

Algorithme 1 : Exercice 2

```

Const N= 100;
Var
f, f1 : fichier d'entier;
choix : entier;
Procédure initialisation(x:fichier d'entier,nomfich:chaîne de caractère, mode: chaîne de caractère);
début
Associer(x,nomfich);
si  $x = nil$  alors
┌ Ecrire('erreur d'ouverture de fichier');
└ Quitter(1);
Ouvrir(x,mode);
fin;
Procédure entree_note ;
Var f: fichier d'entier; x: entier;
Début
rep ← 0;
initialisation(f,'F.dat','écriture');
répéter
┌ ecrire('entrée une valeur entière');
┌ Lire(x);
┌ Ecrire(f,x); Ecrire("voulez vous entrée une nouvelle valeur si oui taper 1 sinon 0");
└ Lire(rep);
jusqu'à  $rep = 0$ ;
fermer(f);
fin;
Procédure moyenne();
Var f: fichier d'entier; x, s, n, :entier; m: réel;
début
initialisation(f,'F.dat','lecture');
s ← 0; n ← 0; m ← 0;
tant que  $f \neq EOF$  faire
┌ Lire(f,x);
┌ s ← s + x;
└ n ← n + 1;
si  $n \neq 0$  alors
└ m ← s / n;
fermer(f);
Ecrire('la moyenne des valeurs = ',m);
fin;
Procédure afficher_valeurs(f: fichier d'entier, nomFich: chaîne de caractère);
Var x, n :entier;
debut
n ← 1;
initialisation(f,'F.dat','lecture');
tant que  $f \neq EOF$  faire
┌ Lire(f,x);
┌ Ecrire(n,',',x);
└ n ← n + 1;
fermer(f);
fin;

```

```

Procédure Pair_impair();
Var
f, f1, f2: fichier d'entier; x :entier;
début
initialisation(f,'F.dat','lecture');
initialisation(f1,'F1.dat','écriture');
initialisation(f2,'F2.dat','écriture');
tant que f ≠ EOF faire
  Lire(f,x);
  si x mod 2 = 0 alors
    | Ecrire(f2,x);
  sinon
    | Ecrire(f1,x);
fermer(f);
fermer(f1);
fermer(f2);
Ecrire('les valeurs paires');
afficher_valeurs(f2,'F2.dat');
Ecrire('les valeurs impaires');
afficher_valeurs(f1,'F1.dat');
fin;

Procédure supprimer();
Var f, f1: fichier d'entier; x :entier;
début
initialisation(f,'F.dat','lecture');
initialisation(f1,'Fn.dat','écriture');
tant que f ≠ EOF faire
  Lire(f,x);
  si x ≠ 0 alors
    | Ecrire(f1,x);
fermer(f); fermer(f1);
remove('F.dat');
rename('Fn.dat','F.dat');
Ecrire('les valeurs de fichier après la suppression des zeros');
afficher_valeurs(f,'F.dat');
fin;

Procédure trie_tab(Var t: tableau[1..N]d'entier, n:entier);
Var i, j, indMax, Max: entier;
début
pour i ← 1 à n - 1 faire
  Max ← t[i];
  indMax ← i;
  pour j ← i + 1 à n faire
    si Max < t[j] alors
      | indMax ← j;
      | Max ← t[j];
  t[indMax] ← t[i];
  t[i] ← Max;
fin;

```

```
Procédure triee();  
Var f, f1: fichier d'entier; x, i, j :entier;  
t: tableau[1..N]d'entier;  
début  
initialisation(f,'F.dat','lecture');  
initialisation(f1,'FTri.dat','écriture');  
i ← 1;  
tant que f ≠ EOF faire  
  Lire(f,x);  
  t[i] ← x;  
  i ← i + 1;  
trree_tab(t, i);  
pour j ← 1 à i faire  
  Ecrire(f1,t[j]);  
fermer(f); fermer(f1);  
Ecrire('les valeurs triee selon l'ordre décroissant');  
afficher_valeurs(f1,'FTri.dat');  
fin;  
Début  
répéter  
  Ecrire("Exercice 2");  
  Ecrire("1- pour calculer la moyenne des valeurs");  
  Ecrire("2- pour distiguer les valeurs paires et impaires");  
  Ecrire("3- pour supprimer les valeurs nulles");  
  Ecrire("4- pour triée les valeurs dans l'ordre décroissant");  
  Ecrire("0- pour sortir du programme");  
  Ecrire("entrer votre choix");  
  Lire(choix);  
  Selon choix faire  
    cas 1 faire  
    | entree_note(); moyenne();  
    cas 2 faire  
    | pair_impair();  
    cas 3 faire  
    | supprimer();  
    cas 4 faire  
    | triee();  
jusqu'à choix = 0;  
Fin.
```
