# بررسی پذیرش رشته بر روی گرامر

- محدودیت زمان : ۳۵۰۰ میلی ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت

برای یک گرامر context-free ، یک method پیاده سازی کنید که به عنوان ورودی، یک شیء گرامر و یک رشته را دریافت کند و درصورتی که رشته توسط گرامر تولید شود، `Accepted` و درغیراین صورت، `Rejected` را به عنوان خروجی برگرداند.

دقت شود برای پیاده سازی این method ابتدا نیاز است تا گرامر داده شده را با توجه به مراحلی که در chapter 6_1 آموختید ساده سازی کرده ( remove nullable variables , unit-productions ) و سپس آنرا به chomsky normal form تبدیل و الگوریتم CYK را برای بررسی پذیرش رشته برروی آن پیاده سازی کنید . ( برای فهم دقیق تر الگوریتم به chapter 6_1 اسلاید 35 و chapter 6_2 مراجعه کنید )

در گرامر ورودی، متغیرها داخل <> هستند، # نشان‌دهنده‌ی nullable transition است و قواعد مختلف با | از همدیگر جدا می‌شوند.

**نکته : دقت شود نمره این فاز از پروژه در صورتی به شما تعلق خواهد گرفت که برای حل سوال دقیقا مراحلی که در اسلاید های مربوطه نوشته شده است را به درستی پیاده سازی کرده باشید . در صورت استفاده از الگوریتم های جایگزین برای حل سوال نمره ای به شما تعلق نخواهد گرفت**

## ورودی

نحوه ورودی گرفتن برنامه به این صورت است که در ابتدا یک عدد n که بیانگر تعداد متغیرهای گرامر می‌باشد، در خط اول می‌آید. سپس، در هریک از n خط بعدی، یکی از قواعد گرامر به عنوان ورودی به شما داده می‌شود. در انتها به عنوان آخرین خط ورودی، یک رشته داده می شود که باید پذیرفته شدن یا نشدن آن را توسط گرامر بررسی کنید.

*توجه: به فاصله‌ها در ورودی دقت کنید که دقیقا ورودی را به همین فرمت باید دریافت کنید.*

$1 \leq n \leq 10$

## خروجی

خروجی برنامه‌ی تنها شامل یک خط است که باید پذیرفته شدن یا نشدن رشته مربوطه را در گرامر به ترتیب با `Accepted` یا `Rejected` نمایش دهید.

## مثال

### ورودی نمونه ۱

```
3
<S> -> a<S>b | a<A> | b<B>
<A> -> a<A> | #
<B> -> b<B> | #
aaab
```

### خروجی نمونه ۱

```
Accepted
```

### ورودی نمونه ۲

```
4
<S> -> <S>a | <S>b | <A>a | <B>b
<A> -> ab<A> | <B>ca | #
<B> -> b<B> | <C>f
<C> -> a<C> | #
abbfcaba
```

### خروجی نمونه ۲

```
Rejected
```

# بررسی پذیرش رشته بر روی PDA

- محدودیت زمان: ۱۰ ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت

در این سوال باید با توجه به آنچه در chapter 7 آموختید بررسی کنید که آیا یک رشته توسط یک PDA پذیرفته خواهد شد یا خیر . دقت شود باید رفتار PDA برای پذیرفتن / نپذیرفتن یک رشته را بطور کامل پیاده سازی کنید

( بعنوان مثال استفاده از stack در حل سوال الزامیست و راه حل های جایگزین برای شبیه سازی رفتار PDA مجاز نیست )

## ورودی

خط اول : مجموعه state ها ( اولین عضو این مجموعه را initial state در نظر بگیرید )

خط دوم : مجموعه الفبای PDA

خط سوم : مجموعه الفبای stack

خط چهارم : مجموعه final state ها

خط پنجم : تعداد transition

سپس به تعداد transition ها در هر خط یک transition بصورت زیر نوشته شده است :

```
(state_1,input_symbol,pop_symbol),(push_symbol,state_2)
```

در خط آخر ورودی نیز رشته مورد نظر وارد میشود .

نکته : **lambda** را # و **initial stack symbol** را $ در نظر بگیرید .

## خروجی

در تنها خط خروجی در صورتی که رشته توسط PDA پذیرفته میشد Accepted و در غیر این صورت Rejected چاپ شود .

# مثال

## ورودی نمونه ۱

```
{q0,q1,q2}
{a,b}
{a,b}
{q2}
6
(q0,a,#),(a,q0)
(q0,b,#),(b,q0)
(q0,#,#),(#,q1)
(q1,a,a),(#,q1)
(q1,b,b),(#,q1)
(q1,#,$),($,q2)
abba
```

## خروجی نمونه ۱

```
Accepted
```

برای بررسی نحوه پذیرفته شدن رشته در PDA داده شده به chapter 7_1 اسلاید ۳۰ مراجعه کنید .

# تبدیل NPDA به CFG ( امتیازی )

- محدودیت زمان: ۱۰ ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت

در این فاز از پروژه قصد داریم با توجه به الگوریتمی که در chapter 7_2 اسلاید ۴۳ آموختید CFG معادل یک NPDA را نمایش دهید .

## ورودی

خط اول : مجموعه state ها

خط دوم : مجموعه الفبای PDA

خط سوم : مجموعه الفبای stack

خط چهارم : مجموعه final state ها

خط پنجم : تعداد transition

سپس به تعداد transition ها در هر خط یک transition بصورت زیر نوشته شده است :

```
(state_1,input_symbol,pop_symbol),(push_symbol,state_2)
```

نکته : **lambda** را # و **initial stack symbol** را $ در نظر بگیرید .

## خروجی

در خروجی CFG معادل NPDA را به فرمت دلخواه نمایش دهید .

## مثال

### ورودی نمونه ۱

```
{q0,qf}
{a,b}
{0,1}
{qf}
7
(q0,a,$),(0$,q0)
(q0,a,0),(00,q0)
(q0,a,1),(#,q0)
(q0,b,$),(1$,q0)
(q0,b,0),(#,q0)
(q0,b,1),(11,q0)
(q0,#,$),(#,qf)
```

برای فهم دقیق تر نحوه تبدیل PDA بالا به CFG معادل آن به chapter 7_2 اسلاید ۵۴ مراجعه کنید .

**توجه : PDA ورودی لزوما به فرمت درستی که برای بدست اوردن CFG معادلش می باشد به شما داده نمیشود و نیاز است modification های که در اسلاید های ۴۷ تا ۵۳ برروی PDA اعمال شده را نیز پیاده سازی کنید**