

به نام خداوند بخشنده مهربان



فاز ۱ پروژه - درس اصول و طراحی کامپایلرها

دکتر ممتازی

ترم زمستان ۱۳۹۸-۱۳۹۹ - دانشکده کامپیوتر، دانشگاه صنعتی امیرکبیر

زمان تحویل: ۵ فروردین ۹۹

لطفا قبل از شروع به حل کردن پروژه به نکات زیر توجه فرمایید:

۱. هدف از انجام پروژه، یادگیری عمیق تر مطالب درسی است. در نتیجه هرگونه کپی برداری موجب کسر نمره خواهد شد.
۲. تا ساعت ۲۳:۵۵ روز ۵ فروردین ماه فرصت دارید تا تمرین را در مدل بارگذاری کنید. تمام فایل های خواسته شده را در یک فایل فشرده قرار دهید. نام فایل نهایی را شماره دانشجویی خود قرار دهید. (برای مثال dlexer\_Family\_9631747.zip)
۳. در کنار این فایل، ۳ ویدئو برای راهنمایی شما وجود دارد که توصیه می کنیم آن ها را مشاهده بفرمایید. محتوای ارائه شده در فیلم ها در آدرس گیت هاب زیر قابل دسترسی است:  
<https://github.com/Mohammad-Robati/compiler99>
۴. لطفا ۳ فایل متنی بارگزاری شده را به عنوان ورودی در نظر بگیرید و خروجی آن را در فایل های جداگانه به همراه کدهای برنامه در فایل فشرده قرار دهید.
۵. در صورت وجود هرگونه سوال می توانید از طریق ایمیل با تدریساران در ارتباط باشید.

[moh.robati@aut.ac.ir](mailto:moh.robati@aut.ac.ir)

[sheykh@aut.ac.ir](mailto:sheykh@aut.ac.ir)

ما از شما در این بخش پروژه می‌خواهیم که یک تحلیل‌گر لغوی<sup>۱</sup> طراحی کنید که بتواند لغات<sup>۲</sup> را در برنامه تشخیص داده و نشانه<sup>۳</sup> مربوط به آن را برگرداند. همچنین تحلیل‌گر شما باید بتواند تشخیص بدهد که آیا لغت درست است یا خیر (یعنی آیا نشانه ای معتبر برای آن وجود دارد یا خیر) و در صورت غلط بودن بخش غلط را اعلام کند و به تحلیل ادامه بدهد.

Lexeme	Token Value
Identifier	ID
Integer number	INTEGERNUMBER
Float number	FLOATNUMBER
"int"	INTEGER
"float"	FLOAT
"bool"	BOOLEAN
"void"	VOID
"true"	TRUE
"false"	FALSE
"print"	PRINT
"return"	RETURN
"main"	MAIN
"if"	IF
"else"	ELSE
"elif"	ELIF
"while"	WHILE
"for"	FOR
"&&"	AND
"  "	OR
"!"	NOT
"="	ASSIGN
"+"	SUM
"_"	SUB
"*"	MUL
"/"	DIV
"%"	MOD
">"	GT
">="	GE

<sup>1</sup>Lexical analyzer

<sup>2</sup> Lexemes

<sup>3</sup> Token

"<"	LT
"<="	LE
"=="	EQ
"!="	NE
"{"	LCB
"}"	RCB
"("	LRB
")"	RRB
"["	LSB
"]"	RSB
","	SEMICOLON
" "	COMMA
Error	ERROR!

#### Identifier:

شناسه<sup>۴</sup> می تواند شامل حروف بزرگ و کوچک لاتین، ارقام عددی و \_ باشد.

شناسه نباید با عدد شروع شود؛ به طور مثال "3id" غلط است و باید به عنوان نشانه "ERROR" برگردانده شود.

همچنین شناسه نمی تواند یکی از کلمات ذخیره شده مانند "main", "print", "for" و ... باشد.

#### Integer number :

عدد طبیعی شامل ارقام بین ۰ تا ۹ است و همچنین تعداد ارقام عدد طبیعی باید کمتر از ۱۰ باشد. توابعی که برای برگرداندن توکن های اعداد صحیح می نویسد نباید صفر های ابتدا<sup>۵</sup> را برگردانند، یعنی برای عدد ۰۰۲ مقدار ۲ برگردانده می شود.

#### Float number :

عدد حقیقی به صورت دو عدد طبیعی با یک "." در وسط است و همچنین تعداد ارقام بخش طبیعی عدد باید کمتر از ۱۰ باشد.

همچنین توابعی که برای برگرداندن توکن های اعداد اعشاری می نویسد نباید صفرهای ابتدای بخش طبیعی و صفرهای انتهایی بخش اعشاری<sup>۶</sup> را برگردانند، یعنی برای عدد ۰۰۲۳۰۱,۲۰۳۰۰ مقدار ۲۳۰۱,۲۰۳ برگردانده می شود.

توجه داشته باشید که اعداد منفی در این مدل تحلیل نداریم.

<sup>4</sup> Identifier

<sup>5</sup> Reserved

<sup>6</sup> Leading Zero

<sup>7</sup> Trailing Zero

همچنین در لکسر باید تضمین کنید که ۲ عملگر اصلی (/\*-+) نمی توانند در کنار هم قرار بگیرند مگر کاراکتری غیر از **whitespace** میان آن ها باشد و در این صورت کل عملگر ها باید در کنار هم در یک **error** باشند.

برنامه شما باید در مقابل ورودی های مقابل خروجی مورد نظر را بدهد.

خروجی	ورودی
ERROR SUM ID ASSIGN INTEGERNUMBER	19a + b = 9
WHILE LRB RRB	while ()
ID ERROR ID	_x-+-x_
ERROR SUB FLOATNUMBER	9.7.9 - 0.0
ID FOR RSB SUB INTEGERNUMBER LSB	For for ]-90[
AND OR SUB EQ	&&    -==
ID ERROR SEMICOLON	C++ ;
ID SUM ERROR	ERROR + 1231231239.509
INTEGERNUMBER NE NOT TRUE	5!=!true
INTEGERNUMBER ERROR INTEGERNUMBER ID	9 - * - 8 _
VOID MUL LE INTEGERNUMBER	void * <= 0

موفق باشید