On défini à chaque fois le script `Scooby`. Que donnera l'exécution de `Scooby(5)` ?

A

```
def Scooby(n) :
....if n > 0 :
........print("0"*n)
........Scooby(n-1)
```

B

```
def Scooby(n) :
....if n > 0 :
........print("n"*n)
........Scooby(n)
```

C

```
def Scooby(n) :
....if n > 0 :
........Scooby(n-1)
........print("0"*n)
```

D

```
def Scooby(n) :
....if n > 0 :
........print(str(n)*n)
........Scooby(n-1)
```

E

```
def Scooby(n) :
....Scooby(n-1)
....print("0"*n)
```

F

```
def Scooby(n) :
....if n > 0 :
........print(str(n)*n)
........Scooby(n-1)
....else :
........print('Fini')
```

G

```
def Scooby(n) :
....if n > 0 :
........Scooby(n-1)
....print("0"*n)
```

H

```
def Scooby(n) :
....if n > 1 :
........Scooby(n-1)
....print("0"*n)
```

I

```
def Scooby(n) :
....print("0"*n)
....print('Fini')
....if n != 0 :
........Scooby(n-1)
```

J

On veut construire une matrice n sur k à coefficients aléatoires 0 ou 1. Trouvez les erreurs.

A

```
def Samy(n, k) :
....r = randrange(2)
....M=[[r for j in range(k)] for i in
range(n)]]
....return(M)
```

B

```
def Samy(n, k) :
....M = [ ]
....for i in range(n) :
........L=[]
........for k in range(k) :
............L.append(randrange(1))
........M.append(L)
....return(M)
```

C

```
def Samy(n, k) :
....M, L = [ ], [ ]
....for i in range(n) :
........for k in range(k) :
............L.append(randrange(2))
........M.append(L)
....return(M)
```

D

```
def Samy(n, k) :
....M = [ ]
....for i in range(n) :
........L=[randrange(2) for j in range(k)]
........M.append(L)
........return(M)
```

E

```
def Samy(n, k) :
....M = [ ]
....L = [randrange(2) for j in range(k)]
....for i in range(n) :
........M.append(L)
....return(M)
```

F

```
def Samy(n, k) :
....M = [ ]
....for i in range(n) :
........L = [randrange(2) for i in range(k)]
........M.extend(L)
....return(M)
```