#3:

--reference chỉ tới một thực thể có giá trị bằng tên khác

trong khi đó pointer chứa giá trị địa chỉ

---reference không thể null còn pointer thì được

---reference phải có khởi tạo còn pointer thì k bắt buộc

---reference chia sẻ bộ nhớ với thằng gốc, pointer thì có phần bộ nhớ riêng

#4:

---đệ quy: năng lực lớn nhất của đệ quy là lặp lại

bản thân chức năng của nó với lượng code cực ngắn mà lại

hiệu quả cao, điểm trừ là khó kiểm soát điểm dừng

---coroutine: tương tự như là thread , nhưng tại một

thời điểm, luồng chạy song song còn coroutine thì không

---scheduled subroutine: subroutine cũng là một callable unit

trình định thời thì là một phần được liệt kê trong

chương trình hoặc được tạo ra trong thư viện, nó được

gọi rad để phục vụ việc phân phối công việc (tùy vào cách hiểu,

có trường hợp có thể hiểu đây là chương trình đã được lên lịch)

liên quan đến scheduled subprograms: có 2 cách thể hiện

phân lịch bằng thời gian

phân lịch bằng ưu tiên thứ tự

-----------------------------------------------------------

---Task: có thể thực hiện cùng lúc với nhiều task khác

chạy bằng tiến trình đa nhân hoặc đơn nhân với trình phân bổ

các vấn đề liên quan: race condition và deadlock

---simple call-return: không có recursion; phải có điểm gọi cụ thể

có điểm vào duy nhất, không cần những thủ tục chuyển giao hoặc khống chế

chạy đơn

#5:

đầu tiên là tham chiếu treo: tham chiếu tới địa chỉ không còn tồn tại, hoặc sai

---lock and key:

phía pointer: cầm địa chỉ và key

phía bộ nhớ: có giá trị và một vùng mở rộng cho lock

---tombstone: tomestone set a value when the memory is set

when it is dealoacated the tomestone value set to null

this say that the variable is no longer exitst

#6:

-------------------------------------------

def visitIf(self, ast, c):

lst = [[],[]]

lst[1] = c[1] + c[0]

lst[0] = []

errCheck = []

for i in ast.ifthenStmt:

errCheck = errCheck + [self.sortType(i[0].accept(self, lst))]

for j in i[1]:

lst[0] = lst[0] + [j.accept(self, lst)]

for j in i[2]:

j.accept(self, lst)

for i in errCheck:

if i != BoolType:

raise TypeMismatchInStatement(ast)

elseStmt = ast.elseStmt

for j in elseStmt[0]:

lst[0] = lst[0] + [j.accept(self, lst)]

for j in elseStmt[1]:

j.accept(self, lst)

--------------------------------------------

def visitReturn(self, ast, c):

if ast.expr == None:

return VoidType

if type(ast.expr.accept(self, c)) in self.primType:

return type(ast.expr.accept(self, c))

return Unknown

------------------------------------------------

#7

a)

---main: a,b,c //1; sub1//2; print(b) //5

---sub1: main tr? a//2; sub2//3; sub3//4

---sub2: sub1 trừ a,c,f //3

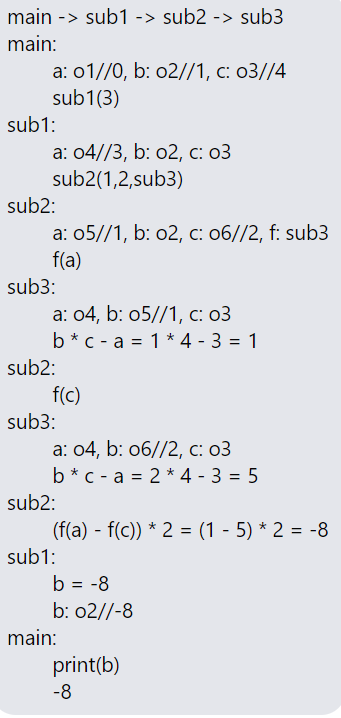
---sub3: sub2 trừ b//4

b)

//xem như compiler đã thực hiện xong, chỉ thực hiện gọi

Phần giải của Khánh

chươngng trình | hoạt động



#8:

a) truyền theo value-result:

---9;1;6;14

b) truyền theo địa chỉ:

---9;1;6;14

c) truyền theo tên:

---10;3;5;13