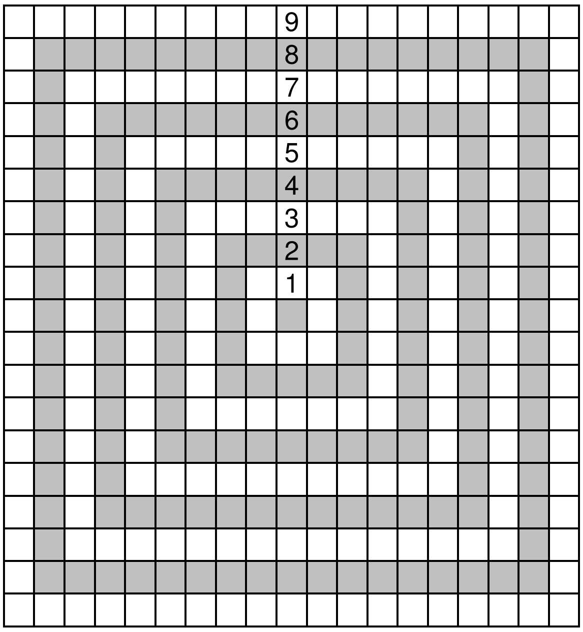
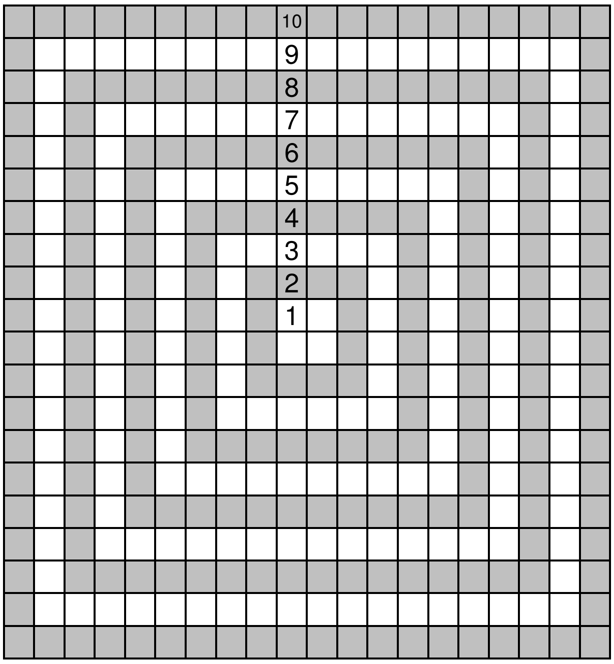
Algorithme de remplissage des carrés magiques selon la méthode des enseintes successives de Blaise PASCAL

La méthode proposée par Pascal consiste à partir d'un carré naturel et à déplacer les nombres pour que le carré devienne magique. De plus, Pascal s'est imposé une contrainte supplémentaire : chaque nombre ne doit être déplacé qu'à l'intérieur de l'enceinte à laquelle il appartient.

Il distingue les carrés de côté pair et ceux de côté impair. Les enceintes sont concentriques comme on le voit ci-dessous. L'enceinte centrale porte le n°1, l'enceinte la plus grande porte le n° $\frac{n-1}{2}$ si n est impair, $\frac{n}{2}$ si n est pair:



Carré impair de côté 19 (9 enceintes)



Carré pair de côté 20 (10 enceintes)

Dans un carré naturel, la moitié des nombres sont des *petits* nombres et l'autre moitié sont des *grands* nombres. Les petits nombres occupent la partie supérieure des carrés :

1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	32	33	34	35
36	37	38	39	40	41	42
43	44	45	46	47	48	49

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56
57	58	59	60	61	62	63	64

Les cellules ont des coordonnées comprises entre (0,0) pour la cellule en haut à gauche, à $(n-1,n-1)$ pour la cellule en bas à droite.

Les carrés impairs ont une cellule centrale de coordonnées (x_m,x_m) où $x_m = \frac{n-1}{2}$. Ci-dessus, $n = 7$. Donc la case 25 a pour coordonnées (3,3), car $\frac{7+1}{2} = 3$.

Les carrés pairs n'ont pas de cellule centrale mais un bloc de 4 cellules au centre. Dans notre exemple il s'agit de $\begin{pmatrix} 28 & 29 \\ 36 & 37 \end{pmatrix}$. La cellule en bas à droite de ce bloc a pour coordonnées $(\frac{n}{2},\frac{n}{2})$, ici (4,4) pour la case 37.

Dans les algorithmes des pages suivantes, les variables utilisées sont les suivantes :

CarréNaturel : tableau de dimension 2 contenant les nombres entiers rangés dans l'ordre naturel.
déclaration : `CarreNaturel : array[0..50,0..50]of integer;`
sa taille maximale est donc de 51×51.
Il ne sera pas visible à l'écran

CarréMagique : objet de type TStringGrid contenant le carré magique. Visible à l'écran.
déclaré sous le nom de `sgCarre`

i, j, k, n : entiers de type integer. *n* désigne le côté du carré à construire. Attention, si on veut faire une division avec un quotient entier (sans aller après la virgule), il faut utiliser l'opérateur **div** et non pas la barre oblique /. Ainsi $\frac{i-2}{2}$ sera traduit par `(i-1)div 2`

L'affectation est notée ← en langage algorithmique. Ainsi $\boxed{i \leftarrow 4}$ sera traduit en pascal par : `i:=4`

Pour traduire l'expression : $\boxed{\text{si } i \text{ est impair}}$, on écrira : `if odd(i)`

Enfin, on n'oubliera pas que le carré magique est affiché sous la forme d'une grille dont les cellules contiennent des chaînes de caractères. Il faut donc traduire les nombres entiers en chaînes de caractères avec la fonction IntToStr.

Ainsi on traduira : $\boxed{\text{CarréMagique}(x_m,\ x_m+k) \leftarrow \text{CarréNaturel}(x_m\text{-}k, x_m\text{-}k)}$

par : `sgCarre.cells[xm,xm+k]:=IntToStr(CarreNaturel[xm-k,xm-k])`

Rappels sur les structures :

Pour $k \leftarrow 1$ à x_m
…
…
FinPour
	se traduit par :	`For k:=1 to xm do`
		`begin`
		`...`
		`...`
		`end;`

Si *j* est impair alors
…
…
Sinon
…
…
FinSi
	se traduit par :	`If odd(j) then`
		`begin`
		`...`
		`end`
		`else`
		`begin`
		`...`
		`end;` (le point-virgule tient lieu de FinSi)

TantQue $j < k$ faire
…
…
…
FinTantQue
	se traduit par :	`While j<k do`
		`begin`
		`...`
		`...`
		`end;`

On trouvera, dans les pages suivantes le détail des transformations des carrés naturels en carrés magiques impairs puis pairs.

Partie gauche : les *petits* nombres sont désignés par des lettres de *a* à *h* (*j* pour les carrés pairs).
Partie droite : positions des même petits nombres après leur déplacement. Chacune des lettres doit rester dans l'enceinte où elle se trouvait dans le carré naturel.

Carré naturel impair

a									b									c
	a	e							b							f	c	
		a							b							c		
			a	e					b					f	c			
				a					b					c				
	g				a	e			b			f	c					
			g			a			b			c						
					g		a	e	b	f	c							
							g	a	b	c	h		h		h		h	
d	d	d	d	d	d	d	d	d										

Carré magique impair

d																		b
	a																b	
		d														b		
			a												b			
				d										b				
					a								b				g	
						d						b			g			
							a				b		g					
								d		b	g							
c	c	c	c	c	c	c	c	c										
									a		h		h		h		h	
								e	d	f								
									a									
						e			d			f						
									a									
				e					d					f				
									a									
		e							d							f		
									a									

à gauche : position des petits nombres à déplacer

à droite : position des mêmes petits nombres après leur déplacement.

Chaque petit nombre doit rester dans l'enceinte où il se trouve.

	b	a	b	a	b	a	b	a		a	b	a	b	a	b	a	b	
			a	b	a	b	a	b		b	a	b	a	b	a			
			b	a	b	a	b	a		a	b	a	b	a	b			
					a	b	a	b		b	a	b	a					
					b	a	b	a		a	b	a	b					
							a	b		b	a							
							b	a		a	b							

		a		a		a		a		a		a		a		a		
			a		a		a				a		a		a			
				a		a		a		a		a		a				
					a		a				a		a					
						a		a		a		a						
							a				a							
								a		a								
							b				b							
								b		b								
					b		b				b		b					
						b		b		b		b						
			b		b		b				b		b		b			
				b		b		b		b		b		b				
	b		b		b		b				b		b		b		b	

e																		f
e	e																f	f
e	e	e														f	f	f
e	e	e	e												f	f	f	f
c		e	e	e										f	f	f	d	d
c	c	c		e	e								f	f	d	d	d	d
c	c	c	c	c		e						f	d	d	d	d	d	d
c	c	c	c	c	c	c						d		d		d		d

f																		
f	f																	
f	f	f																
f	f	f	f															
c		f	f	f														
c	c	c		f	f													
c	c	c	c	c		f												
c	c	c	c	c	c	c												
												d		d		d		d
												e	d	d	d	d	d	d
													e	e	d	d	d	d
														e	e	e	d	d
															e	e	e	e
																e	e	e
																	e	e
																		e

Carré naturel pair

f	c																	d	h
	f	c	j													i	d	h	
	e	f	c													d	h		
			f	c	j									i	d	h			
			e	f	c									d	h				
					f	c	j					i	d	h					
					e	f	c					d	h						
							f	c	j	i	d	h							
							e					g		g		g		g	
b	b	b	b	b	b	b	b					a	a	a	a	a	a	a	a

Carré magique pair

																			f
	e		h													a		b	
	f	h															f		
	g		e		h									a		b			
			f	h											f				
			g		e		h					a		b					
					f	h							f						
					g		e		h	a		b							
							f												
b		b		b		b	g												
												i	a		a		a		a
												j							
								c			d			i					
							c					d		j					
						c							d			i			
					c									d		j			
				c											d			i	
			c													d		j	
		c															d		
	c																	d	

b																			d
b																		d	d
b	b																d	d	d
b	b															d	d	d	d
a	b	b	b												d	d	d	c	c
a	a	a	b	b										d	d	c	c	c	c
a	a	a	a	a	b	b							d	c	c	c	c	c	c
a	a	a	a	a	a	a							c		c		c		c

d																			
d																			
d		d																	
d	d	d																	
a	d	d		d															
a	d	a	d	d															
a	a	a	d	a		d													
a	a	a	a	a	d	a													
	a		a		a														
														c		c		c	
													c	b	c	c	c	c	c
													b		c	b	c	c	c
															b	b	c	b	c
															b		b	b	c
																	b	b	b
																	b		b
																			b
																			b

		e	f	e	f	e	f	e	f	f	e	f	e	f	e	f	e		
				e	f	e	f	e	f	f	e	f	e	f	e				
				e	f	e	f	e	f	f	e	f	e	f	e				
						e	f	e	f	f	e	f	e						
						e	f	e	f	f	e	f	e						
								e	f	f	e								
								e	f	f	e								

		e		e		e		e			e		e		e		e		
				e		e		e			e		e		e				
				e		e		e			e		e		e				
						e		e			e		e						
						e		e			e		e						
								e			e								
								e			e								
									f	f									
									f	f									
							f		f	f		f							
							f		f	f		f							
					f		f		f	f		f		f					
					f		f		f	f		f		f					
			f		f		f		f	f		f		f		f			

1. Carrés magiques impairs (de côté *n* impair)

```
Début
    Pour i de 0 à n-1                         //initialisation du carré magique et du carré naturel
        Pour j de 0 à n-1
            CarréNaturel(i , j) ← ni+j+1      //remplissage du carré naturel
            CarréMagique(i , j) ← ''          //chacune des cellules du carré magique est vidée
        FinPour
    FinPour
    x_m ← (n-1)/2                             //la cellule centrale a pour coordonnées (x_m,x_m)
    CarréMagique(x_m,x_m) ← (n^2+1)/2         //la cellule centrale contient (n^2+1)/2
    Pour k ← 1 à x_m                          //Pour chaque enceinte k
        Si k est impair,  alors CarréMagique(x_m, x_m+k) ←CarréNaturel(x_m-k,x_m-k)      //'a'
                          sinon CarréMagique(x_m-k, x_m-k) ← CarréNaturel(x_m-k,x_m-k)
        CarréMagique(x_m+k, x_m-k) ← CarréNaturel(x_m, x_m-k)                            //'b'
        CarréMagique(x_m-k, x_m) ← CarréNaturel(x_m+k,x_m-k)                             //'c'
        Si k est impair,  alors CarréMagique(x_m-k, x_m-k) ← CarréNaturel(x_m-k,x_m)     //'d'
                          sinon CarréMagique(x_m, x_m+k) ← CarréNaturel(x_m-k,x_m)
        Si k est pair,  alors
            CarréMagique(x_m-k+1, x_m+k) ← CarréNaturel(x_m-k+1,x_m-k)                   //'e'
            CarréMagique(x_m+k-1, x_m+k) ← CarréNaturel(x_m+k-1,x_m-k)                   //'f'
            CarréMagique(x_m+k, x_m-k/2) ← CarréNaturel(x_m-k,x_m-k/2)                   //'g'
            CarréMagique(x_m+k, x_m+1) ← CarréNaturel(x_m+k,x_m-1)                       //'h'
        FinSi
    FinPour
    Pour k ← 3 à x_m                          //pour chaque enceinte k à partir de la 3e
        Si k est impair   alors j ← 1
                          sinon j ← 2
        TantQue j < k faire
            si j est impair  alors            //bandes horizontales
                CarréMagique(x_m-k+j, x_m+k) ← CarréNaturel(x_m-k+j,x_m-k)               //'b'
                CarréMagique(x_m+k-j, x_m+k) ← CarréNaturel(x_m+k-j,x_m-k)
            sinon
                CarréMagique(x_m-k+j, x_m-k) ← CarréNaturel(x_m-k+j,x_m-k)               //'a'
                CarréMagique(x_m+k-j, x_m-k) ← CarréNaturel(x_m+k-j,x_m-k)
            FinSi
            j ← j+1
        FinTantQue
        Pour j ← 1 à (k-1)/2                  //quotient entier bien sûr,  bandes verticales
            CarréMagique(x_m-k, x_m-j) ← CarréNaturel(x_m-k,x_m-j)                       //'c'
            Si k est impair    alors CarréMagique(x_m+k, x_m+j) ← CarréNaturel(x_m+k,x_m-j)      //'d'
                               sinon CarréMagique(x_m+k, x_m+j+1) ← CarréNaturel(x_m+k,x_m-j-1)
            CarréMagique(x_m+k, x_m+k-j) ← CarréNaturel(x_m-k,x_m-k+j)                   //'e'
            CarréMagique(x_m-k, x_m-k+j) ← CarréNaturel(x_m+k,x_m-k+j)                   //'f'
        FinPour
    FinPour
    Pour i ← 0 à n-1                          //placement des grands nombres vis à vis des petits
        Pour j ← 0 à n-1
            Si CarréMagique(i,j) = '' alors   //si la cellule (i,j) du carré magique est vide ...
                Si (i=j) ou (i=n-1-j) alors petit ← CarréMagique(n-1-i,n-1-j)                    //coins
                Si ((i>j)et(i<n-1-j))ou((i>n-1-j)et(i<j)) alors petit ← CarréMagique(i,n-1-j)    //horiz.
                Si ((i<j)et(i<n-1-j))ou((i>n-1-j)et(i>j)) alors petit ← CarréMagique(n-1-i,j)    //vert.
                grand ← n^2+1-petit
                CarréMagique(i,j) ← grand
            FinSi
        FinPour
    FinPour
Fin
```

2. **Carrés magiques pairs (de côté *n* pair)**

```
Début
    Pour i de 0 à n-1                              //initialisation du carré magique et du carré naturel
        Pour j de 0 à n-1
            CarréNaturel(i , j) ← ni+j+1           //remplissage du carré naturel
            CarréMagique(i , j) ← ''                //chacune des cellules du carré magique est vidée
        FinPour
    FinPour
    x_m ← n/2                                      //abscisse de la cellule de droite de la première enceinte
    CarreMagique(x_m-1,x_m-1) ←CarréNaturel(x_m,x_m)             //e devient g
    CarreMagique(x_m,x_m) ←CarréNaturel(x_m-1,x_m-1)             //g devient e
    CarreMagique(x_m,x_m-1) ←CarréNaturel(x_m-1,x_m)             //f devient h
    CarreMagique(x_m-1,x_m) ←CarréNaturel(x_m,x_m-1)             //h devient f
    CarreMagique(x_m-2,x_m-2) ←CarréNaturel(x_m+1,x_m+1)         //a devient c
    CarreMagique(x_m+1,x_m+1) ←CarréNaturel(x_m-2,x_m-2)         //c devient a
    CarreMagique(x_m+1,x_m-2) ←CarréNaturel(x_m-2,x_m+1)         //b devient d
    CarreMagique(x_m-2,x_m+1) ←CarréNaturel(x_m+1,x_m-2)         //d devient b
    Pour i ← 0 à 3                                               //les autres cellules restent inchangées
        Pour j ← 0 à 3
            si CarréMagique(i,j) = '' alors CarréMagique(x_m-2+i,x_m-2+j) ← CarréNaturel(x_m-2+i,x_m-2+j)
        FinPour
    FinPour
    Pour i ← 3 à x_m
        si i est impair, alors CarréMagique(x_m-3+i,x_m-i) ← CarréNaturel(x_m-1+i,x_m-1)      //'a' (impaires)
                         sinon CarréMagique(x_m-1+i,x_m) ← CarréNaturel(x_m-1+i,x_m-1)        //'a'(paires)
        si i est impair, alors CarréMagique(x_m-1+i,x_m-i) ← CarréNaturel(x_m-i,x_m-1)        //'b'(impaires)
                         sinon CarréMagique(x_m-i,x_m-1) ← CarréNaturel(x_m-i,x_m-1)          //'b' (paires)
        CarréMagique(x_m-i+1,x_m+i-1) ← CarréNaturel(x_m-i+1,x_m-i)                           //'c'
        CarréMagique(x_m+i-2,x_m+i-1) ← CarréNaturel(x_m+i-2,x_m-i)                           //'d'
        si i est impair  alors CarréMagique(x_m-i,x_m-i) ←CarréNaturel(x_m-i,x_m-i+1)         //'e'
        si i est impair  alors CarréMagique(x_m-i,x_m-i+1) ←CarréNaturel(x_m-i,x_m-i)         //'f'(impaires)
                         sinon CarréMagique(x_m+i-1,x_m-i) ←CarréNaturel(x_m-i,x_m-i)         //'f' (paires)
        si i est impair  alors CarréMagique(x_m-i,x_m-i+2) ←CarréNaturel(x_m+i-1,x_m-2)       //'g'
        si i est impair  alors CarréMagique(x_m-i+2,x_m-i) ←CarréNaturel(x_m+i-1,x_m-i)       //'h'(impaires)
                         sinon CarréMagique(x_m-i,x_m-i) ←CarréNaturel(x_m+i-1,x_m-i)         //'h'(paires)
        si i est impair  alors CarréMagique(x_m+i-1,x_m+i-3) ←CarréNaturel(x_m+i-3,x_m-i)     //'i'
        si i est impair  alors CarréMagique(x_m+i-1,x_m+i-2) ←CarréNaturel(x_m-i+2,x_m-i)     //'j'
    FinPour
    Pour i ← 4 à x_m     //placement petits nombres restants dans les bandes verticales et horizontales
        Pour j ← 1 à (i-2)/2 (quotient entier)  //placement des bandes verticales
            si i est impair  alors CarréMagique(x_m-i,x_m-j) ←CarréNaturel(x_m-i,x_m-j-1)           //'a'(impaires)
                             sinon CarréMagique(x_m-i,x_m-j-1) ←CarréNaturel(x_m-i,x_m-j-1)         //'a'(paires)
            si i est impair  alors CarréMagique(x_m+i-1,x_m+i-j-3) ←CarréNaturel(x_m-i,x_m-i+j+1)   //'b'
                             sinon CarréMagique(x_m+i-1,x_m+i-j-1) ←CarréNaturel(x_m-i,x_m-i+j)
            si i est impair  alors CarréMagique(x_m+i-1,x_m+j-1) ←CarréNaturel(x_m+i-1,x_m-j-2)     //'c'
                             sinon CarréMagique(x_m+i-1,x_m+j) ←CarréNaturel(x_m+i-1,x_m-j-1)
            si i est impair  alors CarréMagique(x_m-i,x_m-i+j+2) ←CarréNaturel(x_m+i-1,x_m-i+j)     //'d'
                             sinon CarréMagique(x_m-i,x_m-i+j) ←CarréNaturel(x_m+i-1,x_m-i+j)
        FinPour
        si i est impair  alors k ←i-3
                         sinon k ←i-2
        Pour j ← 1 à k                          //placement des bandes horizontales
            si j est impair  alors
                    CarréMagique(x_m-j,x_m+i-1) ←CarréNaturel(x_m-j,x_m-i)              //'f gauche
                    CarréMagique(x_m+j-1,x_m+i-1) ←CarréNaturel(x_m+j-1,x_m-i)          //'f droite
                sinon
                    CarréMagique(x_m-j,x_m-i) ←CarréNaturel(x_m-j,x_m-i)                //'e'gauche
                    CarréMagique(x_m+j-1,x_m-i) ←CarréNaturel(x_m+j-1,x_m-i)            //'e'droite
            FinSi
        FinPour
    FinPour
    Pour i ← 0 à n-1                            //placement des grands nombres vis à vis des petits
        Pour j ← 0 à n-1
            Si CarréMagique(i,j) = '' alors   //si la cellule (i,j) du carré magique est vide …
                Si (i=j) ou (i=n-1-j) alors petit ← CarréMagique(n-1-i,n-1-j)                      //coins
                Si ((i>j)et(i<n-1-j))ou((i>n-1-j)et(i<j)) alors petit ← CarréMagique(i,n-1-j)      //horiz.
                Si ((i<j)et(i<n-1-j))ou((i>n-1-j)et(i>j)) alors petit ← CarréMagique(n-1-i,j)      //vert.
                grand ← n²+1-petit
                CarréMagique(i,j) ← grand
            FinSi
        FinPour
    FinPour
Fin
```