

CAS 8 No 27
MAT. N° 10



CUADRADOS
MAGICOS
DE ORDEN
IMPAR



C U A D R A D O S M A G I C O S

d e o r d e n i m p a r

ASIGNATURA: SEMINARIO

PROFESOR: VASSALLO

ESPECIALIDAD: MATEMATICA Y MATEMATICA APLICADA

AÑO: 1991

EQUIPO INTEGRADO POR:

ROXANA PERSELLO

ROXANA GUZMAN

SILVIA LEMA

CAPITULO I - NOCIONES ELEMENTALES

1.1. En qué consiste el problema?	1
1.2. Cuál es el cuadrado mágico más pequeño?	1
1.3. Determinación (a priori) de la suma de los números de una fila, columna o diagonal	1
ACTIVIDADES	3

CAPITULO II - GENERACION

2.1. Introducción	6
2.2. Procedimiento de Bachet	6
ACTIVIDADES	9

CAPITULO III - ROTACIONES Y REFLEXIONES

3.1. Introducción	18
3.2. Rotaciones	18
3.3. Reflexiones	19
ACTIVIDADES	21
BIBLIOGRAFIA	30

PROLOGO

La primera mención acerca de un cuadrado mágico se encuentra en un antiguo libro oriental que data de los años 4.000 - 5.000 antes de nuestra era.

Los cuadrados mágicos eran más conocidos en la antigua India. La afición a los cuadrados mágicos pasó de la India a los pueblos árabes, los cuales les atribuían a estas combinaciones numéricas propiedades misteriosas.

En Europa occidental los cuadrados mágicos eran en la Edad Media patrimonio de los representantes de las pseudociencias, los alquimistas y los astrólogos. De estas viejas ideas supersticiosas es de donde los cuadrados numéricos recibieron su denominación de "mágicos", es decir, pertenecientes a la magia, tan extraña a las matemáticas.

Los astrólogos y los alquimistas creían que una tablilla con la representación de un cuadrado mágico era capaz de salvar de la desgracia a la persona que lo llevaba como talismán.

La composición de los cuadrados mágicos no es tan solo una distracción. Su teoría fue elaborada por muchos matemáticos eminentes. En el presente trabajo nos dedicamos al estudio de los cuadrados mágicos de orden impar.

Roxana Persello

Roxana Guzman

Silvia Lema

CAPITULO I

NOCIONES ELEMENTALES

1.1. En qué consiste el problema?

El problema consiste en buscar una disposición tal de los números sucesivos (empezando por el 1), en las casillas de un cuadrado cuadrado, que las sumas de los números en todas las filas y columnas y siguiendo las dos diagonales del cuadrado, sean iguales.

1.2. Cual es el cuadrado mágico mas pequeño?

El cuadrado mágico más pequeño es el de 9 casillas.

EJEMPLO

4	3	8
9	5	1
2	7	6

1.3. Determinación (a priori) de la suma de los números de una fila, columna o diagonal.

Siendo C la constante de un cuadrado de orden 3; $X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8, X_9$, las componentes del cuadrado y S la suma de todas las componentes.

Para que el cuadrado sea mágico, deben satisfacerse las ocho

relaciones siguientes:

$$X_1 + X_2 + X_3 = C \quad (1)$$

$$X_4 + X_5 + X_6 = C \quad (2)$$

$$X_7 + X_8 + X_9 = C \quad (3)$$

$$X_1 + X_4 + X_7 = C \quad (4)$$

$$X_2 + X_5 + X_8 = C \quad (5)$$

$$X_3 + X_6 + X_9 = C \quad (6)$$

$$X_4 + X_5 + X_9 = C \quad (7)$$

$$X_3 + X_5 + X_7 = C \quad (8)$$

Sumando (1), (2) y (3) obtenemos:

$$X_1 + X_2 + X_3 + X_4 + X_5 + X_6 + X_7 + X_8 + X_9 = 3C$$

de donde $C = S/3$.

Con la suma de (2), (5), (7) y (8), obtenemos:

$$X_1 + X_2 + X_3 + X_4 + 4X_5 + X_6 + X_7 + X_8 + X_9 = 4C$$

Por lo tanto $3C + 3X_5 = 4C$ y de ahí se deduce que $X_5 = C/3$.

En un cuadrado mágico de orden tres, la constante es pues igual al tercio de la suma de sus números y el número central es a su vez igual al tercio de la constante.

En general:

$$C = \frac{1 + 2 + 3 + \dots + n}{\text{num. de filas}}$$

ACTIVIDADES

1.1. Efectuar un programa que:

1.1.1. Permita el ingreso de un cuadrado.

1.1.2. Escriba dicho cuadrado.

1.1.3. Indique si el cuadrado es mágico o no.

```
Running
INGRESE LAS COMPONENTES DEL CUADRADO
4
3
8
9
5
1
2
7
6
      4   3   8
      9   5   1
      2   7   6
EL CUADRADO ES MAGICO
```

```
Running
INGRESE LAS COMPONENTES DEL CUADRADO
1
2
3
4
5
6
7
8
9
      1   2   3
      4   5   6
      7   8   9
EL CUADRADO NO ES MAGICO
```

```

program verificacion;
const n=3;
type cua=array[1..n,1..n] of real;
var
  matriz:cua;
  ok:boolean;
(*#####*)
PROCEDURE ENTRADATO(var a:cua);
var
  f,c:integer;
begin
  writeln('INGRESE LAS COMPONENTES DEL CUADRADO');
  for f:=1 to n do
    for c:=1 to n do
      readln(a[f,c]);
end;
(*#####*)
FUNCTION ESMAGIC (a:cua):boolean;
var
  f,c:integer;
  suma,total:real;
  ok:boolean;
begin
  total:=n*(n*n + 1)/2;
  ok:=true;
  f:=1;
  while ok and (f<=n) do
    begin
      suma:=0;
      for c:=1 to n do
        suma:= suma + a[f,c];
        ok:= suma=total;
        f:=f+1;
      end;
      if ok then
        begin
          c:=1;
          while ok and (c<=n)do
            begin
              suma:=0;
              for f:=1 to n do
                suma:=suma + a[f,c];
                ok:= suma=total;
                c:=c+1;
              end;
            end;
          end;
          if ok then
            begin
              f:=1;
              c:=1;
              while ok and (f<=n) and (c<=n) do
                begin
                  suma:=0;
                  for f:=1 to n do
                    for c:=1 to n do
                      if f=c then

```



```

        suma:=suma + a[f,c];
        ok:=suma=total;
        f:=f+1;
        c:=c+1;
    end;
end;
if ok then
begin
    f:=1;
    c:=1;
    while ok and (f<=n) and (c<=n) do
    begin
        suma:=0;
        for f:=1 to n do
            for c:=1 to n do
                if (f+c = n+1) then
                    suma:=suma + a[f,c];
                    ok:=suma=total;
                    f:=f+1;
                    c:=c+1;
            end;
        end;
        esmagic:=ok;
    end;
    (*****
PROCEDURE SALIDA (a:cua);
var
    f,c:integer;
begin
    for f:=1 to n do
        begin
            for c:=1 to n do
                write (a[f,c]:4:0);
                writeln;
            end;
        end;
    end;
    (*****
begin(*programa principal*);
entradato(matriz);
salida(matriz);
if esmagic (matriz) then
    writeln('EL CUADRADO ES MAGICO')
else
    writeln('EL CUADRADO NO ES MAGICO');
end.

```

CAPITULO II

GENERACION

2.1. Introducción

Existen diversos procedimientos para generar cuadrados mágicos de orden impar. Entre ellos podemos destacar el procedimiento de Bachet, también llamado procedimiento de las terrazas y el procedimiento hindú.

A continuación vamos a desarrollar el estudio del primero por ser el más apropiado para su aplicación en un programa de computación.

2.2. Procedimiento de Bachet

El siguiente es un viejo procedimiento para componer cuadrados mágicos de orden impar, es decir, cuadrados con cualquier número impar de casillas: 3×3 , 5×5 , 7×7 , etc.

Este procedimiento fue propuesto en el siglo XVII por el matemático francés Bachet.

Este procedimiento de Bachet sirve, entre otras cosas para el cuadrado de 9 casillas, resulta conveniente empezar su descripción por este ejemplo, por ser el más simple.

Después de dibujar un cuadrado cuadriculado en 9 casillas escribimos en orden creciente los números del 1 al 9, disponiéndolos en filas oblicuas, a tres en cada fila, como puede verse en la fig.1.

Los números que quedan fuera del cuadrado los escribimos dentro de él, de forma que pasen a los lados opuestos del cuadrado (pero permaneciendo en las mismas filas o columnas en que estaban).

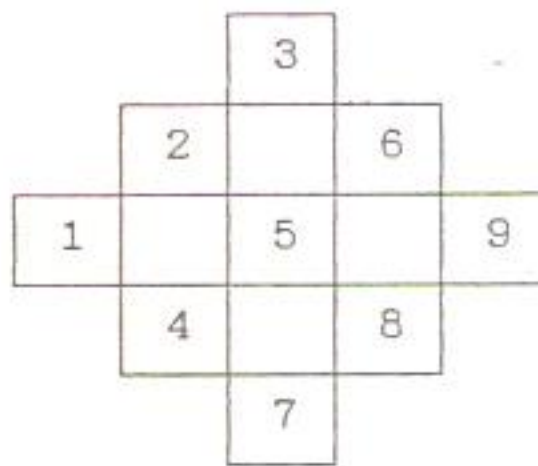


FIG. 1

Como resultado obtenemos el cuadrado:

2	7	6
9	5	1
4	3	8

FIG. 2

Apliquemos la regla de Bachet a la composición de un cuadrado de 5 x 5

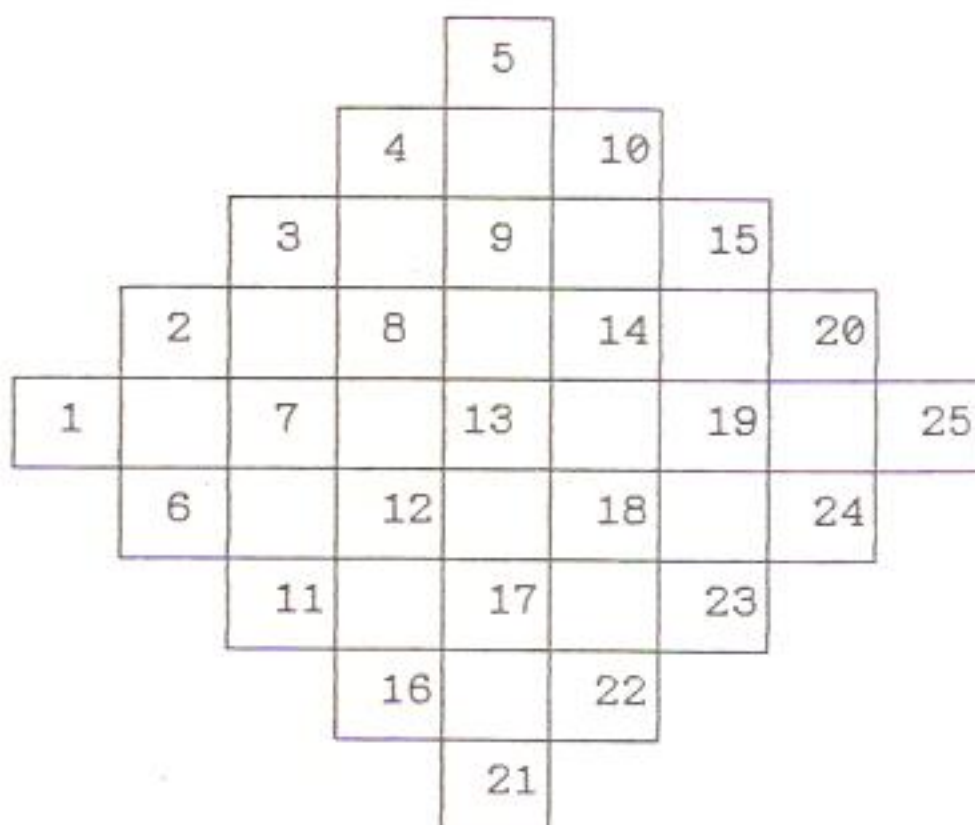


FIG. 3

Queda solamente poner dentro del cuadrado los números que han quedado fuera de su marco. Para eso hay que desplazar mentalmente las figuras formadas por los números que están fuera del cuadrado (terrazas), de modo que pasen a ocupar dentro de este los lados opuestos. De esta manera se obtiene un cuadrado mágico de 25 casillas. (FIG. 4)

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

FIG. 4

ACTIVIDADES

- 2.1. Realizar un programa que genere cuadrados mágicos de orden impar.
- 2.2. A partir de un cuadrado mágico de orden 3 generar un super-cuadrado mágico de orden 3 .

2.1.

>		
Running		
2	7	6
9	5	1
4	3	8

3 x 3

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

5 x 5

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

7 x 7

Running

5	46	15	56	25	66	35	76	45
54	14	55	24	65	34	75	44	4
13	63	23	64	33	74	43	3	53
62	22	72	32	73	42	2	52	12
21	71	31	81	41	1	51	11	61
70	30	80	40	9	50	10	60	20
29	79	39	8	49	18	59	19	69
78	38	7	48	17	58	27	68	28
37	6	47	16	57	26	67	36	77

9 x 9

Running

6	67	18	79	30	91	42	103	54	115	66
77	17	78	29	90	41	102	53	114	65	5
16	88	28	89	40	101	52	113	64	4	76
87	27	99	39	100	51	112	63	3	75	15
26	98	38	110	50	111	62	2	74	14	86
97	37	109	49	121	61	1	73	13	85	25
36	108	48	120	60	11	72	12	84	24	96
107	47	119	59	10	71	22	83	23	95	35
46	118	58	9	70	21	82	33	94	34	106
117	57	8	69	20	81	32	93	44	105	45
56	7	68	19	80	31	92	43	104	55	116

11 x 11

13 x 13

Running

7	92	21	106	35	120	49	134	63	148	77	162	91
104	20	105	34	119	48	133	62	147	76	161	90	6
19	117	33	118	47	132	61	146	75	160	89	5	103
116	32	130	46	131	60	145	74	159	88	4	102	18
31	129	45	143	59	144	73	158	87	3	101	17	115
128	44	142	58	156	72	157	86	2	100	16	114	30
43	141	57	155	71	169	85	1	99	15	113	29	127
140	56	154	70	168	84	13	98	14	112	28	126	42
55	153	69	167	83	12	97	26	111	27	125	41	139
152	68	166	82	11	96	25	110	39	124	40	138	54
67	165	81	10	95	24	109	38	123	52	137	53	151
164	80	9	94	23	108	37	122	51	136	65	150	66
79	8	93	22	107	36	121	50	135	64	149	78	163

15 x 15

Running

8	121	24	137	40	153	56	169	72	185	88	201	104	217	120
135	23	136	39	152	55	168	71	184	87	200	103	216	119	7
22	150	38	151	54	167	70	183	86	199	102	215	118	6	134
149	37	165	53	166	69	182	85	198	101	214	117	5	133	21
36	164	52	180	68	181	84	197	100	213	116	4	132	20	148
163	51	179	67	195	83	196	99	212	115	3	131	19	147	35
50	178	66	194	82	210	98	211	114	2	130	18	146	34	162
177	65	193	81	209	97	225	113	1	129	17	145	33	161	49
64	192	80	208	96	224	112	15	128	16	144	32	160	48	176
191	79	207	95	223	111	14	127	30	143	31	159	47	175	63
78	206	94	222	110	13	126	29	142	45	158	46	174	62	190
205	93	221	109	12	125	28	141	44	157	60	173	61	189	77
92	220	108	11	124	27	140	43	156	59	172	75	188	76	204
219	107	10	123	26	139	42	155	58	171	74	187	90	203	91
106	9	122	25	138	41	154	57	170	73	186	89	202	105	218

17 x 17

Running

9	154	27	172	45	190	63	208	81	226	99	244	117	262	135	280	153
170	26	171	44	189	62	207	80	225	98	243	116	261	134	279	152	8
25	187	43	188	61	206	79	224	97	242	115	260	133	278	151	7	169
186	42	204	60	205	78	223	96	241	114	259	132	277	150	6	168	24
41	203	59	221	77	222	95	240	113	258	131	276	149	5	167	23	185
202	58	220	76	238	94	239	112	257	130	275	148	4	166	22	184	40
57	219	75	237	93	255	111	256	129	274	147	3	165	21	183	39	201
218	74	236	92	254	110	272	128	273	146	2	164	20	182	38	200	56
73	235	91	253	109	271	127	289	145	1	163	19	181	37	199	55	217
234	90	252	108	270	126	288	144	17	162	18	180	36	198	54	216	72
89	251	107	269	125	287	143	16	161	34	179	35	197	53	215	71	233
250	106	268	124	286	142	15	160	33	178	51	196	52	214	70	232	88
105	267	123	285	141	14	159	32	177	50	195	68	213	69	231	87	249
266	122	284	140	13	158	31	176	49	194	67	212	85	230	86	248	104
121	283	139	12	157	30	175	48	193	66	211	84	229	102	247	103	265
282	138	11	156	29	174	47	192	65	210	83	228	101	246	119	264	120
137	10	155	28	173	46	191	64	209	82	227	100	245	118	263	136	281

19 x 19

Running

10	191	30	211	50	231	70	251	90	271	110	291	130	311	150	331	170	351	190
209	29	210	49	230	69	250	89	270	109	290	129	310	149	330	169	350	189	9
28	228	48	229	68	249	88	269	108	289	128	309	148	329	168	349	188	8	208
227	47	247	67	248	87	268	107	288	127	308	147	328	167	348	187	7	207	27
47	246	66	266	86	267	106	287	126	307	146	327	166	347	186	6	206	26	226
245	65	265	85	285	105	286	125	306	145	326	165	346	185	5	205	25	225	45
64	264	84	284	104	304	124	305	144	325	164	345	184	4	204	24	224	44	244
263	83	283	103	303	123	323	143	324	163	344	183	3	203	23	223	43	243	63
82	282	102	302	122	322	142	342	162	343	182	2	202	22	222	42	242	62	262
281	101	301	121	321	141	341	161	361	181	1	201	21	221	41	241	61	261	81
180	300	120	320	140	340	160	360	180	19	200	20	220	40	240	60	260	80	280
299	119	319	139	339	159	359	179	18	199	38	219	39	239	59	259	79	279	99
118	318	138	338	158	358	178	17	198	37	218	57	238	58	258	78	278	98	298
217	137	337	157	357	177	16	197	36	217	56	237	76	257	77	277	97	297	117
136	336	156	356	176	15	196	35	216	55	236	75	256	95	276	96	296	116	316
235	155	355	175	14	195	34	215	54	235	74	255	94	275	114	295	115	315	135
154	354	174	13	194	33	214	53	234	73	254	93	274	113	294	133	314	134	334
253	173	12	193	32	213	52	233	72	253	92	273	112	293	132	313	152	333	153
172	11	192	31	212	51	232	71	252	91	272	111	292	131	312	151	332	171	352

2.2.

A partir del cuadrado mágico inicial obtenemos los nueve cuadrados mágicos que componen el super cuadrado.

2	7	6
9	5	1
4	3	8

11	16	15	56	61	60	47	52	51
18	14	10	63	59	55	54	50	46
13	12	17	58	57	62	49	48	53
74	79	78	38	43	42	2	7	6
81	77	73	45	41	37	9	5	1
76	75	80	40	39	44	4	3	8
29	34	33	20	25	24	65	70	69
36	32	28	27	23	19	72	68	64
31	30	35	22	21	26	67	66	71

```

program generacion;
const n=3;
type
  cua=array[1..n,1..n] of real;
var
  matriz:cua;
  k,a,d,x,y,f,c:integer;
  (*#####*)
procedure generar(var m:cua;f,c:integer);
var
  i,k,a,d,x,y,aux,aux1,aux2,z,z1,z2:integer;
begin
  for k:=1 to N*N do
    begin
      a:=1+(k-1) div N;
      d:=1+(k-1) mod N;
      x:=a-d+N;
      y:=a+d-1;
      if ((x-(n div 2 +1))<0) and ((y-(n div 2 + 1)) <0) then
        begin
          f:=1 + (n - i);
          c:=1 + (n - i);
        end
      else
        begin
          if (x-(n div 2 +1))<0 then
            begin
              i:=((n div 2 +1) -x);
              f:=1 + (n - i);
              c:=1+(y-(n div 2 +1)mod n);
            end
          else
            begin
              if (y-(n div 2 +1))<0 then
                begin
                  i:=((n div 2 +1) - y);
                  f:=1+(x- (n div 2 +1) mod n);
                  c:=1 + (n - i);
                end
              else
                begin
                  F:=1+(x-(n div 2 +1)) mod n;
                  C:=1+(y-(n div 2 +1)) mod n;
                end;
            end;
          end;
        end;
      m[f,c]:=k;
    end;
  end;
  (*#####*)
procedure salida(m:cua);
var
  f,c:integer;
begin
  for f:=1 to N do
    begin

```



```
for c:=1 to N do
  write(lst,m[f,c]:4:0);
  writeln(lst);
end
end;
(#####)
begin (*programa principal*);
generar(matriz,f,c);
salida(matriz)
end.
```

```

program supercua;
const n=3;
type cua=array[1..n,1..n] of real;
      sup=array[1..9,1..9] of real;
var
  matriz:cua;matsup:sup;
  k,a,d,x,y,f,c:integer;
  (******)
PROCEDURE COMPONER (var m:cua; f,c:integer);
var
  i,k,a,d,x,y:integer;
begin
  for k:=1 to n*n do
    begin
      a:=1 + (k-1) div n;
      d:=1 + (k-1) mod n;
      x:=a - d + n;
      y:=a + d - 1;
      if (( x - (n div 2 +1) ) < 0 ) and (( y - (n div 2 +1) ) < 0 ) then
        begin
          f:=1 + (n-1);
          c:=1 + (n-1);
        end
      else
        begin
          if ( x - ( n div 2 +1 ) ) < 0 then
            begin
              i:=( ( n div 2 + 1 ) - x );
              f:=1+(n-i);
              c:=1 + ( y - (n div 2 + 1) mod n);
            end
          else
            begin
              if ( y - ( n div 2 +1 ) ) < 0 then
                begin
                  i:=( (n div 2 +1) - y );
                  f:=1 +(x-(n div 2 +1) mod n);
                  c:=1+(n-1);
                end
              else
                begin
                  f:=1 +(x-(n div 2 +1)) mod n;
                  c:=1 +(y-(n div 2 +1)) mod n;
                end;
            end;
          end;
        end;
      m[f,c]:=k;
    end;
  end;
  (******)
PROCEDURE SALIDA(m:cua);
var
  f,c:integer;
begin
  for f:=1 to n do

```



```

    for c:=1 to n do
        write(lst,m[f,c]:4:0);
        writeln(lst); writeln(lst);

    end;
end;
(#####)
PROCEDURE NEW (var m:cu; var s:sup);
var
    f,c,x,i,j:integer; p:real;
begin
    for f:=1 to n do
        begin
            for c:=1 to n do
                begin
                    p:=m[f,c];
                    begin
                        for x:=1 to n do
                            begin
                                for y:=1 to n do
                                    begin
                                        s[x,y]:=m[x,y] + (p-1)*9;
                                        write(lst,s[x,y]:4:0);
                                    end;
                                writeln(lst);
                            end;
                        end;
                    end;
                end;
            end;
        end;
    end;
end;
(#####)
BEGIN (*PROGRAMA PRINCIPAL*)
componer(matriz,f,c);
salida(matriz);
new(matriz,matsup);
end.

```

CAPITULO III

ROTACIONES Y REFLEXIONES

3.1. Introducción

Una vez compuesto un cuadrado mágico, es fácil obtener sus variantes, es decir, hallar una serie de nuevos cuadrados mágicos. Estos se obtienen mediante rotaciones y reflexiones.

3.2. Rotaciones

3.2.1. Rotación de 90°

A partir de un cuadrado mágico inicial, haciéndolo girar mentalmente un cuarto de vuelta, es decir, 90° se obtiene otro cuadrado mágico.

6	1	8
7	5	3
2	9	4

8	3	4
1	5	9
6	7	2

3.2.2. Rotación de 180°

Haciendo girar el cuadrado inicial 180° (media vuelta completa) obtenemos otra variante.

6	1	8
7	5	3
2	9	4

4	9	2
3	5	7
8	1	6

3.2.3. Rotación de 270°

Girando 270° obtenemos otra variante del cuadrado original.

6	1	8
7	5	3
2	9	4

2	7	6
9	5	1
4	3	8

3.3. Reflexiones

El cuadrado inicial puede modificarse, si nos figuramos como si vieramos su imagen reflejada en un espejo. Según el eje de simetría que se tome, aparecen nuevas variantes.

3.3.1. Reflexión de eje horizontal

6	1	8
7	5	3
2	9	4

2	9	4
7	5	3
6	1	8

3.3.2. Reflexión de eje vertical

6	1	8
7	5	3
2	9	4

8	1	6
3	5	7
4	9	2

3.3.3. Reflexión de eje diagonal

6	1	8
7	5	3
2	9	4

6	7	2
1	5	9
8	3	4

3.3.4. Reflexión de eje contradiagonal

6	1	8
7	5	3
2	9	4

4	3	8
9	5	1
2	7	6

ACTIVIDADES

- 3.1. A partir de un cuadrado mágico, mediante rotaciones y reflexiones obtener otras variantes.
- 3.2. Dados dos cuadrados mágicos indicar:
- 3.2.1. Si son esencialmente distintos.
- 3.2.2. Si no son esencialmente distintos señalar que movimiento se aplicó al primero para obtener el segundo.

3.1.

6	1	8	
7	5	3	
2	9	4	
MEDIANTE UNA REFLEXION de eje horizontal OBTENEMOS:			
2	9	4	
7	5	3	
6	1	8	
MEDIANTE UNA REFLEXION de eje diag. OBTENEMOS:			
6	7	2	
1	5	9	
8	3	4	
MEDIANTE UNA REFLEXION de eje vertical OBTENEMOS:			
8	1	6	
3	5	7	
4	9	2	
MEDIANTE UNA REFLEXION de eje contradiag. OBTENEMOS:			
4	3	8	
9	5	1	
2	7	6	
MEDIANTE UNA ROTACION DE 180 GRADOS OBTENEMOS:			
4	9	2	
3	5	7	
8	1	6	
MEDIANTE UNA ROTACION DE 90 GRADOS OBTENEMOS:			
8	3	4	
1	5	9	
6	7	2	
MEDIANTE UNA ROTACION DE 270 GRADOS OBTENEMOS:			
2	7	6	
9	5	1	
4	3	8	

3.2.

6	1	8
7	5	3
2	9	4

2	9	4
6	1	8
7	5	3

SON ESENCIALMENTE DISTINTOS

6	1	8
7	5	3
2	9	4

4	9	2
3	5	7
8	1	6

El 2do. se obtuvo mediante una ROT. de 180 grados

6	1	8
7	5	3
2	9	4

2	9	4
7	5	3
6	1	8

El 2do. cuad. se obtuvo mediante una REFLEXION de eje HORIZ.


```

program transformacion;
const N=3;
type cua=array [1..N,1..N] of real;
var
matriz,auxiliar,matriz1,matriz2,matriz3,matriz4,matriz5,matriz6,matriz7:c
ua;
(#####)
procedure entradato(var a:cua);
var
  f,c:integer;
begin
  writeln(lst,'INGRESE LAS COMPONENTES DEL CUADRADO MAGICO');
  for f:=1 to n do
    for c:=1 to n do
      readln(a[f,c]);
  for f:=1 to n do
    begin
      for c:=1 to n do
        write(lst,a[f,c]:4:0);
        writeln(lst);
      end;
    end;
end;
(#####)
procedure reflexionh(var m,m1:cua);
var
  f,c:integer;
begin
  for f:=1 to n do
    for c:=1 to n do
      m1[f,c]:=m[n+1-f,c];
      writeln(lst,'MEDIANTE UNA REFLEXION de eje horizontal OBTENEMOS:');
      for f:=1 to n do
        begin
          for c:=1 to n do
            write(lst,m1[f,c]:4:0);writeln(lst);
          end;
        end;
      end;
end;
(#####)
procedure reflexiondiag(var m,m2:cua);
var
  f,c:integer;
begin
  for f:=1 to n do
    for c:=1 to n do
      m2[f,c]:=m[c,f];
      writeln(lst,'MEDIANTE UNA REFLEXION de eje diag. OBTENEMOS:');
      for f:=1 to n do
        begin
          for c:=1 to n do
            write(lst,m2[f,c]:4:0);
            writeln(lst);
          end;
        end;
      end;
end;
end;

```



```

(*#####*)
procedure reflexionv(var m,m3:cu);
var
  f,c:integer;
begin
  for f:=1 to n do
    for c:=1 to n do
      m3[f,c]:=m[F,N+1-C];
    writeln(lst,'MEDIANTE UNA REFLEXION de eje vertical OBTENEMOS:');
    for f:=1 to n do
      begin
        for c:=1 to n do
          write(lst,m3[f,c]:4:0);writeln(lst);
        end;
      end;
end;
(*#####*)
procedure reflexioncd(var m,m4:cu);
var
  f,c:integer;
begin
  for f:=1 to n do
    for c:=1 to n do
      m4[f,c]:=m[n+1-c,n+1-f];
    writeln(lst,'MEDIANTE UNA REFLEXION-de eje contradiag. OBTENEMOS:');
    for f:=1 to n do
      begin
        for c:=1 to n do
          write(lst,m4[f,c]:4:0);writeln(lst);
        end;
      end;
end;
(*#####*)
procedure rotacionA(var m,m5:cu);
var
  f,c:integer;
begin
  for f:=1 to n do
    for c:=1 to n do
      m5[f,c]:=m[n+1-f,n+1-c];
    writeln(lst,'MEDIANTE UNA ROTACION DE 180 GRADOS OBTENEMOS:');
    for f:=1 to n do
      begin
        for c:=1 to n do
          write(lst,m5[f,c]:4:0);writeln(lst);
        end;
      end;
end;
(*#####*)
procedure rotacionB(var m,m6:cu);
var
  f,c:integer;
begin
  FOR F:=1 to n do
    for c:=1 to n do
      m6[f,c]:=m[c,n+1-f];
    writeln(lst,'MEDIANTE UNA ROTACION DE 90 GRADOS OBTENEMOS:');
    for f:=1 to n do
      begin

```



```

        for c:=1 to n do
            write(lst,m6[f,c]:4:0);
            writeln(lst);
        end;
end;
(#####*)
procedure rotacionC(var m,m7:cu);
var
    f,c:integer;
begin
    for F:=1 to n do
        for c:=1 to n do
            m7[f,c]:=m[n+1-c,f];
        writeln(lst,'MEDIANTE UNA ROTACION DE 270 GRADOS OBTENEMOS:');
        FOR f:=1 to n do
            begin
                for c:=1 to n do
                    write(lst,m7[f,c]:4:0);
                    writeln(lst);
                end;
            end;
        end;
end;
(#####*)
begin (*programa principal*);
    entradato(matriz);
    reflexionh(matriz,matriz1);
    reflexiondiag(matriz,matriz2);
    reflexionv(matriz,matriz3);
    reflexioncd(matriz,matriz4);
    rotacionA(matriz,matriz5);
    rotacionB(matriz,matriz6);
    rotacionC(matriz,matriz7);
end.

```

```

program compara;
const n=3;
type cua= array[1..N,1..n] of real;
var
  matriz1,MATRIZ2:cua; M1,M2,M3,M4,M5,M6,M7: INTEGER;
(*#####*)
PROCEDURE ENTRADA (VAR a,b:cua);
var
  f,c:integer;
begin
  writeln('INGRESE LAS COMPONENTES DEL PRIMER CUADRADO');
  for f:=1 to n do
    for c:=1 to n do
      readln(a[f,c]);
  for f:=1 to n do
    begin
      for c:=1 to n do
        write (a[f,c]:4:0);
        writeln;
      end;
  writeln('INGRESE LAS COMPONENTES DEL SEGUNDO CUADRADO');
  for f:=1 to n do
    for c:=1 to n do
      readln(b[f,c]);
  for f:=1 to n do
    begin
      for c:=1 to n do
        write (b[f,c]:4:0);
        writeln;
      end;
  end;
end;
(*#####*)
PROCEDURE HORIZONTAL (var m1:integer; var a,b:cua);
var
  f,c:integer;
begin
  m1:=0;
  for f:=1 to n do
    for c:=1 to n do
      IF (a[f,c]=b[n+1-f,c]) then
        m1:=m1+1
      ELSE
        m1:=m1;
  readln(m1);
end;
(*#####*)
PROCEDURE DIAGONAL (VAR m2:integer; var a,b:cua);
var
  f,c:integer;
begin
  m2:=0;
  for f:=1 to n do
    for c:=1 to n do
      IF a[f,c]=b[c,f] then
        m2:=m2+1

```



```

        m2:=m2;
    readln(m2);
end;
(#####)
PROCEDURE VERTICAL (var m3:integer; var a,b:cua);
var
    f,c:integer;
begin
    m3:=0;
    for f:=1 to n do
        for c:=1 to n do
            IF a[f,c]=b[f,n+1-c] then
                m3:=m3+1
            ELSE
                m3:=m3;
        readln(m3);
    end;
(#####)
PROCEDURE CONTRADIA (var m4:integer; var a,b:cua);
var
    f,c:integer;
begin
    m4:=0;
    for f:=1 to n do
        for c:=1 to n do
            IF a[f,c]=b[n+1-c,n+1-f] then
                m4:=m4+1
            ELSE
                m4:=m4;
        readln(m4);
    end;
(#####)
PROCEDURE ROT180 (VAR m5:integer; var a,b:cua);
var
    f,c:integer;
begin
    m5:=0;
    for f:=1 to n do
        for c:=1 to n do
            IF a[f,c]=b[n+1-f,n+1-c] then
                m5:=m5+1
            ELSE
                m5:=m5;
        readln(m5);
    end;
(#####)
PROCEDURE ROT270 (var m6:integer; var a,b:cua);
var
    f,c:integer;
begin
    m6:=0;
    for f:=1 to n do
        for c:=1 to n do
            IF a[f,c]=b[n+1-c,f] then
                m6:=m6+1
            ELSE

```

```

        m6:=m6;
    readln(m6);
end;
(#####)
PROCEDURE ROT90 (var m7:integer; var a,b:cua);
var
    f,c:integer;
begin
    m7:=0;
    for f:=1 to n do
        for c:=1 to n do
            IF a[f,c]=b[c,n+1-f] then
                m7:=m7+1
            ELSE
                m7:=m7;
        readln(m7);
    end;
(#####)
BEGIN(*PROGRAMA PRINCIPAL*);
ENTRADA(MATRIZ1,MATRIZ2);
HORIZONTAL(M1,MATRIZ1,MATRIZ2);
DIAGONAL(M2,MATRIZ1,MATRIZ2);
VERTICAL(M3,MATRIZ1,MATRIZ2);
CONTRADIA(M4,MATRIZ1,MATRIZ2);
ROT180(M5,MATRIZ1,MATRIZ2);
ROT270(M6,MATRIZ1,MATRIZ2);
ROT90(M7,MATRIZ1,MATRIZ2);
IF M1=9 THEN
    write('ES REFLEXION HORIZONTAL')
ELSE
    BEGIN
        IF M2=9 THEN
            write('ES UNA REFLEXION DIAGONAL')
        ELSE
            BEGIN
                IF M3=9 THEN
                    write('ES UNA REFLEXION VERTICAL')
                ELSE
                    BEGIN
                        IF M4=9 THEN
                            write('ES UNA REF. CONTRADIAG.')
                        ELSE
                            BEGIN
                                IF M5=9 THEN
                                    write('ES ROTACION DE 180')
                                ELSE
                                    BEGIN
                                        IF M6=9 THEN
                                            write('ES ROTACION DE 270')
                                        ELSE
                                            BEGIN
                                                IF M7=9 THEN
                                                    write('ES ROTACION DE 90')
                                                ELSE
                                                    write('SON ESENCIALMENTE DISTINTOS');
                                            END;
                                        END;
                                    END;
                                END;
                            END;
                        END;
                    END;
                END;
            END;
        END;
    END;

```


END;
END;
END;
END;
END;
END;
END;

B I B L I O G R A F I A

* ENCICLOPEDIA SALVAT DEL ESTUDIANTE, TOMO X.

* PROBLEMAS Y EXPERIMENTOS RECREATIVOS, de Perelman, ED. MIR.