

Exercice 10 :

1) INVERSION(Pile:p):Pile

Début

```
variables : Pile p_inv
           entier elt
p_inv<-creer_pile()
```

```
Tant que pile_vide(p)!=vrai faire
    elt<-depiler(p)
    empiler(p_inv,elt)
```

Fin tant que

Retourner p_inv

Fin

2) SUPPRESSION_1/2_ELEMENTS(Pile:p):Pile

Début

```
variables : Pile p_aux
           entier : elt
p_aux<-creer_pile()
```

```
Tant que pile_vide(p)!=vrai faire
    elt<-depiler(p)
    empiler(p_aux,elt)
    depiler(p)
```

Fin tant que

p<-inversion(p_aux)

retourner p_aux

Fin

3) PAIR_IMPAIR(Pile:p):Pile : p_pair, Pile : p_impair

Début

```
variables:Pile : p_pair,p_impair
           entier : elt
p_pair<-creer_pile()
p_impair<-creer_pile()
```

```
Tant que pile_vide(p)!=vrai faire
    elt<-depiler(p)
    Si (elt%2) alors
        empiler(p_impair,elt)
    Sinon
        empiler(p_pair,elt)
    Fin si
```

Fin tant que

Retourner p_pair, p_impair

Fin

4) PAIR(Pile:p1):Pile p2

Début

```
variables : Pile P2,p_pair,inter
```

```

        entier : e
    p_pair<-creer_Pile()
    inter<-creer_Pile()
    P2<-creer_Pile()
    Tant que pile_vide(p1) !=vrai faire
        e<-depiler(p1)
        Si (e modulo 2=0) faire
            empiler(p_pair,e)
            empiler(inter,e)
        Sinon
            empiler(inter,e)
        Fin si
    Fin tant que
    P1<-Inversion(inter)
    P2<-Inversion(p_pair)
    Retourner P2 ;
Fin

```

Exercice 11 :

On définit :

```

    enregistrement file{
        p : pile
        aux : pile
    }
CREER_FILE():File
Début
    variables : File : f
    f.p=creer_Pile()
    f.aux=creer_Pile()
    Retourner f

```

Fin

FILE_VIDE(f:File):booléen

```

Début
    Si pile_vide(f.p) et pile_vide(f.aux) faire
        retourner vrai
    Sinon
        retourner faux
    Fin si

```

Fin

SUPPRIMER_FILE(f:file) :entier

```

Début
    variables : entier : elt
    f.aux<-Inverse(f.p)
    elt<-depiler(f.aux)
    f.p<-Inverse(f.aux)
    Retourner elt

```

Fin

INSERER_FILE(f:file,elt:entier)

```
Début
    empiler(f,p,elt)
Fin
```

Exercice 12 :

On définit :

```
    enregistrement pile{
        f : file
        aux : file
    }
```

CREER_PILE():pile

```
Début
    variables : Pile p
    p.f=creer_file()
    p.aux=creer_file()
    Retourner p
Fin
```

PILE_VIDE(p:Pile):booléen

```
Début
    Si file_vide(p.p) et file_vide(p.aux) faire
        retourner vrai
    Sinon
        retourner faux
    Fin si
Fin
```

SUPPRIMER_PILE(p:Pile):entier

```
Début
    variables : entier : elt,x
    Tant que file_vide(p.p)!=vrai faire
        x<-supprime_file(p.p)
        Si file_vide(p.p)=faux alors
            Insere_file(f.aux,x)
        Fin Si
    Fin tant que
    elt<-x
    Tant que file_vide(p.aux)!=vrai faire
        x<-supprime_file(p.s)
        Insere_file(p.p,x)
    Fin tant que
    Retourner elt
Fin
```

INSERER_PILE(p:Pile,elt:entier)

```
Début
    insere_file(p.p,elt)
Fin
```

