

exo_2

1)

```
fonction existe ( arb : ABR [Persoone] , personne : Persoone ) : booleen
    si est_vide(arb) alors
        retourner faux
    finsi
    si Racine(arb).date == personne.date alors
        retourner vrais
    sinon
        si !est_vide( FG(arb) ) alors
            retourner existe( FG(arb) , personne )
        sinon
            retourner existe( FD(arb) , personne )
        finsi
    finsi
finfonction
```

2)

```
fonction aux_ieme(arb : ABR [Persoone] , i : entier , cpt :entier ) : Personne

    si !est_vide(arb) alors
        si(cpt = i )
            temp = Racine(arb)
            res = temp.nom
        sinon
            aux_ieme(FG(arb),i,cpt)
            cpt = cpt+1
            aux_ieme(FD(arb),i,cpt)
        finsi
    finsi
finfonction
```

```
fonction ieme(arb : ABR [Persoone] , i : entier) : Personne
```

```
    retourner aux_ieme(arb,i,0)
```

```
finfonction
```

3)

```
fonction aux_ajouterRang(arb : ABR [E], i : entier ):ABRR[E]
```

```
    si !est_vide(arb) alors
        aux_ajouterRang(FG(arb),i)
        i+=1
    si est_vide(FG(arb)) et est_vide(FD(arb)) alors
        FG(arb)=Cons(Racine(arb),null,null,i)
    sinon si est_vide(FG(arb)) alors
```

```

    FG(arb)=Cons(Racine(arb),null,FD(arb),i)
sinon si est_vide(FD(arb)) alors
    FG(arb)=Cons(Racine(arb),FG(arb),null,i)
sinon
    FG(arb)=Cons(Racine(arb),FG(arb),FD(arb),i)
finsi

```

```

    aux_ajouterRang(FD(arb),i)
    i+=1
si est_vide(FG(arb)) et est_vide(FD(arb)) alors
    FD(arb)=Cons(Racine(arb),null,null,i)
sinon si est_vide(FG(arb)) alors
    FD(arb)=Cons(Racine(arb),null,FD(arb),i)
sinon si est_vide(FD(arb)) alors
    FD(arb)=Cons(Racine(arb),FG(arb),null,i)
sinon
    FD(arb)=Cons(Racine(arb),FG(arb),FD(arb),i)
finsi

```

finfonction

fonction ajouterRang(arb : ABR [E]) : ABRR[E]

```

    retourner aux_ajouterRang(arb, 0)

```

finfonction

4)

fonction taille (A:ABRR[E]) : entier

```

    si ! est_vide(A) alors
        taille(FD(A) )
    sinon
        retourner rang(A)
    finsi

```

finfonction

5)

fonction aux_chercher (A : ABRR[E] , date : entier , cpt : entier) :entier

```

    si !est_vide(A) alors
        si( )

```

finfonction

fonction chercher (A :ABRR[E] , date : entier) : entier

 retourner aux_chercher(A , date , 0)
fnfonction