



این تمرین مباحث مربوط به فاز تحلیل گر لغوی در طراحی و ساخت کامپایلرها را پوشش می دهد.

۱- می خواهیم یک تحلیل گر لغوی برای زبان برنامه نویسی C ایجاد نماییم. وظیفه اصلی تحلیل گر لغوی شناسایی لغت (Token) های موجود در متن یک برنامه و ارسال آن به تحلیل گر نحوی است. وظایف جانبی تحلیل گر لغوی تشخیص خطاهای لغوی و حذف مواردی مانند جداکننده های توکن ها و توضیحات (Comments) از متن برنامه است. برنامه تحلیل گر لغوی در این تمرین بایستی موارد زیر را شناسایی کند:

- کلمات کلیدی زبان C (فهرست کلمات کلیدی را می توانید از [اینجا](#) مشاهده نمایید).
- شناسه ها
- ثابت های رشته ای (String Literals)
- اعداد صحیح
- اعداد ممیز شناور
- عملگرهای ریاضی و منطقی متداول زبان C
- توضیحات تک خطی
- توضیحات چندخطی

همچنین علائم جدا کننده شامل `\t`، `\n`، `\r`، `white_space`، `{`، `}`، `[`، `]`، `،` و (هستند که برخی از آنها خود Token محسوب شده و برخی دیگر برای جداسازی استفاده می شوند. با توجه به توضیحات فوق، بندهای خواسته شده را انجام دهید.

الف) یک ماشین حالت متناهی (Finite State Automata)، برای شناسایی هر یک از موارد فوق طراحی کنید.

ب) با تجمیع نمودارهای قسمت الف) در یک نمودار، نمودار حالت-گذر (State-Transition Diagram) کلی تحلیل گر لغوی زبان C را به دست آورید.

پ) کد مربوط به تحلیل گر لغوی طراحی شده در قسمت ب) را پیاده سازی نمایید.

ت) عبارت منظم (Regular Expression) مربوط به هر یک از ماشین های مربوط در قسمت الف) را بنویسید.

ث) با استفاده از پاسخ های قسمت ت)، تحلیل گر لغوی زبان C را در محیط مولد Flex پیاده سازی کنید.

ج) برنامه های قسمت های پ) و ث) را با یکدیگر مقایسه کرده، نتایج و برداشت های خود را بنویسید.



نکات مهم در خصوص تحویل پاسخ تمرین ها:

- ۱- کلیه تمرین های کلاس حل تمرین با شماره های متوالی (hw01, hw02 و غیره) در کانال تلگرامی درس به نشانی http://t.me/compiler_iust منتشر می شوند و تنها راه اطلاع از تمرین ها کانال مذکور خواهد بود. مسئولیت عدم اطلاع از اخبار کلاس حل تمرین بر عهده دانشجو است.
- ۲- تمرین ها به صورت انفرادی بایستی حل شوند. در صورت مشاهده تقلب، نمره منفی به افراد متخلف تعلق می گیرد.
- ۳- پاسخ کل پرسش های هر تمرین را در قالب یک فایل فشرده (.zip) تحت عنوان **compiler962_EN_SN** به نشانی ایمیل compiler_iust@outlook.com ارسال نمایید که **EN شماره تمرین** (hw01, hw02) و غیره) بوده و **SN شماره دانشجویی** شما است. به عنوان مثال دانشجویی با شماره دانشجویی 90723043، فایل پاسخ تمرین سری اول را با عنوان **compiler961_hw01_90723043** ارسال خواهد کرد.
- ۴- (مهم) عنوان ایمیل ارسالی دقیقاً عنوان فایل فشرده شده باشد، در غیر این صورت ایمیل شما دریافت نخواهد شد. در متن ایمیل نیز نام، نام خانوادگی و شماره دانشجویی خود را بنویسید.
- ۵- برای پرسش های حل کردنی (تشریحی) یک فایل PDF حاوی پاسخ همه پرسش ها ایجاد کنید. بهتر است پاسخ تمرین های تشریحی تایپ شده باشد اما می توانید پاسخ دست نویس خود را با استفاده از ابزارهایی مانند **CamScanner** تبدیل به فایل pdf کنید. در هر صورت پاسخ شما بایستی خوانا باشد.
- ۶- برای پرسش های برنامه نویسی کلیه فایل کدهای نوشته شده (فقط کد منبع) را به همراه تصاویری از خروجی اجرای برنامه، داخل یک فولدر مجزا قرار دهید و سپس ارسال نمایید. زبان برنامه نویسی دلخواه است، فقط کد خود را به نحو مقتضی توضیح گذاری^۱ نمایید.
- ۷- در صورتی که حجم فایل ارسالی شما زیاد بوده یا حاوی فایل اجرایی است، می توانید ابتدا آن را روی **dropbox** بارگذاری کرده و لینک آن را ایمیل نمایید. توجه کنید که لینک ارسالی شما باید تا پایان ترم فعال و ثابت باشد.
- ۸- منظور از مهلت ارسال پاسخ، حداکثر تاریخ جهت ارسال پاسخ است و ایمیل های ارسالی پس از این تاریخ بررسی نمی شوند.
- ۹- در تحویل حضوری پروژه پایانی از تمرین های طول ترم نیز سؤال پرسیده خواهد شد. همچنین لازم است دانشجویان محترم پاسخ تمرین های خود را همراه داشته باشند.
- ۱۰- برای ارتباط با دستیاران حل تمرین از ایمیل compiler_iust@outlook.com استفاده کنید. در صورت نیاز به مراجعه حضوری روزهای چهارشنبه ساعت ۱۰ تا ۱۲ به آزمایشگاه مهندسی معکوس واقع در طبقه دوم دانشکده مراجعه نمایید.

^۱comment