**ﻟﻄﻔﺎً ﺑﻪ ﻧﮑﺎت زﯾﺮ ﺗﻮﺟﻪ ﻓﺮﻣﺎﯾﯿﺪ:**

* گزارش باید به زبان فارسی در قالب فایل WORD و PDF به همراه پوشه‌‌ای به نام code که همه در یک فایل فشرده شده‌است، باشند.
* فایل گزارش را به فرمت P1\_report\_StdNum.pdf نام‌گذاری نمایید (همانند P1\_report\_97131.(pdf|doc)) و توجه داشته باشید که ارسال تمرین بدون گزارش فاقد ارزش است.
* برای هر سوال جداگانه باید فایل کد، با یکی از زبان‌های Python، Java و C++ نوشته شود. کامنت گذاری در حد لازم نیز انجام پذیرد. فرمت نامگذاری فایل اصلی مربوط به هر بخش از تمرین متناسب با فرمت P1\_ProblemNum\_StdNum و در پوشه Code ذخیره شده باشد.
* فایل‌های کد و گزارش خود را مطابق فرمت‌های فوق آماده و در قالب یک فایل فشرده با نام P1\_StdNum.zipتهیه نمایید.
* مهلت ارسال پروژه یک 1402/16/03 می‌باشد. در ضمن تاخیر در ارسال پروژه مشمول کسر نمره خواهد شد.
* مطابق قوانین دانشگاه هر نوع کپی برداری، اشتراک کار دانشجویان و استفاده مستقیم از کدهای اینترنت غیرمجاز است.
* هرگونه سوال و مشکل در خصوص صورت پروژه را از طریق آدرس [r.ameri@aut.ac.ir](mailto:r.ameri@aut.ac.ir) پیگیری نمایید.

1. سیستمی را طراحی و پیاده سازی نمایید که مشخصات پذیرنده متناهی قطعی[[1]](#footnote-1) (DFA) را از فایل ورودی دریافت کرده می‌کند و بررسی کند که رشته‌های وارد شده توسط کاربر در DFA پذیرفته می‌شود یا پذیرفته نمی‌شود. شایان ذکر است باید رشته را از ورودی برنامه و همچنین پذیرنده را در قالب فایل DFA\_Input\_1.txt دریافت کند که ساختار این فایل در توضیحات پیاده سازی ذکر شده است. در ضمن به عنوان نمونه DFA، فایلی به همین نام در پوشه این پروژه موجود می‌باشد.
2. سیستمی را طراحی و پیاده سازی نمایید که پذیرنده متناهی غیر قطعی[[2]](#footnote-2) (NFA) را به پذیرنده متناهی قطعی (DFA) تبدیل نماید. NFA ورودی را در قالب فایل NFA\_Input\_2.txt دریافت نموده و DFA خروجی را نیز در قالب فایل DFA\_Output \_2.txt تولید نماید که ساختار این فایل‌ها در توضیحات پیاده سازی ذکر شده است. در ضمن به عنوان نمونه NFA، فایلی به نام NFA\_Input\_2.txt در پوشه این پروژه موجود می‌باشد.
3. سیستمی را طراحی و پیاده سازی نمایید که عبارت منظم[[3]](#footnote-3) را به پذیرنده متناهی غیر قطعی (NFA) تبدیل نماید. عبارت منظم ورودی را در قالب فایل RE\_Input\_3.txt دریافت نموده و NFA خروجی را در قالب فایل NFA\_Output \_2.txt تولید نماید که ساختار این فایل‌ها در توضیحات پیاده سازی ذکر شده است. در ضمن به عنوان نمونه عبارت منظم، فایلی به نام RE\_Input\_3.txt در پوشه این پروژه موجود می‌باشد.

قالب ورودی برای عبارت منظم بدین صورت می‌باشد:

* در خط اول حروف الفبا قرار دارند که با فاصله و یا space از هم جدا شده‌اند.
* در خط دوم عبارت منظم مورد نظر نوشته شده است و از ^ برای توان استفاده می‌کنیم

نمونه یک عبارت منظم:

a b  
(a+b)^\*b

**توضیحات پیاده‌سازی**

1. نیازی نیست که DFA و NFA حاصل بهینه باشد. می‌دانیم که برای هر زبان مشخص می‌توان چندین DFA و NFA داشت.
2. قالب ورودی و خروجی برای DFA و NFA بدین صورت می‌باشد:

* در خط اول حروف الفبا قرار دارند که با فاصله و یا space از هم جدا شده‌اند.
* در خط دوم حالات ماشین آمده است که با فاصله و یا space از هم جدا شده‌اند.
* در خط سوم نام حالت شروع قرار دارد.
* در خط چهارم حالات نهایی که با فاصله و یا space از هم جدا شده‌اند، آمده است.
* از خط چهارم به بعد مقادیر تابع انتقال و یا هر یال ازگراف ماشین به صورت “qi a qj” آمده است به این معنی که در این ماشین یک یال از حالت qi با برچسب a به حالت qj وجود دارد.

نمونه یک DFA:

a b  
Q0 Q1 Q2  
Q0  
Q1  
Q0 a Q1  
Q0 b Q1  
Q1 a Q2  
Q1 b Q2  
Q2 a Q2  
Q2 b Q2

1. Deterministic Finite Accepter [↑](#footnote-ref-1)
2. Nondeterministic Finite Accepter [↑](#footnote-ref-2)
3. Regular Expressions [↑](#footnote-ref-3)