PARIS DESCARTES

PROG2 - Programmation impérative

TP 08 – Les tableaux à deux dimensions

Julien Sopena

Février 2008

Exercice 1 : Carré magique

Un carré magique de taille n est un arrangement en carré de n^2 valeurs. Ces nombres sont disposés de manière à ce que leurs sommes sur chaque rangée, sur chaque colonne et sur chaque diagonale soient égales. Un carré magique est dit **normal** s'il est rempli avec les nombres entiers compris entre 1 et n^2 (inclus).

Le dessin suivant représente un carré magique de taille 5 :

15	8	1	24	17
16	14	7	5	23
22	20	13	6	4
3	21	19	12	10
9	2	25	18	11

Question 1

Formuler, par rapport àn, la valeur constante $\mathcal{S}$ des sommes des lignes, colonnes et diagonales dans un carré magique **normal**.

Puisque le carré considéré est normal, la somme des valeurs contenues dans l'ensemble de ses cases est égale à la somme des n^2 premiers entiers. Et puisqu'il y a n lignes dans le carré :

$$n \times \mathcal{S} = \sum_{i=1}^{n^2} i \quad \Longrightarrow \quad \mathcal{S} = \frac{n(n^2+1)}{2}$$

Question 2

Déclarer les types et constantes nécessaires à l'élaboration d'un carré magique.

```
const
   N_MAX = 10 ;
```

```
type
  t_indice = 1..N_MAX ;
  t_cMag = array[t_indice, t_indice] of word ;
```

Question 3

Écrire un sous-programme **afficherCarreMagique** qui permet d'afficher un carré magique.

```
procedure afficherCarreMagique(c  : t_cMag; n :t_indice);
var
   i, j : t_indice;
begin
   for i :=1 to n do
   begin
      for j :=1 to n do
         write(c[i,j] :3,'  ');
      writeln;
  end;
end;
```

Question 4

Écrire un sous-programme **sommeLigne** qui calcule la somme des valeurs contenues dans la $i$ème ligne d'un carré.

```
function sommeLigne(c : t_cMag; n : t_indice; l : t_indice) :word;
var
   j : t_indice;
   s : word;
begin
   s :=0;
   for j :=1 to n do
      s :=s+c[l,j];
   sommeLigne :=s;
end;
```

Question 5

Écrire un sous-programme **sommeColonne** qui calcule la somme des valeurs contenues dans la $i$ème colonne d'un carré.

```
function sommeColonne(c : t_cMag; n :t_indice; col : t_indice) : word;
var
   i : t_indice;
   s : word;
begin
   s :=0;
   for i :=1 to n do
      s :=s+c[i,col];
   sommeColonne :=s;
```



```
end;
```

Question 6

Écrire un sous-programme **sommeDiag1** qui calcule la somme des valeurs contenues dans la diagonale NO-SE d'un carré.

```
function sommeDiag1(c : t_cMag; n : t_indice) : word;
var
   i :t_indice;
   s :word;
begin
   s :=0;
   for i :=1 to n do
      s :=s+ c[i,i];
   sommeDiag1 :=s;
end;
```

Question 7

Écrire un sous-programme **sommeDiag2** qui calcule la somme des valeurs contenues dans la diagonale SO-NE d'un carré.

```
function sommeDiag2(c :t_cMag; n : t_indice) :word;
var
   i :t_indice;
   s :word;
begin
   s :=0;
   for i :=1 to n do
      s :=s+ c[i,n-i+1];
   sommeDiag2 :=s;
end;
```

Question 8

Écrire un sous-programme **testCarreMagique** qui teste si un carré est bien un carré magique normal.

```
function testCarreMagique(c :t_cMag; n :t_indice) :boolean;
var
   valM : word;
   valid :boolean;
   i :t_indice;
begin
   valM :=n*(sqr(n)+1) div 2;
   valid :=(sommeDiag1(c,n)=valM) and (sommeDiag2(c,n)=valM);
   i :=1;
   while (valid) and  (i<=n) do
   begin
      valid :=(sommeLigne(c,n,i)=valM) and (sommeColonne(c,n,i)=valM);
      i :=i+1;
```

```
    end;
    testCarreMagique :=valid;
end;
```

Question 9

Écrire un sous-programme **carreMagique** qui construit un carre magique normal de taille n, n impair, en plaçant les valeurs 1, 2, ..., n^2 suivant le principe suivant :

- On place la valeur 1 au milieu de la ligne 1,
- On continue en montant en diagonale vers la gauche :
  - si cela conduit à déborder en haut ou à gauche, le nombre est placé dans la dernière ligne ou la dernière colonne. Par exemple, 2 est placé dans la dernière ligne, et 23 est placé dans la dernière colonne.
  - Si on atteint une case déjà remplie, le nombre est placé en dessous du nombre précédent ; cette dernière situation se produit chaque fois qu'on vient de placer un multiple de N. Par exemple, 6 est placé sous 5 et 11 est placé sous 10.

```
function successeur (x : t_indice; n :t_indice) : t_indice ;
begin
    successeur := (x mod n) + 1;
end; function predecesseur (x : t_indice; n :t_indice) : t_indice ;
begin
    if (x = 1) then
        predecesseur := n
    else
        predecesseur := x - 1 ;
end; procedure carreMagique (var c : t_cMag; n :t_indice);
var
    i,j :t_indice;
    k :word;
begin
    for i := 1 to n do
        for j := 1 to n do
            c[i,j] :=0;
    i := 1 ;
    j := (n div 2)+1;
    for k := 1 to sqr(n) do
    begin
        c[i,j] :=k;
        if c[predecesseur(i,n),successeur(j,n)] = 0 then
        begin
            i := predecesseur(i,n);
            j := predecesseur(j,n);
        end
        else
            i := successeur(i,n);
    end
end;
```

Question 10

A l'aide de l'ensemble de ces sous-programmes, écrire un programme qui construise et teste un carré magique normal d'au moins une case.

```
var
   c : t_cMag;
   n : t_indice ; begin
   repeat
      write ('Enrer la largeur du carre magique (entre 1 et ',N_MAX,') : ');
      readln (n);
   until (1<n) and (n<N_MAX);
   carreMagique(c,n);
   afficherCarreMagique(c,n);
   writeln;
   writeln('Le carre est-il magique ? ',testCarreMagique(c,n));
   readln;
end.
```

