

*** بخش اول: پرسش ۱ مباحث مربوط به انواع خطاها در هنگام کامپایل و اجرای برنامه را پوشش می دهد.**

۱- انواع مختلفی از خطا توسط کامپایلرها در زمان کامپایل گزارش می شوند. همچنین در صورت کامپایل صحیح یک برنامه و تولید کداجرایی آن در زمان اجرا نیز برنامه ممکن است دچار خطا شود. در یک طبقه بندی کلی و سطح بالا عناوین این خطاها عبارتند از:

A. خطاهای زمان کامپایل

I. خطای لغوی

II. خطای نحوی

III. خطای معنایی

B. خطاهای زمان اجرا

I. خطای فساد حافظه (memory corruption error)

II. خطای منطقی (logic error)

قطعه کد زیر به زبان C استاندارد را در نظر بگیرید. خطاهای موجود در این قطعه کد و نوع هریک را مشخص کنید. توضیح دهید هر خطا چگونه تشخیص داده می شود؟

```
01. void main() {
02.     const float f = 10.5; const int i = 20
03.     int j = 20;
04.     int A[j], B[i];
05.     B[30] = 15;
06.     B[j] = i;
07.     B[0] = 1.5;
08.     B[1] = i;
09.     A[x] = 15;
10.     A[19.5] = 15;
11.     B[f] = -1;
12.     int re@ = 2**3 + 1;
13.     j = f + i;
14.     return 0;
15. } // end of main function
```



* بخش دوم: پرسش‌های ۲ تا ۵ مباحث مربوط گرامرهای مستقل از متن و نیز مسئله ابهام در این گرامرها، مشکلات ناشی از آن و چگونگی برطرف کردن این مشکلات را پوشش می‌دهد.

۲- در نظر بگیرید: L_{CS} مجموعه زبان‌های حساس متن، L_{CF} مجموعه زبان‌های مستقل از متن، L_{ACF} مجموعه زبان‌های مستقل از متن معین (قطعی)، L_{ambig} مجموعه زبان‌های مستقل از متن گنگ و L_{reg} مجموعه زبان‌های منظم.

الف) رابطه بین قلمرو این مجموعه زبان‌ها را در قالب نمودار ون^۱ رسم کنید.

ب) برای هر یک از نواحی موجود در نمودار ون قسمت (الف) یک زبان (همراه با گرامر) مثال بنویسید.

۳- گرامر مستقل از متن زیر با غیر پایانه S داده شده است:

$$S \rightarrow aSbS \mid bSaS \mid \varepsilon$$

الف) نشان دهید گرامر داده شده مبهم است.

ب) زبان تولید شده توسط این گرامر را مشخص کنید.

پ) جایگاه زبان منتج از این گرامر را در نمودار پرسش (۲-الف) مشخص نمایید.

۴- گرامر زیر برای دستور if در یک زبان برنامه‌نویسی نوشته شده است (غیرپایانه‌ها با حروف بزرگ شروع شده‌اند).

$$St \rightarrow \text{if Expr then } St$$

$$\mid \text{if Expr then } St \text{ else } St$$

$$\mid \text{other}$$

مشکل این گرامر به‌هنگام تحلیل نحوی **else سرگردان** است. یعنی مشخص نیست **else** در یک کد دارای ساختار **if** تودرتو متعلق به کدام **if** است.

الف) این مشکل را با یک مثال نشان دهید.

ب) گرامر را به‌گونه‌ای بازنویسی نمایید که ضمن حفظ ساختار کد، مشکل تشریح شده را نداشته باشد.

(راهنمایی: در زبان‌های برنامه‌نویسی هر **else** به نزدیک‌ترین **if** اختصاص دارد. با استفاده از این اصل گرامر جدید را بنویسید).

^۱ Venn diagram



۵- گرامر زیر برای توصیف صوری عبارت‌های ریاضی در زبان‌های برنامه‌نویسی داده شده است (غیرپایانه‌ها با حروف بزرگ شروع شده‌اند):

$$Expr \rightarrow Expr + Expr$$

$$| Expr - Expr$$

$$| Expr * Expr$$

$$| Expr / Expr$$

$$| id | num | (Expr)$$

این گرامر مبهم است. می‌توان ابهام در این گرامر را با در نظر گرفتن اولویت (تقدم و شرکت‌پذیری) برای عملگرها از بین برد. بدین منظور نیاز است تا گرامر به نحوی بازنویسی شود که اولویت‌ها را لحاظ کند.

الف) گرامر فوق را به گونه‌ای بازنویسی نمایید که عملگر + تقدم بیشتری نسبت به * و / داشته باشد. * و / هم تقدم بوده و تقدمشان از - بیشتر باشد. همچنین شرکت‌پذیری + و - از چپ به راست و شرکت‌پذیری * و / از راست به چپ باشد.

ب) بر اساس گرامر قسمت الف) مقدار عددی عبارت زیر را با تشکیل درخت تجزیه نحوی مشخص نمایید:

$$4 + 3 \times 2 - 4 / 4 = ?$$

۶- هدف از این پرسش اهمیت درک شرکت‌پذیری از راست‌به‌چپ و از چپ‌به‌راست در چگونگی تفسیر عبارت‌های ریاضی یک زبان است. گرامر زیر برای عبارت‌های ریاضی را در نظر بگیرید (حروف بزرگ غیرپایانه هستند):

$$E \rightarrow E + T | T - E | T$$

$$T \rightarrow num$$

الف) در صورت وجود درخت یا درخت‌های تجزیه نحوی را برای عبارت 2-3+4 رسم و مقدار عبارت را در هر تجزیه حساب کنید.

ب) در صورت وجود درخت یا درخت‌های تجزیه نحوی را برای عبارت 2+3-4 رسم و مقدار عبارت را در هر تجزیه حساب کنید.

پ) برداشت خود را از نتایج قسمت‌های الف) و ب) تشریح کنید.

*** بخش سوم: پرسش ۷ ساده ترین روش تحلیل نحوی پایین به بالا را برای گرامرهای مستقل از متن غیر مبهم مطرح براساس جدول تجزیه نحوی LR(0) بیان می کند.**

۷- روش LR(0) یک روش ایجاد جدول تجزیه نحوی برای تجزیه نحوی پایین به بالا به شکل shift-reduce است.

الف) برنامه ای بنویسید که یک گرامر را از ورودی دریافت و جدول تجزیه نحوی LR(0) آن را تولید کند.

ورودی*: گرامر افزوده برنامه در قالب یک فایل است که قانون $S \rightarrow S' \$$ هم در خط اول آن اضافه شده است. غیرپایانه ها با حروف بزرگ الفبای لاتین مشخص می شوند و سایر نمادها پایانه هستند. در نسخه اول برنامه فرض کنید هر غیر پایانه و هر پایانه یک نشان ساده و به طول یک هستند.

خروجی: جدول تجزیه نحوی LR(0) در قالب یک آرایه دوبعدی (ماتریس)

ب) برنامه قسمت (الف) را تکمیل کنید طوری که تحلیل نحوی یک رشته ورودی را بر اساس جدول تجزیه تولید شده انجام دهد و پذیرش یا عدم پذیرش رشته را مشخص کند.

ورودی: یک رشته از الفبای زبان.

خروجی: جدول تجزیه شامل وضعیت پشته تجزیه، ورودی و عمل انجام شده و نیز نتیجه پذیرش یا

رد رشته ورودی.

* یک نمونه ورودی و خروجی در اختیار شما قرار می گیرد.



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 **compiler962_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^۲ 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.