

# Universidad Politécnica Salesiana

Nombre: Lucy Garay

## Algoritmo de Cuadros Medios

### Ejemplo1

In [42]:

```
import random
import numpy as np
import pandas as pd
import math

xn=[]
multiplicacion=[]
log=[]
ui = []
rn=[]

numero = int(input())
print(f"xo: , {numero}")

digito=int(input())
print(f"digito: , {digito}")

iteraciones = int(input())
print(f"iteraciones: , {iteraciones}")

def centros (mul):
    cortarI=int(digito/2)
    cortarD=digito-cortarI
    mitad=math.floor(len(mul)/2)
    unir=''
    for i in range(mitad-cortarI, mitad+cortarD, 1):
        unir=unir+mul[i]
    ui.append(unir)
    return unir

def cuadrado (num):
    multi=(num*num)
    m=str(multi)
    lon=len(m)
    if (len(m)%2!=0):
        if (lon < len(m)+1):
            m=str(m).zfill(len(m)+1)
    multiplicacion.append(m)
    return m

def dividido (n):
    ceros=[int(str(num).ljust(digito+1, "0")) for num in [1]]
    res=n/ceros[0]
    rn.append(res)
    return res

for i in range(iteraciones):
    m=str(cuadrado(int(numero)))
    if (len(m)-1>digito and int(numero)>0):
        xn.append(numero)
        dividido(int(centros(m))) #RN
        numero=ui[-1]
    else:
        print('ingrese datos correctos')
        break

#print(xn)
```

```

# print(multiplicacion)
# print(ui)
# print(rn)

df=pd.DataFrame({"Xn":xn, "Xn*Xn":multiplicacion, "UI ":ui, "RN":rn})
df

```

```

8370
xo:., 8370
4
digito:., 4
30
iteraciones:., 30

```

Out[42]:

	Xn	Xn*Xn	UI	RN
0	8370	70056900	0569	0.0569
1	0569	323761	2376	0.2376
2	2376	05645376	6453	0.6453
3	6453	41641209	6412	0.6412
4	6412	41113744	1137	0.1137
5	1137	01292769	2927	0.2927
6	2927	08567329	5673	0.5673
7	5673	32182929	1829	0.1829
8	1829	03345241	3452	0.3452
9	3452	11916304	9163	0.9163
10	9163	83960569	9605	0.9605
11	9605	92256025	2560	0.2560
12	2560	06553600	5536	0.5536
13	5536	30647296	6472	0.6472
14	6472	41886784	8867	0.8867
15	8867	78623689	6236	0.6236
16	6236	38887696	8876	0.8876
17	8876	78783376	7833	0.7833
18	7833	61355889	3558	0.3558
19	3558	12659364	6593	0.6593
20	6593	43467649	4676	0.4676
21	4676	21864976	8649	0.8649
22	8649	74805201	8052	0.8052
23	8052	64834704	8347	0.8347
24	8347	69672409	6724	0.6724
25	6724	45212176	2121	0.2121
26	2121	04498641	4986	0.4986
27	4986	24860196	8601	0.8601
28	8601	73977201	9772	0.9772
29	9772	95491984	4919	0.4919

## Ejemplo2

In [43]:

```

import random
import numpy as np
import pandas as pd
import math

```

```

xn=[]
multiplicacion=[]
log=[]
ui = []
rn=[]

numero = int(input())
print(f"xo:., {numero}")

digito=int(input())
print(f"digito:., {digito}")

iteraciones = int(input())
print(f"iteraciones:., {iteraciones}")

def centros(mul):
    cortarI=int(digito/2)
    cortarD=digito-cortarI
    mitad=math.floor(len(mul)/2)
    unir=''
    for i in range(mitad-cortarI, mitad+cortarD, 1):
        unir=unir+mul[i]
    ui.append(unir)
    return unir

def cuadrado(num):
    multi=(num*num)
    m=str(multi)
    lon=len(m)
    if (len(m)%2!=0):
        if (lon < len(m)+1):
            m=str(m).zfill(len(m)+1)
    multiplicacion.append(m)
    return m

def dividido(n):
    ceros=[int(str(num).ljust(digito+1, "0")) for num in [1]]
    res=n/ceros[0]
    rn.append(res)
    return res

for i in range(iteraciones):
    m=str(cuadrado(int(numero)))
    if (len(m)-1>digito and int(numero)>0):
        xn.append(numero)
        dividido(int(centros(m))) #RN
        numero=ui[-1]
    else:
        print('ingrese datos correctos')
        break

#print(xn)
#print(multiplicacion)
#print(ui)
#print(rn)

df=pd.DataFrame({"Xn":xn, "Xn*Xn":multiplicacion, "UI ":ui, "RN":rn})
df

```

```

7543
xo:., 7543
5
digito:., 5
20
iteraciones:., 20

```

Out[43]:

	Xn	Xn*Xn	UI	RN
0	7543	56896849	89684	0.89684
1	89684	8043219856	32198	0.32198

2	32198	1036711204	67113	0.67113
3	67112	4504020544	40205	0.40205
4	40205	1616442025	64420	0.64420
5	64420	4149936400	99364	0.99364
6	99364	9873204496	32044	0.32044
7	32044	1026817936	68179	0.68179
8	68179	4648376041	83760	0.83760
9	83760	7015737600	57376	0.57376
10	57376	3292005376	20053	0.20053
11	20053	0402122809	21228	0.21228
12	21228	0450627984	06279	0.06279
13	06279	39425841	42584	0.42584
14	42584	1813397056	33970	0.33970
15	33970	1153960900	39609	0.39609
16	39609	1568872881	88728	0.88728
17	88728	7872657984	26579	0.26579
18	26579	0706443241	64432	0.64432
19	64432	4151482624	14826	0.14826

CUENCA, 13 DE J

O DEL 2020

