

**Exercice 1**

a) Déroulez les procédures récursives suivantes pour k=6 :

**Procédure test** (↓k : entier)

Début

Si (k>0) alors test (k-1);  
Écrire (k);

fsi;Fin;

**Procédure essai** (↓k : entier)

Début

Si (k>0) alors Écrire (k);  
essai (k-1) ;

fsi ;Fin ;

**Déroulement** : Appel de la **Procédure Test**(6)

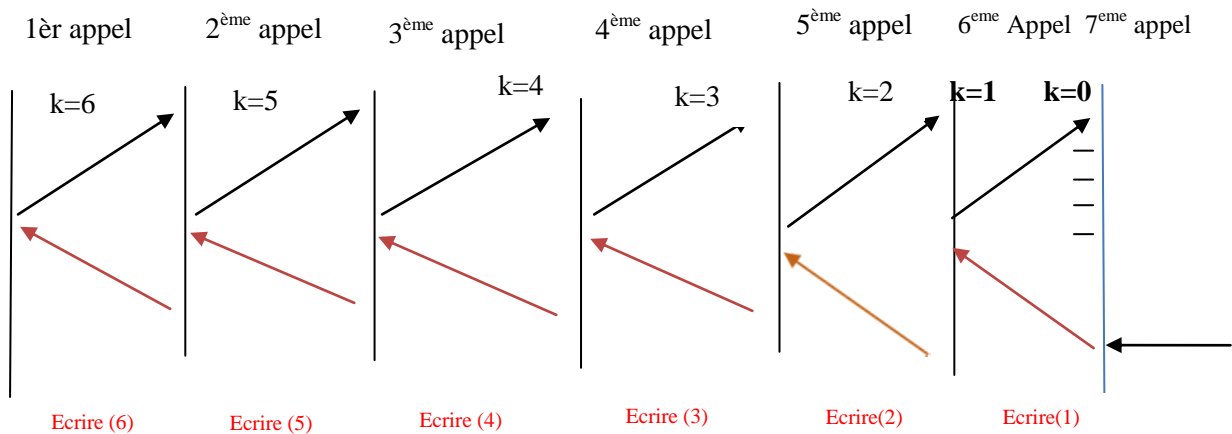


Schéma d'exécution de la procédure récursive (test(6))

**Résultat du déroulement** : On affiche le contenu lors de la **remontée** et on aura sur écran:

**1 2 3 4 5 6**

**Déroulement** : Appel de la **Procédure essai**(6)

**Schéma d'exécution** en utilisant un tableau exécution de Essai(6)

Niveau de récursivité	Appel Exécuté	Valeur retournée
0	Essai (6)	Ecrire (6)
1	Essai (5)	Ecrire (5)
2	Essai (4)	Ecrire (4)
3	Essai (3)	Ecrire (3)
4	Essai (2)	Ecrire (2)
5	Essai (1)	Ecrire (1)
6	Essai (0)	Cond terminale

D  
e  
s  
c  
e  
n  
t  
e

Point d'arrêt ↓  
Point d'appui

Schéma d'exécution de la procédure récursive (essai(6))

**Résultat du déroulement** : On affiche le contenu lors de la **descente** et on aura sur écran :

**6 5 4 3 2 1**

b) L'affichage est croissant pour la procédure **test** car la récursivité est non terminale. Par contre il est décroissant dans la procédure **essai** car la récursivité est terminale

c) **Procédure tester** ( $\downarrow n$ :entier)

**Début**

**Si** ( $n > 0$ ) **alors** tester ( $n/2$ ) ;

Écrire( $n \bmod 2$ ) ;

**fsi;fin**

**Déroulement : Appel de la Procédure tester(19)**

Niveau de récursivité	Appel Exécuté	Valeur à calculer		Valeur retournée	
0	Tester (19)	ecrire (1)	D e s c e n t e	0	R e m o n t é e
1	Tester (9)	ecrire (1)		1	
2	Tester (4)	ecrire (0)		0	
3	Tester (2)	ecrire (0)		0	
4	Tester (1)	ecrire (1)		1	
5	Tester (0)	Cond terminale Point d'arrêt Point d'appui			

Schéma d'exécution de la procédure récursive (tester(19))

**Résultat du déroulement** : On affiche le contenu lors de la **remontée** et on aura sur écran:

**10011.**

Nous remarquons que 10011 n'est que 19 en binaire.

**Déroulement : Appel de la Procédure tester(13)**

Niveau de récursivité	Appel Exécuté	Valeur à calculer		Valeur retournée	
0	Tester (13)	ecrire (1)	D e s c e n t e	1	R e m o n t é e
1	Tester (6)	ecrire (0)		0	
2	Tester (3)	ecrire (1)		1	
3	Tester (1)	ecrire (1)		1	
4	Tester (0)	Cond terminale Point d'arrêt Point d'appui			

Schéma d'exécution de la procédure récursive (tester(13))

**Résultat du déroulement** : On affiche le contenu lors de la **remontée** et on aura sur écran:

**1101.**

Nous remarquons que 1101 n'est que 13 en binaire.

La procédure tester est **non terminale**

d) Déroulez la fonction récursive suivante et dites ce qu'elle fait

```

fonction produit(n:entier, x :entier) : entier
Début
si (n > 0) alors
  écrire("avant appel", n,x);
  produit ← produit(n - 1, x) + x;
  écrire ("apres appel :", n,x);}
sinon
  | produit ←0;

```

```

fsi, fin
Début
n = 8, x = 5;
écrire (n, '*', x, '=', produit(n, x));
fin.

```

1<sup>er</sup> appel de la fonction Produit (8,5);

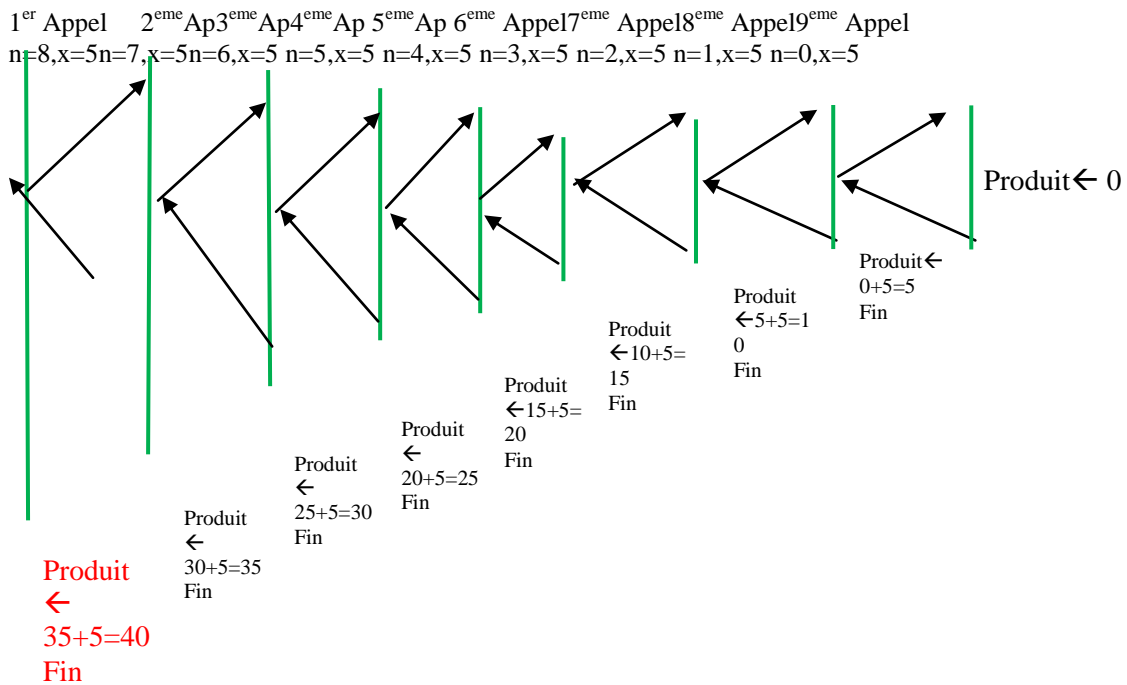
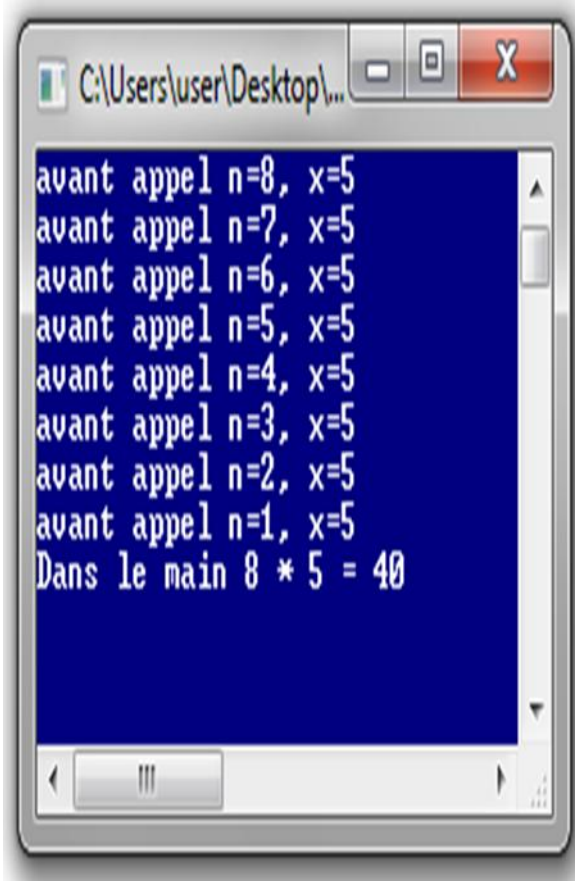


Schéma d'exécution de la fonction récursive (produit(8,5))

Le résultat de l'exécution en C de la fonction récursive (produit(8,5)) est donné dans la figure ci-dessus



**Résultat du déroulement** : On affiche le contenu lors de la **remontée** et on aura sur écran : **40**  
La fonction fait le produit de  $n*x$ .

Nous remarquons que l'instruction écrire ("apres appel :",n,x); dans la fonction produit n'est jamais exécutée.

### Exercice 2

- a) Écrire une fonction itérative qui renvoie le reste de la division euclidienne d'un entier a par un entier b en utilisant les soustractions successives.

```
Fonction q_it (a,b :entier) :entier
Début
S :entier ;
S ← 0 ;
Tque a ≤ b faire
a ← a-b ;
s ← s+1 ;
ftque
q_it ← s ;
fin
```

b) Donner la fonction récursive correspondante.

```

Fonction q_rec (a,b :entier) :entier
Début
Si a<b alors q_rec←0 ;
    Sinon
q_rec←q_rec (a-b,b)+1 ;
fsi ;fin
    
```

Exemple : a=8 et b=3

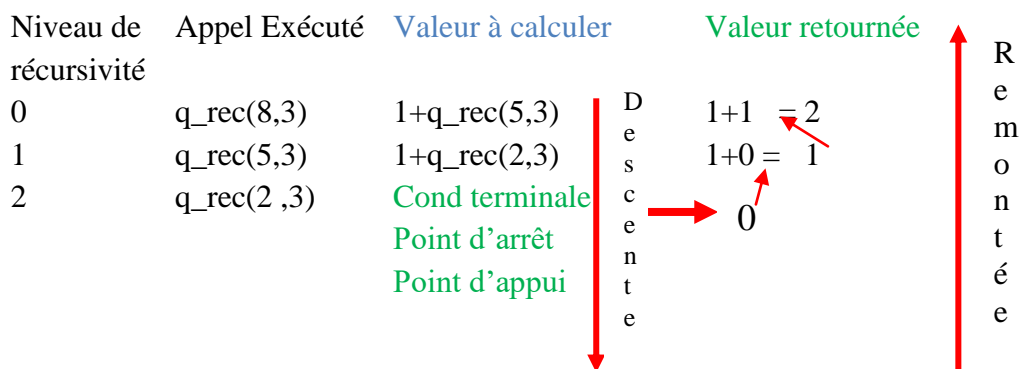


Schéma d'exécution de la fonction récursive (q\_rec(8,3))

**Résultat du déroulement** : On affiche le contenu lors de la remontée et on aura : **2**  
 Il s'agit d'une fonction non terminale, donc il y a une descente et une remontée.  
 Une fois avoir atteint le point d'arrêt, on remonte pour effectuer les calculs (flèches en rouges).