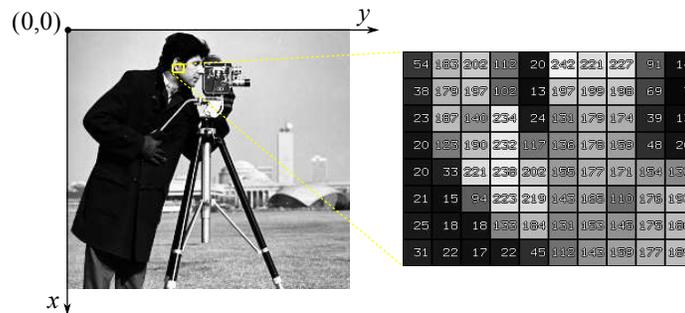


- Uma imagem em níveis de cinza é comumente representada por uma matriz de inteiros sem sinal de 8 bits, ou seja, contendo intensidades de 0 (preto) a 255 (branco):



Para tornar o desenvolvimento mais interessante em Prolog, pode-se representar a imagem como uma lista de pixels, sendo cada pixel definido por uma coordenada e uma intensidade ou (x, y, I) . O exemplo a seguir ilustra, por exemplo, o resultado de uma regra chamada `coord` que transforma uma lista de listas M (na forma de uma matriz) em uma lista de pixels S :

```
matrix([[50,10,30],
        [10,20,40]]).
```

```
?- matrix(M), coord(M,S).
```

```
M = [[50,10,30], [10,20,40]]
```

```
S = [(0,0,50), (0,1,10), (0,2,30), (1,0,10), (1,1,20), (1,2,40)]
```

Todas as demais operações são realizadas a partir desta lista de pixels ou simplesmente imagem. Outras regras podem ser consultadas e livremente utilizadas a partir da seguinte implementação inicial:

– <https://www.inf.ufsc.br/~alexandre.silva/courses/15s1/ine5416/exercicios/imagem.pl>

Pede-se a complementação da implementação, em Prolog, das seguintes **regras aplicadas sobre listas de pixels**:

1. **Limiarização (*thresholding*)**: dado um valor T como argumento, para cada intensidade $I < T$ na imagem de entrada, o pixel correspondente na imagem resultante se torna zero; para $I \geq T$, a saída se torna um (produz-se uma imagem binária).
2. **Negativo**: para cada intensidade I na imagem de entrada, produz-se $255 - I$ na imagem de saída; se a entrada for binária, a subtração passa a ser $1 - I$.
3. **Soma de constante**: dado um valor k , para cada intensidade I na imagem de entrada, produz-se $I + k$ na imagem resultante; no entanto, se $(I + k) > 255$, o valor da soma deve se tornar 255; se $k < 0$ e $(I + k) < 0$, então o valor da soma deve se tornar 0.
4. **Soma entre imagens**: cada pixel da imagem resultante é obtido pela soma dos pixels correspondentes de duas imagens de entrada com as mesmas dimensões (observar a saturação em 255).

-
5. **Detecção de pixels isolados:** um pixel de intensidade I é isolado se seus quatro vizinhos (de cima, de baixo, da esquerda e da direita) têm intensidades menores que I .
 6. **Verificação de caminho entre dois pixels:** há um caminho entre dois pixels, se há um conjunto de pixels adjacentes subsequentes (considerando os quatro vizinhos), todos com intensidades maiores do que zero, ligando estes dois pixels.
 7. *Implemente, em Prolog, uma nova operação de processamento de imagens ou reconhecimento de padrões a sua escolha.*
 8. *Escreva uma sugestão de nova operação de processamento de imagens ou reconhecimento de padrões que pudesse ser facilmente implementada em Prolog (não é preciso codificar).*

- **Entrega do T_3 –parte A:**

- **Prazo:** dia [25jun2015](#) até [23h55](#)
- **Forma:** Individual ou em grupo de [até três alunos](#)
- **Entrega pelo Moodle:**
 1. **Códigos** fontes (Prolog)
 2. **PDF** com explicações e exemplos de aplicação de cada regra e as respostas obtidas

- **Observação:** A parte *B* deste trabalho será basicamente a integração das regras implementadas aqui com uma outra linguagem de programação, de modo que seja possível a leitura/escrita de variados formatos reais de imagem, a interação com o usuário para a especificação de parâmetros, e a visualização de resultados em interface gráfica.