به نام خدا

تمرین سری اول کامپایلر سوالات تئوری (مهلت ارسال: چهارشنبه ۸ آبان)

۱- تفاوت مفسر و کامپایلر چیست و هر یک چه مزایا و معایبی دارد؟

۲- تفاوت اصلی کد میانی و کد نهایی تولید شده در طی مراحل یک کامپایلر را بیان کرده و مزیت ترجمه به کد میانی پیش از ترجمه به کد ماشین را توضیح دهید.

۳- هر یک از خطاهای زیر در چه مرحله از کامپایل و توسط کدام بخش تولید می شود؟ دلیل پاسخ خود را توضیح دهید.

a. استفاده از = به جای == در شرط های زبان C
b. بیشتر بودن اندیس های یک آرایه از ابعاد تعریف شده برای آن آرایه
c. همخوان نبودن نوع متغیرهای a و b در عبارت a+b
d. استفاده از کلمه whil به جای while در زبان C
e. نگذاشتن ; در انتهای یک دستور در زبان C

۴- زیردنباله یک رشته مانند S عبارت است از یک دنباله شامل صفر یا تعداد بیشتر کاراکتر از S به همان ترتیبی که در S ظاهر می‌شوند مثلا رشته " "backرا در نظر بگیرید عبارات "",
" "back","ack","bkزیردنباله ی S می‌باشند اما " "bb","kb زیردنباله نیستد.
زیررشته‌های S شامل یک زیردنباله پیوسته از S که شامل صفر یا تعداد بیشتری کاراکتر از S می‌باشند.
فرض کنید رشته‌ای مانند S به طول N را داریم:
الف) تعداد زیردنباله‌های رشته S را بدست آورید.
ب) تعداد زیررشته‌های رشته S را بدست آورید.

۵- عبارت منظم معادل هر یک از عبارات زیر را بنویسید.
الف)همه رشته‌هایی باینری که با صفر شروع و با صفر خاتمه یابد.
ب)همه رشته‌های باینری که با یک شروع نمی‌شوند.
ج)همه رشته‌های باینری که یک دقیقا سه بار در آنها تکرار شده.
ه)همه رشته‌های باینری که با ۰۰۰و۰۰۱و۰۱۰و۰۱۱ پایان می‌یابند.

۶- در زیر الگوهای توکن های زبان برنامه نویسی --C را مشاهده می کنید. برای هر یک از الگوهای با روش هایی که در کلاس آموخته اید یک DFA ایجاد کنید.

a. یک عدد صحیح:

Integer: [+, -][0-9]+

b. یک عدد اعشاری (نمایش علمی):

floatPoint: [-, +][0-9]+ "." (e, E) [+, -][0-9]+

c. شناسه متغیرها:

ID: [a-z , A-Z]+[0-9 , a-z , A-Z]*

d. فاصله ها (spacers):

Spacers: [' ', '\t', '\n']$^{+}$

e. علائم:

Operators: - ! + * / == != <= > < >= && ||

f. کلمات کلیدی:

int, char, return, read, write, break, if, else, while

DFA های ایجاد شده را به یک NFA واحد تبدیل کرده و حاصل را به DFA تبدیل کنید. سعی کنید با بهینه سازی این DFA، یک DFA حداقل ایجاد کنید.

سوالات عملی (مهلت ارسال: چهارشنبه ۸ آبان)

۷- با زبان برنامه نویسی دلخواه، DFA های ایجاد شده در سوال ۶ را پیاده سازی کرده و یک تحلیلگر لغوی برای شناخت توکن های زبان برنامه نویسی --C بسازید. برای شناسه ها و کلمات کلیدی (در صورت نیاز) جدول نمادها را طراحی و پیاده سازی کنید. در پایان اجرای برنامه باید جدول نمادها و دنباله توکن ها چاپ شود.

۸- به کمک کتابخانه re در زبان پایتون عبارات زیر را پیاده‌سازی کنید.

الف)رشته‌های که در آنها تمام حروف صدادار به ترتیب و هر کدام دقیقا یک بار تکرار شود.

ب) رشته‌های متشکل از حروف a تا f که در آن‌ها ترتیب الفبایی رعایت شود.(لزومی به وجود همه حروف نیست اما حروف موجود می‌بایست به ترتیب باشند)

ج) رشته‌های باینری شامل تعداد زوج صفر و تعداد فرد یک.

د) رشته‌های باینری که در آنها زیررشته "۰۱۱" نباشد.

ه)رشته‌های باینری که در آنها زیردنباله "۰۱۱" نباشد.

و) URL های یک وب

ز)آدرس ایمیل

بخش flex (مهلت ارسال: پنج شنبه ۱۶ آبان)

- یک برنامه lex بنویسید که آدرس یک فایل c++ را به عنوان ورودی دریافت کرده و تمام کامنت های آن را حذف نماید.

- یک برنامه lex بنویسید که آدرس یک فایل و یک کلمه را به عنوان ورودی دریافت کرده و در خط اول خروجی تعداد کل رخدادهای کلمه در فایل را نمایش داده و در خطوط بعدی خروجی در هر خط، شماره خط رخداد کلمه و خود خط چاپ شود.

- در این تمرین می خواهیم با استفاده از مفاهیم شی گرایی یک Symbol Table را طراحی نماییم.
  یک Symbol Table می تواند با استفاده از ساختمان داده های زیر طراحی شود:
  - لیست پیوندی
  - جدول hash
  - درخت

  آنچه از این Symbol Table انتظار می رود در جدول زیر قابل مشاهده است:

Operation	Function
allocate	to allocate a new empty symbol table
free	to remove all entries and free storage of symbol table
lookup	to search for a name and return pointer to its entry
insert	to insert a name in a symbol table and return a pointer to its entry
set_attribute	to associate an attribute with a given entry
get_attribute	to get an attribute associated with a given entry

---

- با استفاده از ابزار lex برنامه ای بنویسید که به عنوان یک ماشین حساب ساده عمل نماید.
  ورودی و خروجی های نمونه به صورت زیر می باشند:

```
Input :
3+3
Output :
6.0

Input :
5*4
Output :
20.0
```

- با استفاده از ابزار lex برنامه ای بنویسید که تعداد کلمات، خطوط و space های موجود در یک فایل متنی را به عنوان خروجی نمایش دهد.

- با استفاده از ابزار lex برنامه ای بنویسید که یک فایل html را به عنوان ورودی دریافت کرده و تمام تگ های html آن را استخراج نموده و در خروجی نمایش دهد.
  به نمونه ورودی و خروجی زیر توجه شود:

Input

```
<HTML>
<HEAD>
    <TITLE>Page</TITLE>
</HEAD>
<BODY>
</BODY>
</HTML>
```

Output

```
<HTML>

<HEAD>

<TITLE>
</TITLE>

</HEAD>

<BODY>

</BODY>

</HTML>
```
