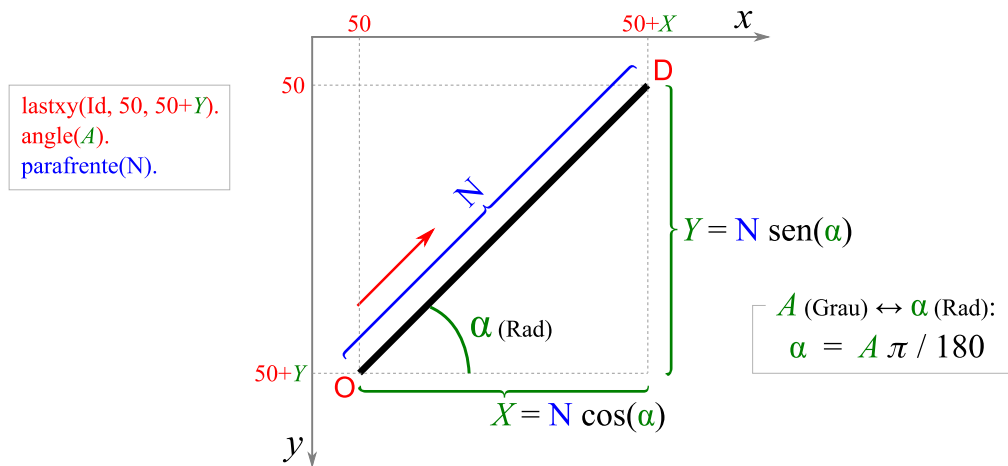


- Gramática para implementação de um subconjunto de comandos da Linguagem LOGO
  - Área de teste:  
<https://turtleacademy.com/playground/pt>
  
  - Documentação:
    - \* `pf numero` (ex.: `pf 50`)  
para frente – avança o número de passos indicados pelo numero.
    - \* `pt numero` (ex: `pt 100`)  
para trás – recua o número de passos indicados pelo numero.
    - \* `gd numero` (ex: `gd 90`)  
gira à direita – gira à direita tantos graus quanto indicados.
    - \* `ge numero` (ex: `ge 45`)  
gira à esquerda – gira à esquerda tantos graus quanto indicados.
    - \* `repita numero [<comando>]` (ex: `repita 5[pf 50 gd 45]`)  
repete comando/programa tantas vezes quanto indicadas pelo numero.
    - \* `un`  
use nada – levanta o lápis, não deixando mais traço ao se deslocar.
    - \* `ul`  
use lápis – usa o lápis, desenhando/escrevendo ao se deslocar.
    - \* `tartaruga`  
limpa os desenhos e reinicia no centro da tela (origem).
  
  - Exemplos de uso da gramática:
    - \* `?- comando([pf, '10'], []).`  
ou simplesmente  
`?- cmd("pf 10").`
    - \* `?- comando([repita, '5', '[', pf, '50', gd, '45', ''], []).`  
ou simplesmente  
`?- cmd("repita 5[pf 50 gd 45]").`
  
- Materiais:
  - `programa.pl` (versão modificada):  
<https://www.inf.ufsc.br/~alexandre.silva/courses/17s2/ine5416/exercicios/t2B/programa.pl>
  - `desenhos.pl`  
<https://www.inf.ufsc.br/~alexandre.silva/courses/17s2/ine5416/exercicios/t2B/desenhos.pl>
  - `gramatica.pl`  
<https://www.inf.ufsc.br/~alexandre.silva/courses/17s2/ine5416/exercicios/t2B/gramatica.pl>

- Entenda a gramática definida em `gramatica.pl` e implemente, em SWI-Prolog, os seguintes predicados em `programa.pl`:

1. tartaruga
2. parafrente
3. paratras
4. giradireita
5. giraesquerda
6. usenada
7. uselapis

- Dica para deslocamento “para frente” (de origem **O** para destino **D**):



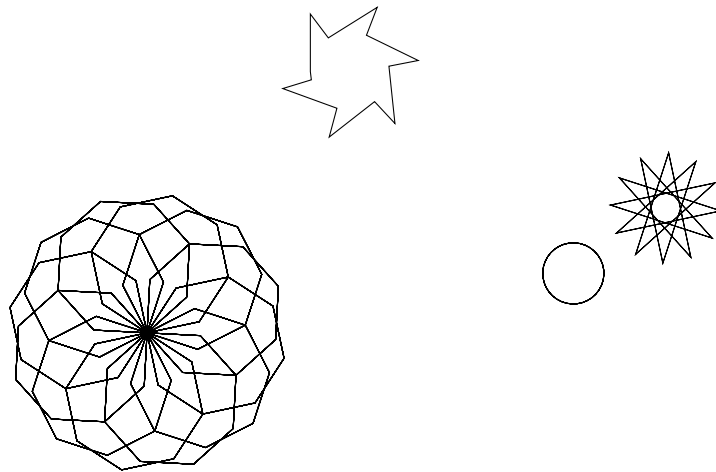
- Testes efetuados no VPL:

```
:- set_prolog_flag(double_quotes, codes).
:- initialization(testes).
```

```
testes :-
    consult('programa.pl'),
    load,
    cmd("pf 54 ge 37 pt 28 gd 95 pf 54 ge 37 pt 28 gd 95 pf 54
        ge 37 pt 28 gd 95 pf 54 ge 37 pt 28 gd 95 pf 54 ge 37
        pt 28 gd 95 pf 54 ge 37 pt 28 gd 95"), %estrela de seis pontas
    cmd("un pt 200 ge 90 pf 200 ul"),
    cmd("repita 36 [ gd 150 repita 8 [ pf 50 ge 45 ] ]"), %flor grande
    cmd("un pt 400 ul"),
    cmd("repita 72 [ ge 10 pf 5 ]"), %circunferencia
    cmd("un pt 150 ul"),
    cmd("repita 12 [ pf 100 gd 150 ]"), % estrela 'a direita
    svg.
```

---

Saída gerada para estes testes:



- **Entrega do  $T_2$ -parte B:**

- **Prazo:** dia [02nov2017 até 23h55](#)

- **Forma:** [individual](#)

- **Submissão pelo VPL-Moodle:**

1. Ao editar, salvar, executar/avaliar o **código-fonte**, no VPL, tem-se o registro de submissão
2. A execução/avaliação pode ser feita, pelo navegador, quantas vezes forem necessárias
3. Os exemplos de execução são produzidos pelo próprio VPL (os predicados dever ter os mesmos nomes indicados em cada exercício)

---

## IMPORTANTE:

Caso o 'cmd' apresente o erro a seguir:

```
?- cmd("repita 8[pf 50 gd 45]").  
ERROR: Type error: 'list' expected, found '"repita 8[pf 50 gd 45]"' (a string)
```

É preciso instruir o Prolog a interpretar aspas duplas com uma lista de códigos de caracteres. Desta forma, basta colocar, no início do arquivo 'gramatica.pl', o seguinte comando:

```
:- set_prolog_flag(double_quotes, codes).
```

Testando novamente:

```
?- cmd("repita 8[pf 50 gd 45]).  
Revisar: pf 50  
Implementar: gd 45
```