



## فاز ۱ پروژه

شماره دانشجویی:

نام و نام‌خانوادگی:

## ۱ توضیحات گرامر:

گرامر کامل زبان ترکیبی از گرامر سی، کمی از گرامر پایتون و ساده‌سازی‌های بسیار برای راحتی پیاده‌سازی و نوشتن لکسر و پارسر است، برای مثال، در گرامر استفاده شده تنها از `while` برای دستورات حلقه استفاده شده و از ایف‌های تک خطی (بدون اسکوپ) پشتیبانی نشده.

برای ساپورت کردن از آرایه، تصمیم به این شد که علاوه بر دیتا تایپ‌های بیسیک مانند `int` و `float` از دیتا تایپی به نام دیتا تایپ لیست استفاده شود، که سینتکس تعریف لیست به صورت زیر است:

```
list<typename> name;
```

example

```
list<list<int>> a;
```

در این زبان، تمام آرایه‌ها داینامیک سائز تعریف می‌شوند و آرایه با سائز ثابت در آن وجود ندارد، همچنین برای تورینگ کامپلیت بودن سینتکس زبان، علاوه بر دو استیتمنت ابتدایی `input` و `output` چهار دستور کار با لیست وجود دارد که عبارتند از `push`، `pop`، `get`، `set` که عملکرد آن‌ها از اسم آنها مشخص است، دقت کنید چون آرایه‌ها داینامیک سائز هستند و اجازه دسترسی به آنها توسط آپرند داده نشده، داشتن حداقل ۲ تا از این توابع یعنی `get` و `push` برای تورینگ کامپلیت بودن الزامیست، ۲ تابع دیگر هم برای راحتی به استیتمنت‌های از پیش تعریف شده زبان اضافه شدند.

دقت کنید که تمامی ۶ دستور از پیش معین شده، با علامت `$` شروع می‌شوند.

در این زبان، درست مثل پایتون می‌توان در هر اسکویی تابع را تعریف کرد و تعریف توابع مانند سی پلاس پلاس به اسکوپ گلوبال محدود نیستند، همچنین توابع باید با سمیکالن پایان یابند، برای مثال:

```
int a;
$input(a);
int factorial(int n) {
    if (n == 0) {
        return 1;
    } else {
        return n * factorial(n - 1);
    }
};
```

```
$output(factorial(a));
```

## ۲ گرامر کامل زبان:

در ادامه گرامر کامل زبان را نوشته‌ایم:

$\text{Spc} \rightarrow \text{Space} \mid \text{Tab} \mid \text{NewLine}$

$\text{Program} \rightarrow \text{StatementSeq}$

$\text{SimpleStatement} \rightarrow \text{VarDec} \mid \text{FunDec} \mid \text{Assignment} \mid \text{Expression} \mid$   
 $\text{BreakStatement} \mid \text{ContinueStatement} \mid \text{ReturnStatement} \mid \text{PreDefinedStatement}$

$\text{Statement} \rightarrow \text{SimpleStatement}; \mid \text{IfStatement} \mid \text{WhileStatement}$

$\text{StatementSeq} \rightarrow \text{Statement StatementSeq} \mid \text{Scope StatementSeq} \mid \text{empty}$

$\text{Scope} \rightarrow \{ \text{StatementSeq} \}$

$\text{Expression} \rightarrow \text{Exp7}$

$\text{IfStatement} \rightarrow \text{if} ( \text{Expression} ) \text{Scope} \mid$   
 $\text{if} ( \text{Expression} ) \text{Scope} \text{ else } \text{Scope} \mid$   
 $\text{if} ( \text{Expression} ) \text{Scope} \text{ else } \text{IfStatement}$

$\text{WhileStatement} \rightarrow \text{while} ( \text{Expression} ) \text{Scope}$

$\text{Word} \rightarrow "(a-z, A-Z, \_)(a-z, A-Z, \_, 0-9)^*"$

$\text{VarName} \rightarrow \text{Word}$

$\text{FunName} \rightarrow \text{Word}$

$\text{VarType} \rightarrow \text{int} \mid \text{float} \mid \text{string} \mid \text{char} \mid \text{bool} \mid \text{list} < \text{VarType} >$

$\text{VarTypeName} \rightarrow \text{VarType Spc VarName}$

$\text{VarDec} \rightarrow \text{VarTypeName} \mid \text{VarTypeName} = \text{Expression}$

$\text{VarSeq} \rightarrow \text{VarTypeName}, \text{VarSeq} \mid \text{VarTypeName}$

$\text{FunDec} \rightarrow \text{VarType Spc FunName} ( ) \text{Scope} \mid$   
 $\text{VarType Spc FunName} ( \text{VarSeq} ) \text{Scope}$

$\text{ExpSeq} \rightarrow \text{Expression} \mid \text{Expression}, \text{ExpSeq}$

$\text{FunCal} \rightarrow \text{FunName} ( \text{ExpSeq} ) \mid \text{FunName} ( )$

$\text{BreakStatement} \rightarrow \text{break}$

$\text{ContinueStatement} \rightarrow \text{continue}$

$\text{ReturnStatement} \rightarrow \text{return} \mid \text{return Spc Expression}$

$\text{Assignment} \rightarrow \text{VarName} = \text{Expression}$

$\text{Op1} \rightarrow ! \mid \sim$

$\text{Op2} \rightarrow * \mid / \mid \%$

$\text{Op3} \rightarrow + \mid -$

$\text{Op4} \rightarrow < = \mid > = \mid < \mid > \mid = = \mid ! =$

$\text{Op5} \rightarrow \& \mid ^ \mid |$

$\text{Op6} \rightarrow \&\&$

$\text{Op7} \rightarrow ||$

$\text{Exp0} \rightarrow \text{Atom} \mid ( \text{Exp7} )$

$\text{Exp1} \rightarrow \text{Op1 Exp0} \mid \text{Exp0}$

$\text{Exp2} \rightarrow \text{Exp2 Op2 Exp1} \mid \text{Exp1}$

$\text{Exp3} \rightarrow \text{Exp3 Op3 Exp2} \mid \text{Exp2}$

$\text{Exp4} \rightarrow \text{Exp4 Op4 Exp3} \mid \text{Exp3}$

$\text{Exp5} \rightarrow \text{Exp5 Op5 Exp4} \mid \text{Exp4}$

$\text{Exp6} \rightarrow \text{Exp6 Op6 Exp5} \mid \text{Exp5}$

$\text{Exp7} \rightarrow \text{Exp7 Op7 Exp6} \mid \text{Exp6}$

$\text{Atom} \rightarrow \text{PreDefinedStatement} \mid \text{FunCal} \mid \text{VarName} \mid \text{Value}$

$\text{Value} \rightarrow \text{BoolVal} \mid \text{IntVal} \mid \text{FloatVal} \mid \text{CharVal} \mid \text{StrVal}$

$\text{BoolVal} \rightarrow \text{false} \mid \text{true}$

$\text{IntVal} \rightarrow (+-)?(0-9)^+$

$\text{FloatVal} \rightarrow \text{IntVal} \mid (+-)?(0-9)^*\backslash.(0-9)^+$

$\text{CharVal} \rightarrow \text{'.'}' \mid \text{'\.'}'$

$\text{StrVal} \rightarrow \text{""} \mid \text{"} \cdot * (\text{non } \backslash) \text{"}$

$\text{PreDefinedStatement} \rightarrow \text{PrintStatement} \mid \text{InputStatement} \mid \text{GetStatement} \mid$   
 $\text{SetStatement} \mid \text{PushStatement} \mid \text{PopStatement}$

$\text{PrintStatement} \rightarrow \$\text{print} ( \text{Expression} )$

$\text{InputStatement} \rightarrow \$\text{input} ( \text{VarName} )$

$\text{GetStatement} \rightarrow \$\text{get} ( \text{VarName}, \text{Expression} )$

$\text{SetStatement} \rightarrow \$\text{set} ( \text{VarName}, \text{Expression}, \text{Expression} )$

$\text{PushStatement} \rightarrow \$\text{push} ( \text{VarName}, \text{Expression} )$

$\text{PopStatement} \rightarrow \$\text{pop} ( \text{VarName} )$