

**Correction de la série d'exercice N° 03***Les structures Répétitives***Exercice 1 :**

Écrire un programme qui affiche les diviseurs d'un entier positif N non nul

```
Algorithme Exercice1
Var
    n,i : entier
Début
    Ecrire("Donner un nombre entier : ")
    lire(n)
    pour i de 1 à n -1 faire
        si n mod i = 0 alors
            ecrire(i," ")
        FinSi
    FinPour
Fin
```

Exercice 2 :

Écrire un programme qui détermine si un nombre est premier ou non (Un nombre premier n'est divisible que par 1 et par lui-même)

```
Algorithme Exercice2
Var
    n,i,nbDiv : Entier
Début
    Ecrire("Donner un nombre entier : ")
    lire(n)
    nbDiv <- 0
    pour i de 1 à n faire
        si n mod i = 0 alors
            nbDiv <- nbDiv + 1
        FinSi
    FinPour
    si nbDiv = 2 alors
        ecrire(n, " est un nombre premier")
    sinon
        ecrire(n, " n'est pas un nombre premier")
    FinSi
Fin
```

```
Algorithme Exercice2V2
Var
    n,i,nbDiv : Entier
Début
    Ecrire("Donner un nombre entier : ")
```

```

lire(n)
i <- 2
tantque n mod i <> 0 et n <> 1 faire
    i <- i + 1
FinTantQue

si i = n ou n=1 alors
    ecrire(n, " est un nombre premier")
sinon
    ecrire(n, " n'est pas un nombre premier")
FinSi
Fin

```

Exercice 3 :

Écrire un programme qui demande à l'utilisateur de taper un entier N et qui calcule Nième terme de la suite défini par :

$$U_n = \begin{cases} U_0 = 1 \\ U_{n+1} = 4U_n + 6 \end{cases}$$

Algorithme Exercice3

```

Var
    n,i,U : Entier
Début
    Ecrire("Donner un nombre entier : ")
    lire(n)
    U <- 1
    pour i de 1 à n faire
        U <- 4 * U + 6 // Un+1 = 4*Un + 6
    FinPour
    Ecrire("U",n," = " , U)
Fin

```

Exercice 4 :

Écrire un programme qui demande à l'utilisateur de taper un entier N supérieur à 2, jusqu'à que la réponse convienne, puis qui calcul et affiche tous les termes de la suite de Fibonacci, inférieur ou égaux à N, la suite de Fibonacci est définie comme suite :

$$U_n = \begin{cases} U_0 = 0 \\ U_1 = 1 \\ U_n = U_{n-1} + U_{n-2} \end{cases}$$

Algorithme Exercice4

```

Var
    i,n,U0,U1, U : entier
Début
    répéter
        Ecrire("Donner un nombre entier : ")
        lire(n)
    jusqu'a n >=2

    U0 <- 0
    U1 <- 1
    pour i de 2 à n faire
        U = U0 + U1
        ecrire(U , " , ")
        U0 <- U1

```

```

        U1 <- U
    FinPour
Fin

```

Exercice 5 :

Écrire un programme qui affiche la table de multiplication pour les nombres 1 à 10

Algorithme Exercice5

Var

 i,j : entier

Début

```

    pour i de 1 à 10 faire
        pour j de 1 à 10 faire
            si j*i < 10 alors
                ecrire(" ")
            sinonsi i*j < 100 alors
                ecrire(" ")
            sinon
                ecrire(" ")
            Finsi
            ecrire( i*j," | ")
        FinPour
        ecrireln("-----")
    FinPour
    pour i de 1 à 10 faire
        ecrireln("La table de multiplication de ", i)
        pour j de 1 à 10 faire
            ecrireln( i , " x ",j, " = " , i*j)
        FinPour
    FinPour

```

Fin

Exercice 6 :

Écrire un programme qui demande un nombre N, puis calcule et affiche la somme :

$$S = \sum_{i=1}^N 10^i$$

Algorithme Exercice6

var

 i,n,S : entier

Début

```

    ecrire("Donner un nombre : ")
    Lire(n)
    S <- 0
    pour i de 0 à n faire
        S <- S + 10^ i
    FinPour
    Ecrire("La somme : " , S)

```

Fin

Exercice 7 :

Écrire un programme qui demande un nombre N, puis calcule et affiche la somme :

$$S = \sum_{i=1}^N \frac{1}{i}$$

Algorithme Exercice7

```
var
    i,n,S : entier
Début
    ecrire("Donner un nombre : ")
    Lire(n)
    S <- 0
    pour i de 0 à n faire
        S <- S + 1/i
    FinPour
    Ecrire("La somme : " , S)
Fin
```

Exercice 8 :

Écrire un programme qui demande un nombre positif non nul, et qui calcule sa factorielle

$$N! = \prod_{i=N}^1 i$$

Algorithme Exercice8

```
var
    i,n,P : entier
Début
    Ecrire("Donner N : ")
    Lire(n)
    P <- 1
    pour i de n à 1 pas -1 faire
        P <- P * i
    FinPour
    ecrire(n,"! = " , P)
Fin
```

Exercice 9 :

Écrire un programme qui demande à l'utilisateur d'entrer un entier, puis le programme compte et affiche les chiffres qui composent cet entier

Algorithme Exercice9

```
var
    N,c : entier
Début
    Ecrire("Donner un nombre : ")
    Lire(N)
    c <- 0
    tantque N <> 0 faire
        ecrire(N mod 10," , " )
        c <- c + 1
        N <- N Div 10
    FinTantQue
    ecrireln()
    Ecrire("ce nombre compte " , c , " chiffres")
Fin
```

Exercice 10 :

Écrire un programme qui demande à l'utilisateur d'entrer un entier, puis le programme trouve et affiche l'inverse de ce nombre.

```

Algorithme Exercice10
var
    N,Nb,Inv : entier
Début
    Ecrire("Donner un nombre : ")
    Lire(N)
    Inv <- 0
    Nb <- N
    Ecrire("l'inverse de ", N)
    tantque N <> 0 faire
        Inv <- Inv*10 + N mod 10
        N <- N Div 10
    FinTantQue
    Ecrireln(" est ", Inv)
    Ecrire("l'inverse de ", Nb , " est ", Inv) // N = 0
Fin

```

Exercice 11 :

Écrire le programme qui affiche la somme d'une suite d'entiers saisie par l'utilisateur se terminant par zéro.

```

Algorithme Exercice11
var
    N, S : Entier
Début
    S <- 0
    repeter
        Ecrire("Donner un nombre entier : ")
        Lire(N)
        S <- S + N
    jusqu'a N = 0
    Ecrire("La somme est ", S)
Fin

```

Exercice 12 :

Écrire un programme qui demande un nombre N puis calcule :

$$S = \begin{cases} 1 + 3 + 5 + \dots + N & \text{si } N \text{ est impair} \\ 2 + 4 + 6 + \dots + N & \text{si } N \text{ est pair} \end{cases}$$

```

Algorithme Exercice12
var
    N,i,S : entier
Début
    Ecrire("Donner un nombre : ")
    Lire(N)
    S <- 0
    si N mod 2 = 0 alors // 0+ 2 + 4 + .... + N
        pour i de 0 à N pas 2 faire
            S <- S + i
        FinPour
    sinon // 1+ 3 + 5 + .... +N

```

```

        pour i de 1 à N pas 2 faire
            S <- S + i
        FinPour
    FinSi
    // Autre méthode
    S <- 0
    pour i de N à 0 pas -2 faire
        S <- S + i
    FinPour
    ecrire("La somme est ", S)
Fin

```

Exercice 13 :

Écrire un programme qui lit un nombre N, puis affiche si ce nombre est parfait ou non : un nombre est parfait s'il est égale à la somme de ses diviseurs

Algorithme Exercice13

```

var
    i,N,S : Entier
Début
    Ecrire("Donner un nombre : ")
    Lire(N)
    S <- 0
    pour i de 1 à N - 1 faire
        si N mod i = 0 alors
            S <- S + i
        FinSi
    finpour

    Si S = N alors
        Ecrire(N, " est un nombre parfait")
    sinon
        Ecrire(N, " n'est pas un nombre parfait")
    FinSi
Fin

```

Exercice 14 :

Écrire un programme qui demande à l'utilisateur un nombre compris entre 1 et 3 jusqu'à ce que la réponse convienne

Algorithme Exercice14

```

var
    N : entier
Début
    repeter
        Ecrire("Donner un nombre entre 1 et 3 : ")
        Lire(N)
    jusqu'à N >= 1 et N <= 3
Fin

```

Exercice 15 :

Écrire un programme qui demande à l'utilisateur un nombre compris entre 1 et 3 jusqu'à ce que la réponse convienne

Exercice 16 :

Écrire un programme qui demande un nombre compris entre 10 et 20, jusqu'à ce que la réponse convienne. En cas de réponse supérieure à 20, on fera apparaître un message : "Plus petit !", et inversement, "Plus grand !" si le nombre est inférieur à 10.

Algorithme Exercice16

```
var
    N : entier
Début
    repeter
        Ecrire("Donner un nombre entre 10 et 20 : ")
        Lire(N)
        si N < 10 alors
            EcrireLN("Plus grand !")
        FinSi
        si N > 20 alors
            EcrireLN("Plus Petit !")
        FinSi
    jusqu'a N >= 10 et N <= 20
Fin
```

Exercice 17 :

Écrire un programme qui demande un nombre de départ, et qui ensuite affiche les dix nombres suivants. Par exemple, si l'utilisateur entre le nombre 17, le programme affichera les nombres de 18 à 27

Algorithme Exercice17

```
var
    i, N : entier
Début
    Ecrire("Donner un nombre entre : ")
    Lire(N)
    pour i de N+1 à N +10 faire
        Ecrire(i, " ", " ")
    FinPour
Fin
```

Exercice 18 :

Écrire un programme qui demande successivement des nombres à l'utilisateur, et qui calcule le nombre de valeurs saisies. La saisie des nombres s'arrête lorsque l'utilisateur entre zéro

Algorithme Exercice18

```
var
    N, c : entier
Début
    c <- -1
    repeter
        Ecrire("Donner un nombre : ")
        Lire(N)
        c <- c + 1
    jusqu'a N = 0
    Ecrire("vous avez entré ", c , " nombres")
Fin
```

Exercice 19 :

Écrire un programme qui demande successivement des nombres à l'utilisateur, et qui lui dise ensuite quel était le plus grand parmi ces nombres et quel était sa position. La saisie des nombres s'arrête lorsque l'utilisateur entre un

zéro.

Algorithme Exercice19

```

var
    n,M,P,i : entier
Début
    i <- 0
    repeter
        Ecrire("Donner un nombre : ")
        Lire(n)
        si i = 0 alors
            M <- n
            P <- i
        FinSi
        si n > M alors
            M <- n
            P <- i
        Finsi
        i <- i + 1
    jusqu'a n = 0
    ecrire("Le max est ", M," est sa position est ", P)
Fin

```

Algorithme Exercice19V2

```

var
    n,M,i,P : entier
Début
    Ecrire("Donner un nombre : ")
    Lire(n)
    M <- n
    i <- 1
    P <- 1
    tantque n <> 0 faire
        Ecrire("Donner un nombre : ")
        Lire(n)
        si n > M alors
            M <- n
            P <- i
        FinSi
        i <- i + 1
    FinTantque
    ecrire("Le max est ", M," est sa position est ", P)
Fin

```

Exercice 20 :

Écrire un programme qui demande successivement des nombres à l'utilisateur, et qui calcule leur moyenne. La saisie des nombres s'arrête lorsque l'utilisateur entre un zéro

Exercice 21 :

Écrire un programme qui demande successivement des nombres à l'utilisateur, et qui calcule leur moyenne. La saisie des nombres s'arrête lorsque l'utilisateur entre un zéro.

Exercice 22 :

Écrire un algorithme qui : lit d'abord une valeur, ensuite il va lire successivement 10 nombres. Enfin il va déterminer combien de fois la première valeur a été saisie (sans compter la première saisie).


```

Algorithme Exercice22
var
    N1,N2,i, C : entier
Début
    Ecrire("Donner le nombre à rechercher : ")
    Lire(N1)
    C <- 0
    pour i de 1 à 10 faire
        Ecrire("Donner le nombre : ")
        Lire(N2)
        Si N1 = N2 alors
            C <- C +1
        FinSi
    FinPour
    ecrire(N1, " est répété ", C, " fois")
Fin

```

Exercice 23 :

Écrire un programme qui lit un réel X, puis calcule sa racine carrée en utilisant la suite suivante :

$$U_n = \begin{cases} U_0 = 1 \\ U_n = \frac{1}{2}(U_{n-1} + \frac{X}{U_{n-1}}) \end{cases}$$

On arrêtera les calculs lorsqu'on obtient une précision $|U_n - U_{n-1}| < 0.001$

```

Algorithme Exercice23
Var
    x,U0,U1 : réel
    i : entier
Début
    Ecrire("Donner un réel : ")
    Lire(x)
    U0 <- 0
    U1 <- 1
    tantque abs(U1 - U0) > 0.000000001 faire
        U0 <- U1
        U1 <- (U0 + x/U0)/2
    FinTantQue
    ecrireln("La racine de ", x, " est ", U1)
    ecrireln("La racine de ", x, " est ", racine(x))
Fin

```

Exercice 24 :

Écrire un programme qui lit le numérateur et dénominateur d'une fraction. Puis la simplifier

$$\frac{15}{45} = \frac{1}{3}$$

```

Algorithme exercice24
var
    a,b,d : entier
Début
    Ecrire("Donner a : ")
    Lire(a)
    Ecrire("Donner b : ")
    Lire(b)
    si a < b alors

```

```

        d <- a
    sinon
        d <- b
    FinSi

    tantque a mod d <> 0 ou b mod d <> 0 faire
        d <- d - 1
    FinTantQue
    si b/d <> 1 alors
        Ecrire(a,"/",b, " = " , a/d , "/" , b/d)
    sinon
        Ecrire(a,"/",b, " = " , a/d )
    Finsi
Fin

```

Exercice 25 :

Affichez une figure formé d'étoiles de N lignes (N est fourni au clavier):

```

*
**
***
****
*****
*****
*****
*****

```

Algorithme Exercice25

```

Var
    N,i,j : entier
Début
    Ecrire("Entrez le nombre de lignes : ")
    Lire(N)

    pour i de 1 à N faire
        pour j de 1 à N faire
            si j <= i alors
                Ecrire("*")
            sinon
                Ecrire(" ")
            FinSi
        FinPour
        Ecrireln()
    FinPour
Fin

```

Algorithme Exercice25v2

```

Var
    N,i,j : entier
Début
    Ecrire("Entrez le nombre de lignes : ")
    Lire(N)

    pour i de 1 à 2*N - 1 faire
        pour j de 1 à N faire
            si (j <= i et i <= N) ou ( j <= 2*N - i et i > N) alors

```

```

        Ecrire("*")
    sinon
        Ecrire(" ")
    FinSi
FinPour
Ecrireln()
FinPour
Fin

```

Exercice 26 :

Affichez une figure formé d'étoiles de N lignes (N est fourni au clavier):

```

*****
*****
*****
*****
****
***
**
*

```

Algorithme Exercice26

```

Var
    N,i,j : entier
Début
    Ecrire("Entrez le nombre de lignes : ")
    Lire(N)

    pour i de N à 1 pas -1 faire
        pour j de 1 à N faire
            si j <= i alors
                Ecrire("*")
            sinon
                Ecrire(" ")
            FinSi
        FinPour
    Ecrireln()
FinPour
Fin

```

Exercice 27 :

Affichez une figure formé d'étoiles de N lignes (N est fourni au clavier):

```

*
* *
* * *
* * * *
* * *
* *
*

```

Algorithme Exercice27

```

Var
    N,i,j : entier
Début

```

```

Ecrire("Entrez le nombre de lignes : ")
Lire(N)

pour i de 1 à 2*N - 1 faire
    pour j de 1 à N faire
        si (j<=i et i<=N) ou ( j<=2*N-i et i>N) alors
            Ecrire("*")
        sinon
            Ecrire(" ")
        FinSi
    FinPour
    Ecrireln()
FinPour
Fin

```

Exercice 28 :

Affichez une figure formé d'étoiles de N lignes (N est fourni au clavier):

```

*****
*       *
*       *
*       *
*       *
*       *
*       *
*       *
*       *
*****

```

Algorithme Exercice28

Var

N,i,j : entier

Début

```

Ecrire("Entrez le nombre de lignes : ")
Lire(N)

```

```

pour i de 1 à N faire
    pour j de 1 à N faire
        si i=1 ou j=1 ou i=N ou j=n alors
            Ecrire("*")
        sinon
            Ecrire(" ")
        FinSi
    FinPour
    Ecrireln()
FinPour

```

Fin

Exercice 29 :

Affichez une figure formé d'étoiles de N lignes (N est fourni au clavier):

```

*****
**      **
* *    * *
* *    * *

```

```

*   **  *
*   **  *
*  *  *  *
* *    * *
**      **
*****

```

Algorithme Exercice29

Var

N,i,j : entier

Début

Ecrire("Entrez le nombre de lignes : ")

Lire(N)

pour i de 1 à N faire

pour j de 1 à N faire

si i=1 ou j=1 ou i=N ou j=N ou i=j ou j= N - i +1 alors

Ecrire("*")

sinon

Ecrire(" ")

FinSi

FinPour

Ecrireln()

FinPour

Fin

Exercice 30 :

Écrire un programme permettant d'afficher les pyramides de nombres suivants :

```

1
121
12321
1234321
123454321
12345654321

```

Algorithme Exercice30

Var

i,j,k,s,N : entier

Début

Ecrire("Donner N : ")

Lire(N)

pour i de 1 à 2*N pas 2 faire

K <- 1

s <- 1

pour j de 1 à i faire

Ecrire(k)

si k > i / 2 alors

s <- - 1

FinSi

k <- k + s

FinPour

ecrireln()

Finpour

Fin

Exercice 31 :

Le "jeu du nombre mystérieux" consiste à jouer contre l'ordinateur comme suit : L'ordinateur choisit, au hasard, un entier entre 1 et 100 et on doit le trouver en 7 essais au maximum grâce aux indices "C'est grand" et "C'est petit". Au moment venu on affichera "Bravo vous avez gagné !!". Si le nombre d'essais est atteint sans trouver le nombre mystérieux, le programme affichera alors "Perdu, le nombre cherché est : ", suivie du nombre à trouver. Écrire un programme pour s'amuser avec l'ordinateur.

Algorithme Exercice31

```

var
    N1,N2,t : entier
Début
    N1 <- alea(1,100) // random
    //ecrireln(N1)
    t <- 0
    repeter
        ecrire("Donner un nombre : ")
        lire(N2)
        t <- t + 1
        si N1 = N2 alors
            ecrireln("Bravo")
        sinon si t= 7 alors
            ecrireln("Game Over")
        sinon si N1 < N2 alors
            Ecrireln("Plus Petit")
        sinon
            Ecrireln("Plus grand")
        FinSi
    jusqu'a N1=N2 ou t = 7
Fin

```

Exercice 32 :

Écrire un programme qui détermine toutes les manières possibles d'obtenir un total de 15 en ajoutant trois entiers choisis entre 1 et 9.

Algorithme Exercice32

```

var
    i,j,k : entier
Début
    pour i de 1 à 9 faire
        pour j de 1 à 9 faire
            pour k de 1 à 9 faire
                si i+j+k = 15 alors
                    EcrireLn(i,"+",j,"+",k,"=",15)
                FinSi
            FinPour
        FinPour
    FinPour
Fin

```

Exercice 33 :

On se propose d'afficher un histogramme à l'aide des lettres A, B et C comme celui de l'exemple ci-dessous. Écrire un programme qui saisit le nombre de A, le nombre de B et le nombre de C puis affiche l'histogramme correspondant. Les nombres sont des entiers naturels inférieurs ou égaux à 15.

Exemple d'exécution : A =4,B = 7, C = 2

A
A
A
A B
A B
A B C
A B C

Exercice 34 :

Écrire un programme qui permet de déterminer la somme des chiffres d'un nombre entier donné
exemple : pour $N = 25418$, on aura $2+5+4+1+8 = 20$

Exercice 35 :

Écrire un programme qui cherche et affiche tous les entiers cubiques de trois chiffres.
Un entier naturel de trois chiffres est dit cubique s'il est égal à la somme des cubes de ses trois chiffres.
Exemple : 153 est cubique car $153 = 1^3 + 5^3 + 3^3$

```

Algorithme Exercice35
var
    i,j,s : entier
Début
    pour i de 100 à 999 faire
        j <- i
        s <- 0
        tantque j <> 0 faire
            s <- s + (j mod 10)^3
            j <- j div 10
        FinTantQue
        si i = s alors
            EcrireLn(i, " est nombre cubique")
        FinSi
    FinPour
Fin

```

Exercice 36 :

On donne un entier naturel N strictement positif et on définit la suite de Syracuse par :

$$\text{suite de Syracuse} : \begin{cases} S_0 = N \\ S_{n+1} = \frac{S_n}{2} & S_n \text{ pair} \\ S_{n+1} = 3S_n + 1 & S_n \text{ impair} \end{cases}$$

Écrire un programme qui fait afficher les 50 premiers termes de cette suite.

Exercice 37 :

Si nous lançons 3 dés, le total des points est compris entre 3 et 18. Quelle est la probabilité d'avoir un total de 12 ?
Écrire un programme qui répond à cette question en simulant 100 lancers successifs.

Exercice 38 :

Pour un entier naturel N donné. Écrire un programme qui fait calculer et afficher la suite :

$$S = \sum_{i=1}^N \frac{1}{i!}$$

Exercice 39 :

Deux entiers m et n sont dit amis si et seulement si la somme des diviseurs de m sauf lui-même est égale à n et la somme des diviseurs de n sauf lui-même est égale à m . Écrire un programme permettant de déterminer et d'afficher tous les nombres amis compris entre 1 et 1000.

Exercice 40 :

Écrire un programme permettant de décomposer un entier N donné ($2 < N < 100$) en produit de facteurs premiers et d'afficher N et le produit de ses facteurs trouvés.

Exemple : Si $n = 60$ alors on affiche $60 = 2*2*3*5$.

Exercice 41 :

Sachant que :

$$\sin(x) = \sum_{i=1} (-1)^i \frac{x^{2i+1}}{(2i+1)!} \text{ tel que } x \in [0; 2\pi]$$

Écrire un programme qui permet d'afficher $\sin(x)$ en utilisant la formule ci-dessus. Le calcul s'arrête quand la différence entre deux termes consécutifs devient $< 10^{-4}$. La dernière somme calculée est une valeur approchée de $\sin(x)$.

Algorithme Exercice41

```

var
    i,Fi,S : entier
    X, Xi,U0,U1 : réel
Début
    Ecrire("Donner la valeur de x : ")
    Lire(X)
    Fi <- 1
    Xi <- 1
    U0 <- 1
    U1 <- 0
    i <- 1
    S <- 1
    tantque abs(U1 -U0) > 0.00001 faire
        Fi <- Fi * i
        Xi <- Xi * X
        si i mod 2 != 0 alors
            U0 <- U1
            U1 = U1 + S*(Xi/Fi)
            S <- - S
        FinSi
        i <- i +1
    FinTantQue
    ecrireln("sin(",x,") = ", U1)
    ecrireln("sin(",x,") = ", sin(x))
Fin

```

Exercice 42 :

Un entier de n chiffres est dit bien ordonné si ses chiffres forment, de gauche à droite, une suite strictement croissante. Écrire un programme qui saisit un entier N et fait sortir tous les entiers bien ordonnés de N chiffres et le nombre total de ces entiers.

Exercice 43 :

En mathématiques, un entier N est un carré parfait s'il existe un entier k tel que $N = k^2$.

Écrire un programme permettant de déterminer et d'afficher tous les carrés parfaits compris entre 1 et 9999

Exercice 44 :

Un entier de N chiffres ($1 < N < 9$) est dit bien ordonné si ses chiffres forment, de gauche à droite, une suite strictement croissante.

Exemple : L'entier de 3 chiffres, 147 est bien ordonné car $1 < 4 < 7$

Écrire un programme qui saisit un entier N ($1 < N < 9$) et fait sortir tous les entiers bien ordonnés de n chiffres et le nombre total de ces entiers.

Exercice 45 :

Écrire un programme qui affiche tous les entiers positifs impairs inférieurs à 100, en omettant les nombres divisibles par 7. L'affichage doit être sur 5 colonnes comme l'aspect suivant :

1	3	5	9	11
13	17	19	23	25
27	29	31	33	37
...

Exercice 46 :

En mathématiques, un entier N est un carré parfait s'il existe un entier k tel que $N = k^2$.

Écrire un programme permettant de déterminer et d'afficher tous les carrés parfaits compris entre 1 et 9999

Exercice 47 :

On se propose d'écrire un programme qui permet de déterminer et d'afficher toutes les représentations sous forme de sommes d'entiers consécutifs d'un entier n donné. Exemple d'exécution pour le nombre 45 :

45 = 1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9

45 = 5 + 6 + 7 + 8 + 9 + 10

45 = 7 + 8 + 9 + 10 + 11

45 = 14 + 15 + 16

45 = 22 + 23

Algorithme Exercice47

var

N,i,j,k,S : Entier

Début

Ecrire("Donner un nombre : ")

Lire(N)

pour i de 1 à N/2 faire

s <- 0

j <- i

tantque s < N faire

s <- s + j

j <- j + 1

FintantQue

si s = N alors

ecrire(N," = ")

pour k de i à j -1 faire

si k <> j -1 alors

Ecrire(k," + ")

sinon

Ecrire(k)

FinSi

FinPour

EcrireLn()

FinSi

FinPour

Fin

Exercice 48 :

Affichez une figure formée d'étoiles de N lignes (N est fourni au clavier):

* *

```
**      **
***     ***
****    ****
*****  *****
*****  *****
*****  *****
****     ****
***      ***
**       **
*        *
```

Algorithme Butterfly

Var

 N,i,j : entier

Début

 Ecrire("Entrez le nombre de lignes : ")

 Lire(N)

 pour i de 1 à 2*N -1 faire

 pour j de 1 à 2*N -1 faire

 si (j<=i et i<=N) ou (j<=2*N-i et i>N) ou (j>=2*N-i et i<=N) ou (j>=i et i>N) Alors

 Ecrire("*")

 sinon

 Ecrire(" ")

 FinSi

 FinPour

 Ecrireln()

FinPour

Fin