```
//exercice 3

algorithme paire
début
    saisir(N)
    remplir(T,N)
    Nb <-- NbPairs(T,N)
    écrire('Resultat = ',Nb)
fin
```

Nouveaux Types
tab = tableau de 100 entier

T.D.O.G

Objet	Nature / Type
N	entier
T	tab
saisir ,remplir	procédure
NbPairs	fonction

```
## Sous Programme

procédure saisir(@N:entier)
début
    répéter
        écrire("Donner N =")
        lire(N)
    jusqu'à (N>0)
fin

procédure remplir(@T:tab , N:entier)
début
    pour i de 0 à N-1 faire
        répéter
            écrire("T[",i,"]=")
            lire(T[i])
        jusqu'à (T[i] > 0 )
    fin_pour
fin
```

T.D.O.L

Objet	Nature / Type
i	entier

```
fonction NbPairs(T: tab , N:entier):entier
début
    nb <-- 0
    pour i de 0 à N-1 faire
        si T[i] mod 2 = 0 alors
            nb <-- nb + 1
        fin_si
    fin_pour

    retourner (nb)
fin
```

Exercice 7 : Minimum et Maximum

```
procédure MinMax(T:tab , N,max,min:entier )
début

    pour i de 1 à N-1 faire
        si max < T[i] alors
            max <-- T[i]
        fin_si
        si min > T[i] alors
            min <-- T[i]
        fin_si
    fin_pour

    écrire("Min = ",min)
    écrire("Max = ",max)

fin
```

T.D.O.L

Objet	Nature / Type
i	entier

```
algorithme chaine_palandrome
début
  saisir_nom(ch)
  test <-- palandrome(ch)
  si test = vrai  alors
    écrire("Votre nom est palandrome",ch)
  sinon
    écrire("Votre nom n'est pas palandrome",ch)
  fin_si
fin
```

T.D.O.G

Objet	Nature / Type
ch	chaine de caractère
test	booléen
saisir_nom	procédure
palandrome	fonction

```
# Sous programme
procédure saisir_nom(@ch : chaine)
début
    répéter
        écrire("Donner votre nom :")
        lire(ch)
    jusqu'à (alpha(ch) = vrai)
fin
```

T.D.O.L

Objet	Nature / Type
alpha	fonction

```
fonction alpha(ch :chaine ):booléen
début
    test <-- vrai
    pour i de 0 à long(ch)-1 faire
        si (Non("A" <= majus(ch[i]) <="Z")) alors
            test <-- faux
        fin_si
    fin_pour
    retourner (test)
fin
```

Objet	Nature / Type
i	entier
test	booléen

```
fonction palandrome(ch:chaine):booléen
début
    ch2 <-- ""
    pour i de long(ch) -1  à 0 faire
        ch2 <-- ch2 + ch[i]
    fin_pour

    si ch = ch2 alors
        retourner vrai
    sinon
        retourner faux
    fin_si

    /// retourner ch = ch2

fin
```

```
algorithme gestion des ventes d'un magasin
début
    saisir(N) // saisir mta3 3dad layamet
    SaisirVentes(T, N) // n3amro koll case 9adech da5el fe nhar
    chiffer <-- ChiffreAffaires(T, N) // somme de chiffer 9adech da5el flous fel layamet ely 3ando N
    écrire("Chiffre d'affaires total :",chiffer)
    m <-- MeilleurJour(T, N) /// bech ytala3lek akther nhar da5el fih flous maximum fel tableau
    écrire("Meilleur jour :",m)
    perfo <-- NbJoursPerformants(T, N) /// NbJoursPerformants  moyenne= (chiffer / N) nhar hedhaka yfot moyenne bech yesama jour performent
    écrire("Nombre de jours performants :",perfo)
    test <--VentesStables(T, N) /// lazem yken  triee croissante
    écrire("Ventes stables : ",test)

fin

procédure saisir(@N:entier)
début
    répéter
        écrire("Donner le nombre de jours :")
        lire(N)
    jusqu'à (N > 0)
fin

procédure SaisirVentes(T:tab, N:entier)
début
    pour i de 0 à N-1 faire
        répéter
            écrire("Jour ",i+1," :")
            lire(T[i])
        jusqu'à (T[i] > 0)
    fin_pour
fin
```

```
procédure MeilleurJour(T:tab, N:entier)
début
    max <-- T[0]
    indice <-- 0
    pour i de 0 à N-1 faire
        si max < T[i] alors
            max <-- T[i]
            indice <-- i
        fin_si
    fin_pour
    retourner indice+1
fin
```