BinaryOp(op:String, left:Exp, right:Exp) extends Exp`.
Các phương thức của Emitter và Frame đã nêu ở câu trên.

38. Hãy nêu điểm khác biệt chính của các cơ chế gọi chương trình con đệ quy (recursive), biến cố (exception), trình cộng hành (coroutine), trình định thời (scheduled subprogram) và công tác (task) so với cơ chế gọi - trở về đơn giản (simple call-return).

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

```
int A[3] = {1, 9, 45}; // index of A start from 0
int j = 0;
int n = 3;

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);
    cout << s << A[0] << A[1] << A[2]; // 1
}
```

Hãy cho biết và giải thích kết quả in ra của chương trình trong các trường hợp sau:

(a) Nếu `a` và `i` được truyền bằng trị - kết quả.

(b) Nếu `a` và `i` được truyền bằng tham khảo.

(c) Nếu `a` và `i` được truyền bằng tên.

40. Cho đoạn mã sau được viết trên ngôn ngữ **quy tắc tầm vực tĩnh** (static-scope rule):

```
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);
}
```

(a) Cho biết môi trường tham khảo tĩnh (static referencing enviroment) cho các thủ tục `main`, `sub1`, `sub2`, `sub3`.

(b) Cho biết tầm vực tĩnh của các khai báo `a//1`, `b//1`, `a//2`, `sub2`, `a//3`, `sub3`.

41. 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;
```

(a) Vẽ mô hình mô tả một đối tượng (các thành phần, kích thước các thành phần) có kiểu Figure và qua đó tính tổng kích thước của một đối tượng kiểu Figure. Giả sử kích thước của kiểu `Integer`, `Float`, `Boolean` và `Enumeration` lần lượt là 4, 4, 1 và 4. Cần chú ý về vấn đề padding.

(b) Trên những ngôn ngữ lập trình không có kiểu union như Java, Scala, C# thì làm thế nào để thực hiện một đối tượng kiểu union (như `Figure`)? Hãy thực hiện kiểu Figure trên một trong các ngôn ngữ lập trình Java, Scala, C#.

42. Cho đoạn mã sau:

```
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);
    }
}
```

(a) Hãy xác định và giải thích các lỗi liên quan đến con trỏ (tham chiếu treo - dangling reference, rác - garbage) có thể xảy ra khi thực thi đoạn mã trên.

(b) Khi hàm `foo` được gọi đệ quy thì các bản hoạt động của foo dùng chung những đối tượng dữ liệu nào (`p`, `q`, `s`, `x`, đối tượng được tạo ra bởi `new int`)? Giải thích.

43. Chuyển sang dạng tiền tố (Cambridge Polish Prefix):

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

Yêu cầu: cùng thứ tự xuất hiện toán hạng. Số dấu '(' và ')' ít nhất. Cùng thứ tự tính toán.

44. `def lookup[T](x: String, lst: List[T], f: T => String): Option[T]`
`x` là chuỗi cần tìm kiếm, `lst` là danh sách để thực hiện tìm kiếm, `f` là hàm để trích xuất thành phần kiểu chuỗi trong mỗi phần tử của `lst` để so sánh với `x` khi tìm kiếm.
Hãy hiện thực (viết thân hàm) `lookup` trên bằng ngôn ngữ Scala để thực hiện tìm kiếm một chuỗi trên danh sách `lst` với hàm `f`.

45. Hãy hiện thực (viết các chương trình thể hiện các ràng buộc kiểu) suy diễn kiểu để suy ra kiểu của hàm sau:

```
H(x, f, h) {
    s := 0
    for i := f(x) to h(x) do s := s + i
    return s
}
```