

به نام خدا

پروژه درس اصول طراحی کامپایلر

نیمسال اول ۱۴۰۲-۱۴۰۳

طراحی کامپایلری که یک عبارت محاسباتی را بخواند و ضمن تولید three-address code به زبان C، حاصل عبارت را محاسبه کند.

- عبارت شامل اعداد صحیح (حداکثر ۱۰ رقمی)، عملگرهای جمع، تفریق، ضرب و تقسیم و پرانتز و فاصله خالی است و هر عملگر بصورت زیر عمل می کند:

- $a+b$: ارقامی از عدد b که در عدد a نیستند به انتهای a الحاق می شوند.

- $a-b$: ارقامی از عدد b که در عدد a هستند از a حذف می شوند.

- $a*b$: رقم حاصل از جمع ارقام (یا جمع جمع ارقام) عدد b در صورت عدم وجود در a به انتهای a الحاق می شود.

- a/b : رقم حاصل از جمع ارقام (یا جمع جمع ارقام) عدد b در صورت وجود در a از a حذف می شود.

- اولویت عملگرها و شرکت پذیری آنها مطابق معمول است.

- فرض می شود که عبارت ورودی فاقد خطای کامپایلری است.

- چند نمونه :

Input of compiler	Output of compiler
$34276524+781=$	$t1 = 34276524+781;$ $t1 = 3427652481;$
$342765- 7661 =$	$t1 = 342765-7661;$ $t1 = 3425;$
$3142752 *782=$	$t1 = 3142752*781;$ $t1 = 31427528;$
$34276524 / 121=$	$t1 = 34276524/121;$ $t1 = 327652;$
$34276+342 *34 - 734/(25 +44) =$	$t1 = 342*34;$ $t1 = 3427;$ $t2 = 25 +44;$ $t2 = 2544;$ $t3 = 734/t2;$ $t3 = 734;$ $t4 = 34276+t1;$ $t4 = 34276;$ $t5 = t4-t3;$ $t5 = 26;$

- تنها سه فاز lexical analysis، syntax analysis و intermediate code generation طراحی شوند.
- طراحی فاز lexical analysis با استفاده از ابزار Lex یا FLex صورت گیرد.
- برای فاز syntax analysis و ترجمه همزمان به intermediate code از ابزار Yacc یا Bison استفاده شود.
- پروژه (۱ نمره از ۲۰ نمره کل) بصورت تک نفری انجام شود.
- یک فایل zip شامل پروژه و گزارش بصورت Family_Name.rar به ایمیل egmansoori@yahoo.com ارسال شود.
- مهلت دریافت ایمیل قبل از ساعت ۲۴ روز ۲۳ دی می باشد.
- زمان و ترتیب تحویل حضوری در بازه ۸ تا ۱۲ بهمن اطلاع رسانی خواهد شد.

موفق باشید