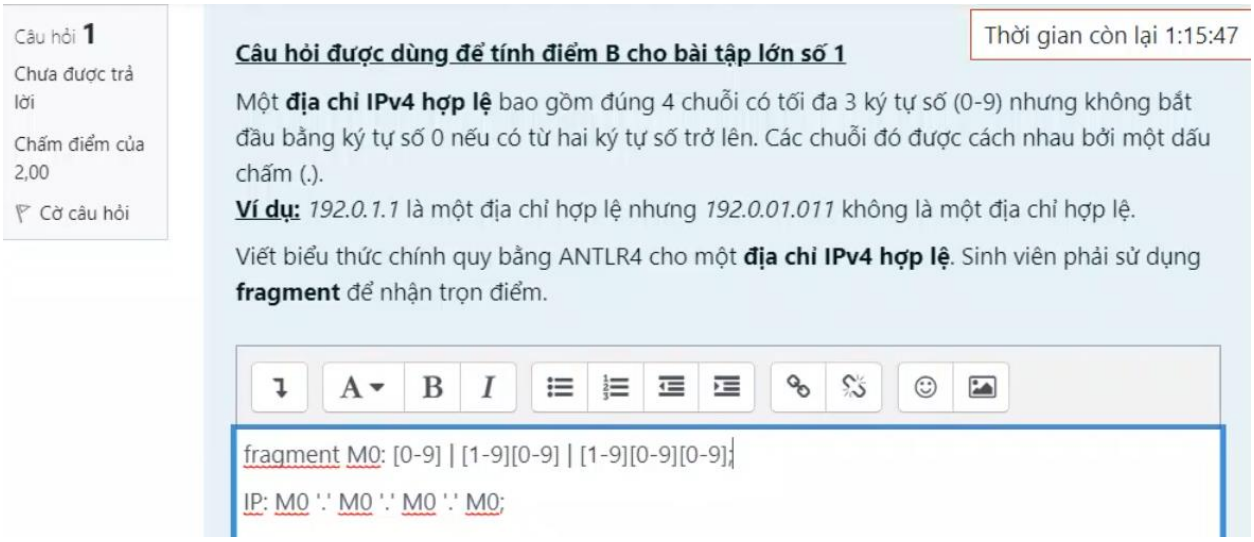



```
fragment M: [0-9] | [1-9][0-9] | [1-9][0-9][0-9];

IP: M '.' M '.' M '.' M;
```

Câu hỏi **2**

Chưa được trả lời

Chấm điểm của 3,00

Cờ câu hỏi

Câu hỏi được dùng để tính điểm B cho bài tập lớn số 1

Trong ngôn ngữ **MiniPHP**, chương trình bao gồm các khai báo biến. Một *khai báo biến* được gắn liền với lần đầu tiên biến đó được gán giá trị. Phép gán trong MiniPHP bao gồm các thành phần theo thứ tự tên biến ***VARNAME***, dấu bằng ***EQ***, một biểu thức và kết thúc bởi dấu chấm phẩy ***SEMI***.

Biểu thức trong MiniPHP là tổ hợp của các *toán hạng* và các *toán tử* được viết theo trung thứ tự (infix expression).

- Các toán hạng bao gồm: tên biến, hằng số nguyên ***INTLIT***, hằng số thực ***FLOATLIT***, hằng chuỗi ***STRINGLIT*** hoặc một mảng. Có hai loại mảng trong MiniPHP là mảng chỉ số (indexed array) và mảng phối hợp tên (associative array).

+ *Mảng chỉ số* bắt đầu bằng từ khóa array ***ARRAY*** tiếp theo là một danh sách có thể rỗng các biểu thức được phân cách bởi một dấu phẩy ***COMMA*** và được bao lại bằng một cặp ngoặc tròn ***LP*** và ***RP***.

+ *Mảng phối hợp tên* bằng từ khóa array **ARRAY** tiếp theo là một danh sách có thể rỗng *các cặp kết hợp* (associative pair) được phân cách bởi một dấu phẩy ***COMMA*** và được bao lại bằng một cặp ngoặc tròn ***LP*** và ***RP***. Một cặp kết hợp bao gồm một tên cặp ***PAIRNAME***, tiếp theo là dấu mũi tên ***ARROW*** và sau đó là một biểu thức.

- Các toán tử được liệt kê theo độ ưu tiên từ cao xuống thấp (các toán tử được mô tả trên cùng một dòng sẽ cùng một độ ưu tiên) và chỉ rõ tính kết hợp:

+ Toán tử ** ***DSTAR***: kết hợp phải

+ Toán tử . ***DOT***: kết hợp trái

+ Toán tử * ***MUL***, / ***DIV***, % ***MOD***: kết hợp trái

+ Toán tử + ***ADD***, - ***SUB***: kết hợp phải.

+ Toán tử ?? ***DQUES***: không có tính kết hợp.

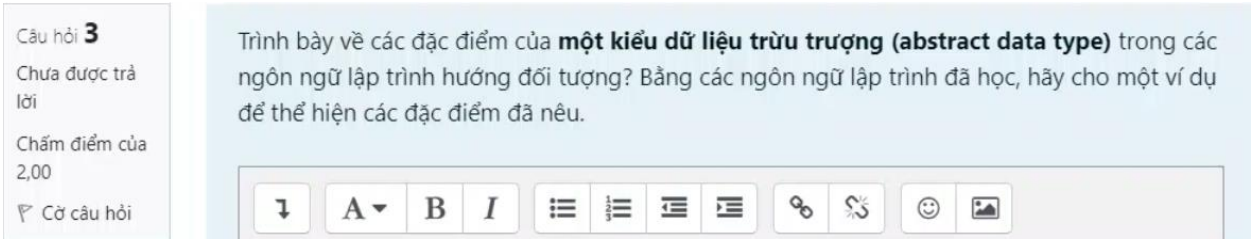
- Để thay đổi được độ ưu tiên và tính kết hợp, người ta có thể sử dụng cặp ngoặc tròn để tạo biểu thức con.

Sử dụng ANTLR4 để mô tả ngôn ngữ nói trên. Biết rằng các từ in đậm và nghiêng là tên các từ vựng trong ngôn ngữ đã đặt, sinh viên trực tiếp sử dụng các tên này.

Ví dụ:

```
abc = 1 + 2 ?? 3;
u = array(a1 => 3 . 4, a2 => 3 + (u2 % 5));
```

```
program: var_decl* EOF;

var_decl: VARNAME EQ exp SEMI;


indexexd_array: ARRAY LP explist RP;

explist: exps | ;

exps: exp | exp COMMA exps;


ass_array: LP ass_pairlist RP;

ass_pairlist: ass_pairs | ;

ass_pairs: ass_pair | ass_pair COMMA ass_pairs;


ass_pair: PAIRNAME ARROW exp;


exp: exp1 DQUES exp1 | exp1;

exp1: exp2 (ADD | SUB) exp1 | exp2;

exp2: exp2 (MUL | DIV | MOD) exp3 | exp3;

exp3: exp3 DOT exp4 | exp4;

exp4: exp5 DSTAR exp4 | exp5;

exp5: VARNAME | INTLIT | FLOATLIT | STRINGLIT | indexed_array |

ass_array | exp6;

exp6: LP exp RP;
```



Một kiểu dữ liệu trừu tượng cần thoả mãn hai điều kiện:

- Encapsulation (Tính bao đóng): Thuộc tính trong một đối tượng được quy định tầm vực rõ ràng
- Information Hiding (ẩn dữ liệu): Khi sử dụng phương thức, người dùng chỉ cần biết phương thức đó làm gì, không cần biết chi tiết bên trong các thuộc tính -> tăng độ tin cậy cho dữ liệu

Ví dụ:

Trong ngôn ngữ C++, chúng ta có thể khai báo một lớp như sau:

```
class A {
        private:
        string name;
        string age;
        public:
        string getName() {
                return this.name;
        }
}
```

Vì name và age có tầm vực là private nên ở bên ngoài class A, người dùng không thể truy cập vào name hoặc age, tuy nhiên người dùng có thể lấy được giá trị của name thông qua hàm getName(), đó chính là tính ẩn dữ liệu

Câu hỏi **4**

Chưa được trả lời

Chấm điểm của 3,00

Cờ câu hỏi

Câu hỏi được dùng để tính điểm B cho bài tập lớn số 2

Yêu cầu: Sinh viên sử dụng lập trình hàm để thực hiện câu hỏi sau đây. Nếu không sử dụng lập trình hàm, sinh viên chỉ nhận tối đa 1.5 điểm cho câu hỏi này.

Cho một đoạn ngữ pháp mô tả ngôn ngữ X như sau:

```
program: vardecl+ EOF;
vardecl:  idlist AT typ SM;
idlist: mem (CM mem)*;
mem: STATIC? ID;
typ: INT | FLOAT;
INT: 'int'; FLOAT: 'float';
AT: '@';
CM: ',';
STATIC: 'static';
SM: ';';
ID: [a-zA-Z]+;
WS: [ \r\n] -> skip;
```

Mô tả: chương trình dùng để khai báo biến (static/ non-static).

Ví dụ: a, static b @ int; static c, d @ float; là hai lệnh phù hợp trong ngôn ngữ đã mô tả, trong đó biến a, d (không từ khóa static) được khai báo là các biến non-static, biến b, c (dùng từ khóa static) được khai báo là các biến static.

Cho các lớp AST như sau:
class AST(ABC)
class Program(AST): decl: List[VarDecl]
class VarDecl(AST): #id: str, static: bool, typ: Type, tail: VarDecl = None
class Type(AST)
class IntType(Type)
class FloatType(Type)

Hãy **viết các phương thức** của lớp ASTGeneration để sinh cây AST cho trường hợp ngôn ngữ X nói trên.

Ví dụ: a, static b @ int; static c, d @ float; -> Program([VarDecl("a", False, IntType(), VarDecl("b", True, IntType(), None)), VarDecl("c", True, FloatType(), VarDecl("d", False, FloatType(), None))]
Lưu ý: để sử dụng tab indent, sinh viên có thể sử dụng whitespace hoặc sử dụng các nút  trên trình soạn thảo trực tiếp.

```
class ASTGeneration(D96Visitor):
   def visitProgram(self, ctx: D96Parser.ProgramContext):
      varDeclList = []
      for i in range(len(ctx.vardecl())):
         varDeclList += self.visit(ctx.vardecl(i))
      return Program(varDeclList)




   def visitVardecl(self, ctx: D96Parser.VardeclContext):
      varType = self.visit(ctx.typ())
      member = self.visit(ctx.idlist())
      size = len(member)
      tailVarDecl = VarDecl(member[size - 1][1], member[size -
1][0], varType)

      for i in range(0, size - 1):
         idx = size - 2 - i
         tailVarDecl = VarDecl(member[idx][1], member[idx][0],
varType,tailVarDecl)

      return [tailVarDecl]
```

```
    def visitIdlist(self, ctx: D96Parser.IdlistContext):
        if ctx.getChildCount() == 1:
            return [self.visit(ctx.mem(0))]
        else:
            memList = []
            for i in range(0, len(ctx.CM()) + 1):
                memList.append(self.visit(ctx.mem(i)))
            return memList

    def visitMem(self, ctx: D96Parser.MemContext):
        return (True, ctx.ID().getText()) if ctx.STATIC()
else (False, ctx.ID().getText())



    def visitTyp(self, ctx: D96Parser.TypContext):
        return IntType() if ctx.INT() else FloatType()
```