

$$U_{n+1} = U_n + 2V_n$$

$$V_{n+1} = 3U_n - V_n$$

$$U_0 = 1 \text{ et } V_0 = 2$$

```
int U (int n)
{
    if(n==0) return 1 ;
    else return (U (n-1) + 2*(V(n-1)));
}
```

Calculez U_5 et V_7

$$U_5 = 245$$

$$V_7 = 343$$

```
int V (int n)
{
    if(n==0) return 2 ;
    else return (3*U (n-1) - V(n-1));
}
```

Exo7

Type arbre : ^ cel

Type cel : eng

Info : chaîne ;

Fg : arbre ;

Fd : arbre ;

Fin ;

$$X = (24/6) + 19 * (((11+4) - 10) / 5).$$

Fonction evaluer (E/ T: arbre) : réel

Début

Cas où (T^.info) vaut

"+" : retourner (evaluer(T^.Gauche) + evaluer(T^.Droit)) ;

"-" : retourner (evaluer(T^.Gauche) - evaluer(T^.Droit)) ;

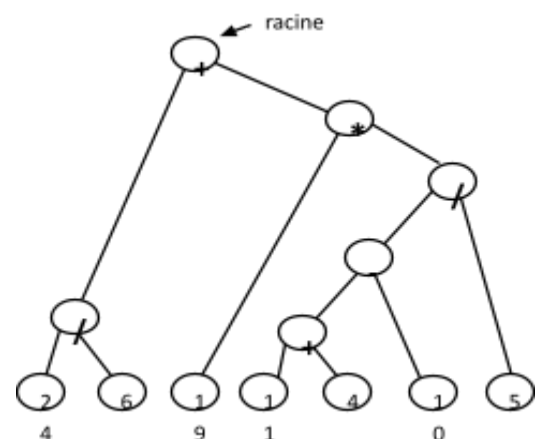
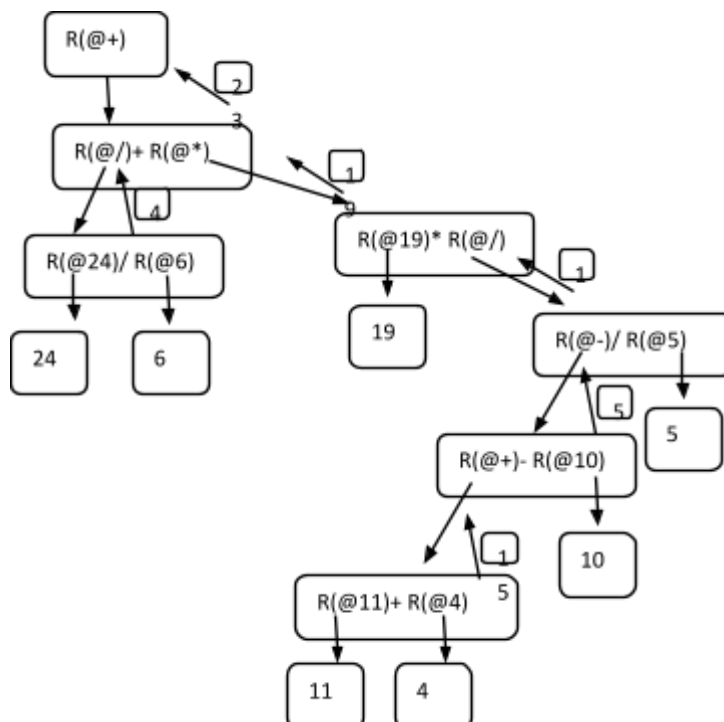
"*" : retourner (evaluer(T^.Gauche) * evaluer(T^.Droit)) ;

"/" : retourner (evaluer(T^.Gauche) / evaluer(T^.Droit)) ;

Autre : retourner (Trans_Ch_nb(T^.info)) ;

Fin cas

Fin



Fonction evaluer (E/ T: arbre) : réel

Début

Cas où (T^.info) vaut

"+" : retourner (evaluer(T^.Droit) + evaluer(T^.Gauche)) ;

"-" : retourner (evaluer(T^.Droit) - evaluer(T^.Gauche)) ;

"*" : retourner (evaluer(T^.Droit) * evaluer(T^.Gauche)) ;

"/" : retourner (evaluer(T^.Droit) / evaluer(T^.Gauche)) ;

Autre : retourner (Trans_Ch_nb(T^.info)) ;

Fin cas

Fin

X= -18.75

Exercice 1 :

Type arbre : ^ cel

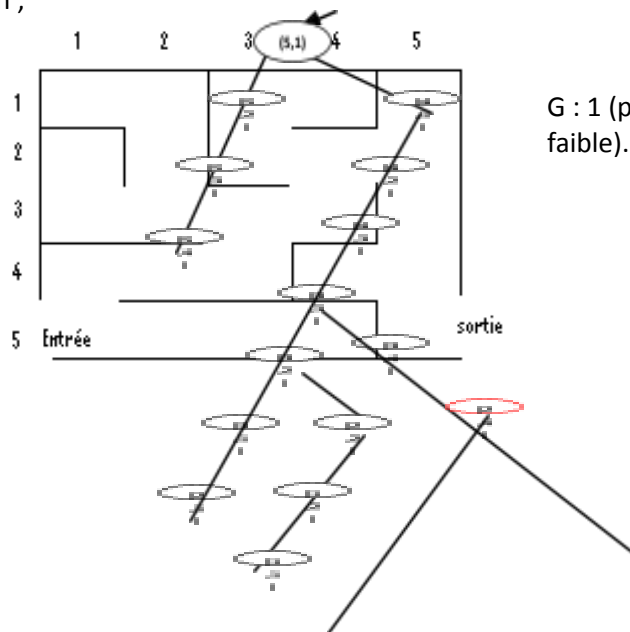
Type cel : eng

Info : Tab[2] : entier;

Fg : arbre ;

Fd : arbre ;

Fin ;



G : 1 (priorité élevée), D : 2, H : 3 et B : 4 (priorité faible).

racine

