Universidad de Costa Rica
Facultad de Ciencias Exactas
Escuela de Matemáticas
MA-0320

# Tarea 4

II Ciclo 2020

## Fecha de entrega: 05 de noviembre del 2020

---

### Solución

---

1. **[15 Puntos]** El día martes 27 de octubre, se estará realizando una charla con la Dra. Gabriela Marín la cual hablará de las ciencias del diseño, la información de la charla se compartió previamente en los grupos en Telegram, aquel que no pueda conectarse a la charla, podrá verla en diferido por la página en Facebook de la carrera en donde quedará guardada. A partir de dicha charla se pedirá investigar sobre los siguientes puntos:

   *a*) Busque 3 artículos (en áreas diferentes de la ciencias de la computación) de la expositora y realice un resumen de cada uno de ellos. Ojo, no copiar el resumen que viene en el artículo.

   *b*) Mencione al menos a tres autores importantes de las ciencias del diseño (tema de la charla).

   *c*) Describa los principales elementos que se deben considerar al desarrollar investigación usando ciencias del diseño.

2. **[12 Puntos]** Del libro que se encuentra en mediación virtual titulado: Matemáticas discretas con aplicaciones de Susanna S. Epp,

   *a*) Realizar de la página 448 en adelante los ejercicios: 15,18,20

   Dibuje las gráficas dirigidas de las relaciones definidas en los ejercicios 13 al 18.

   15) Sea A $= \{2, 3, 4, 5, 6, 7, 8\}$ y se define una relación $R$ sobre A, como sigue: Para todas x, y $\in$ A, x $R \Leftrightarrow$ y x|y.

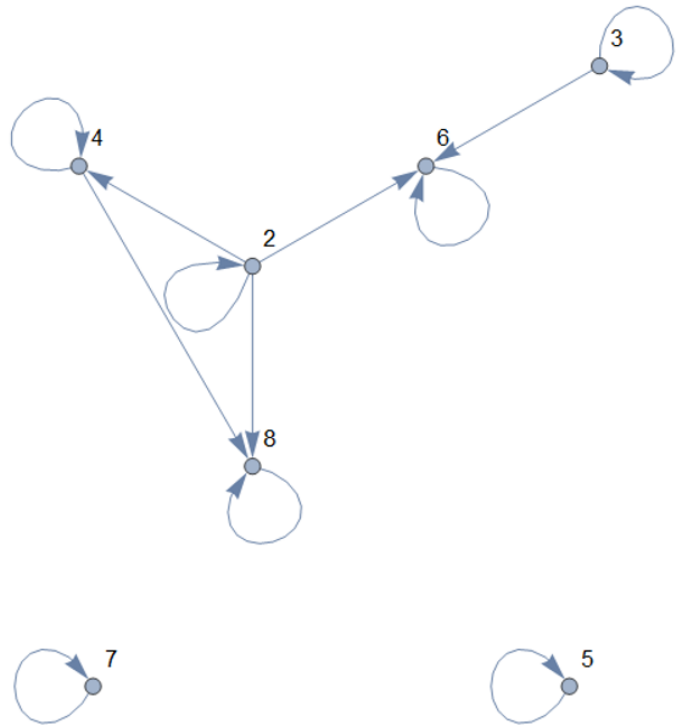



Figura 1: Hecho en mathematica

18) Sea A $= \{0, 1, 2, 3, 4, 5, 6, 7, 8\}$ y se define una relación V sobre A como sigue: Para todas x, y $\in$ A, x V y $\Leftrightarrow 5|(x^2 - y^2)$.

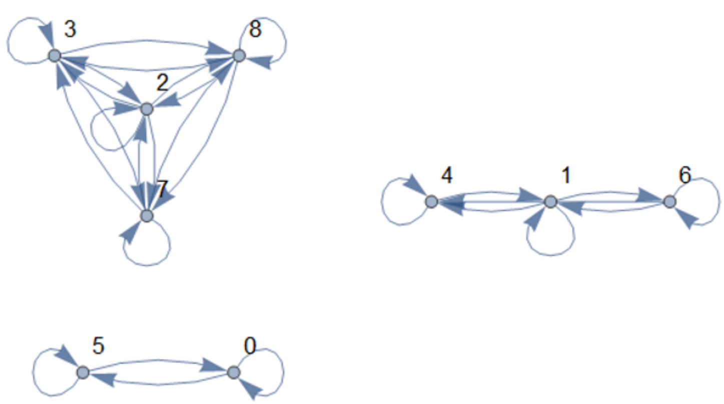



Figura 2: Hecho en mathematica

20) Sea A $= \{1, 1, 2, 4\}$ y B $= \{1, 2\}$ y se definen las relaciones R y S de A a B como sigue: Para toda (x, y) $\in$ A X B,
x $R$ y $\Leftrightarrow |x| = |y|$ y
x $S$ y $\Leftrightarrow x - y$ es par.

Establezca explícitamente cuáles pares ordenados están en A x B, R, S, $R \cup S$ y $R \cap S$

$G_{AxB}$= {(-1,1),(-1,2),(1,1),(1,2),(2,1),(2,2),(4,1),(4,2)}
$G_R$= {(-1,1),(1,1),(2,2)}
$G_S$= {(-1,1),(4,2)}
$G_{R \cup S}$= {(-1,1),(1,1),(2,2),(4,2)}
$G_{R \cap S}$= {(1,1),(2,2)}

*b*) Realizar de la página 458 en adelante los ejercicios: 1 al 6, 9, 11.

En los ejercicios del 1 al 8, se definen una serie de relaciones en el conjunto A $= \{0, 1, 2, 3\}$. Para cada relación:

a) Dibuje el grafo dirigido.
b) Determine si la relación es reflexiva.
c) Determine si la relación es simétrica.
d) Determine si la relación es transitiva.

Dé un contraejemplo en cada caso en el que la relación no satisface una de las propiedades.

1) $R_1 = \{(0,0), (0,1), (0,3), (1,1), (1,0), (2,3), (3,3)\}$

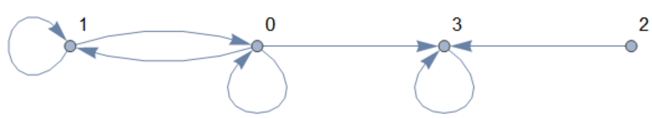



Figura 3: Hecho en mathematica

1) **Reflexiva**: No, por que no todos los elementos de $R_1$ se relacionan consigo mismo, donde 2 $\not R_1$ 2.
2) **Simétrica**: No, porque no tienen una relación recíproca entre los elementos, dado que 2 $R_1$ 3 pero 3 $\not R_1$ 2.
3) **Transitiva**: No, porque 1 $R_1$ 0 y 0 $R_1$ 3 pero 1 $\not R_1$ 3.

2) $R_2 = \{(0,0), (0,1), (1,1), (1,2), (2,2), (2,3)\}$

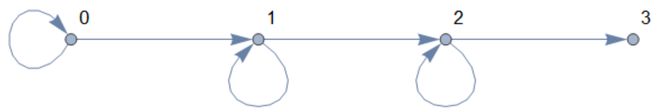



Figura 4: Hecho en mathematica

1) **Reflexiva**: No, porque $3 \not R_2 3$.
2) **Simétrica**: No, porque $1 R_2 2$ pero $2 \not R_2 1$.
3) **Transitiva**: No, porque $0 R_2 1$ y $1 R_2 2$ pero $0 \not R_2 2$.

3) $R_3 = \{(2,3),(3,2)\}$

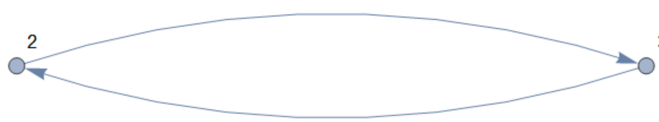



Figura 5: Hecho en mathematica

1) **Reflexiva**: No, porque ningún elemento se relaciona consigo mismo, $(0,0) \notin R_3$.
2) **Simétrica**: Sí, porque $2 R_3 3$ y $3 R_3 2$.
3) **Transitiva**: No, porque $2 R_3 3$ y $3 R_3 2$ pero $2 \not R_3 2$

4) $R_4 = \{(1,2),(2,1),(1,3),(3,1)\}$

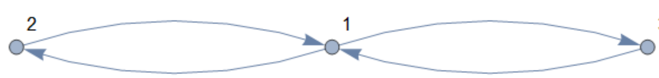



Figura 6: Hecho en mathematica

1) **Reflexiva**: No, por que los elementos de $R_4$ no se relacionan consigo mismo, asimismo $(0,0) \notin R_4$.
2) **Simétrica**: Sí, porque $1 R_4 3$ y $3 R_3 1$, asimismo $1 R_4 2$ y $2 R_4 1$.
3) **Transitiva**: No, porque $2 R_4 1$ y $1 R_4 3$ pero $2 \not R_4 3$.

5) $R_5 = \{(0,0),(0,1),(0,2),(1,2)\}$

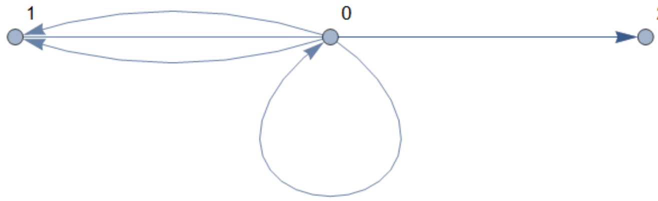



Figura 7: Hecho en mathematica

1) **Reflexiva**: No, porque $(3,3) \notin R_5$, 1 $\not R_5$ 1 y 2 $\not R_5$ 2.
2) **Simétrica**: No, por que 1 $R_5$ 2 pero 2 $\not R_5$ 1, asimismo $(3,3) \notin R_5$.
3) **Transitiva**: Sí, por que 0 $R_5$ 1, 1 $R_5$ 2 y 0 $R_5$ 2.

6) $R_6 = \{(0,1),(0,2)\}$

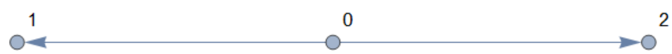



Figura 8: Hecho en mathematica

1) **Reflexiva**: No, porque ya que $(3,3) \notin R_6$, y ningún elemento de $R_6$ se relaciona consigo mismo.
2) **Simétrica**: No, por que 0 $R_6$ 2 pero 2 $\not R_6$ 0.
3) **Transitiva**: No, no existe un contra-ejemplo.

En los ejercicios del 9 al 33 determine si la relación dada es reflexiva, simétrica, transitiva o ninguna de las anteriores. Justifique sus respuestas.

9) R es la relación de "mayor o igual a" en el conjunto de los números reales: Para toda x, y $\in R$ y $\Leftrightarrow x \geq y$.

1) **Reflexiva**: Sí, por que se tiene que si sólo si para todos los números
1 = 1, 1 $R$ 1
2 = 2, 2 $R$ 2
-12 = -12, -12 $R$ -12
100 = 100, 100 $R$ 100

2) **Simétrica**: No, por que
$x = 1$ y $y = 0$
$13 \not> 153 = 13 \not R\ 153$

3) **Transitiva**: Sí por que porque para todos los números reales $x, y$y $z$, si $x$ es mayor o igual que $y$ $(x \geq y)$ y $y$ es mayor o igual que $z$ $(y \geq z)$.
$83 > 15$

11) C es la relación de la circunferencia sobre el conjunto de números reales: Para todos x, y $\in R$, x $D$ y $\Leftrightarrow xy \geq 0$.

1) **Reflexiva**: Sí, por que -100 $R$ -100.
2) **Simétrica**: Sí, por que 40 $R$ 34.
3) **Transitiva**: No, por que no se cumple que, para todos los números reales $x, y$ y $z$, si $xDy$ y $yDz$

3. **[20 Puntos]** Construya una rutina en Mathematica que reciba un conjunto $A$, su gráfico y determine

   a) La matriz asociada a la relación.
   b) El gráfo asociado a la relación.
   c) Haciendo uso del gráfico de la relación determine si la relación es: simétrica, reflexiva, antisimétrica, transitiva o total.
   d) Haciendo uso de la matriz de la relación determine si la relación es: simétrica, reflexiva, antisimétrica, transitiva o total.

```
Ejercicio3[A_, B_]:=Module[{i2, j2, bandera = 0},

EsReflexiva = True;

EsSimetrica = True;

EsAntisimetrica = True;

EsTransitiva = True;

EsTotal = True;

(*Primero construimos la matriz de la relación*)

T = ConstantArray[0, {Length[A], Length[A]}];

For[i2 = 1, i2 ≤ Length[A], i2++,
```

```
For[j2 = 1, j2 ≤ Length[A], j2++,
If[MemberQ[B, {A[[i2]], A[[j2]]}] == True,
T[[i2, j2]] = 1
];
];
];
Print [La matriz asociada a la relación corresponde a M_R = textMatrixForm[T]] ;
```

```
(*Vamos con las propiedades usando el gráfico*)
Print["La relación cumple las siguientes propiedades a partir de su gráfico"];
For[i = 1, i ≤ Length[A], i++,
If[MemberQ[B, {A[[i]], A[[i]]}] == False, bandera = 1;
Break[]
];
];
If[bandera==1, Print["La relación no es reflexiva"];
EsReflexiva = False,
Print["La relación es reflexiva"]];
bandera = 0;
```

```
For[i = 1, i ≤ Length[B], i++,
If[MemberQ[B, {B[[i, 2]], B[[i, 1]]}] == False,
bandera = 1;
Break[]
];
];
If[bandera==1, Print["La relación no es simétrica"];
EsSimetrica = False,
Print["La relación es simétrica"]];
bandera = 0;

For[i = 1, i ≤ Length[B], i++,
If[MemberQ[B, {B[[i, 2]], B[[i, 1]]}] == True&&B[[i, 1]]!=B[[i, 2]],
bandera = 1;
Break[]
];
];
If[bandera==1, Print["La relación no es antisimétrica"];
EsAntisimetrica = False,
Print["La relación es antisimétrica"]];
bandera = 0;

For[i = 1, i ≤ Length[B], i++,
For[j = 1, j ≤ Length[B], j++,
If[B[[i, 2]] == B[[j, 1]]&&MemberQ[B, {B[[i, 1]], B[[j, 2]]}] == False,
bandera = 1;
Break[];
```

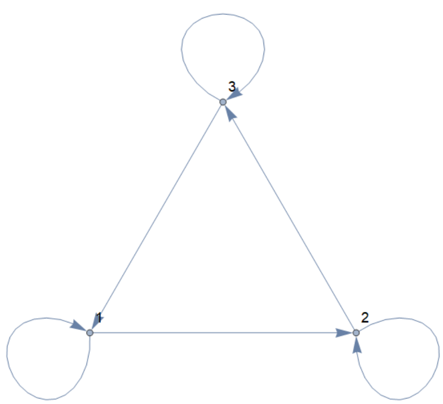
```
Break[];
];
];
];

If[bandera==1, Print["La relación no es transitiva"];
EsTransitiva = False,
Print["La relación es transitiva"]];
bandera = 0;

For[i = 1, i ≤ Length[A], i++,
For[j = 1, j ≤ Length[A], j++,
If[MemberQ[B, {A[[i]], A[[j]]}] == False&&MemberQ[B, {A[[j]], A[[i]]}] == False,
bandera = 1;
Break[];
Break[];
];
];
];

If[bandera==1, Print["La relación no es total"];
EsTotal = False,
Print["La relación es total"]];
bandera = 0;
(*Vamos con la demostración de las propiedades por medio de la matriz*)
Id = IdentityMatrix[Length[A]];
```

```
For[i = 1, i ≤ Length[A], i++,
For[j = 1, j ≤ Length[A], j++,
If[Id[[i, j]] > T[[i, j]], bandera = 1;
Break[];
];



];
];

If[bandera==1, Print["La matriz ", MatrixForm[Id], " no es menor o igual que la matriz ",
MatrixForm[T], "por ello la relación no es reflexiva."]; ,
Print["La matriz ", MatrixForm[Id], " es menor o igual que la matriz ", MatrixForm[T],
"por ello la relación es reflexiva."];
];
bandera = 0;



If[T ≠ Transpose[T],
Print["Como ", MatrixForm[T], "≠", MatrixForm[Transpose[T]], " la relación no es simétrica."]; ,
Print["Como ", MatrixForm[T], "=", MatrixForm[Transpose[T]], " la relación es simétrica."];
];

Print["El grafo de la relación corresponde a"];
AdjacencyGraph[T, VertexLabels → "Name"]
];

A = {1, 2, 3};
```

$\boldsymbol{B = \{\{1,1\},\{1,2\},\{2,2\},\{2,3\},\{3,3\},\{3,1\}\};}$

**Ejercicio3[$\boldsymbol{A, B}$]**

La matriz asociada a la relación corresponde a $M_R \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}$
La relación cumple las siguientes propiedades a partir de su gráfico
La relación es reflexiva
La relación no es simétrica
La relación es antisimétrica
La relación no es transitiva
La relación es total
La matriz $\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ es menor o igual que la matriz $\begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}$ por ello la relación es reflexiva.
Como $\begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix} \neq \begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}$ la relación no es simétrica.
El grafo de la relación corresponde a




4. **[13 Puntos]** Construya una rutina en Mathematica que reciba un conjunto $A$, su gráfico y determine

   *a*) Si la relación es de equivalencia.

   *b*) Determina las clases de equivalencia de la relación.

**Ejercicio3[A_, B_]:=Module[{i2, j2, bandera = 0},**

EsReflexiva = True;

EsSimetrica = True;

EsTransitiva = True;

(*Vamos con las propiedades usando el gráfico*)

Print[“La relación cumple las siguientes propiedades a partir de su gráfico”];

For[$i = 1, i \leq$ Length[$A$], $i$++,

If[MemberQ[$B$, {$A$[[$i$]], $A$[[$i$]]}] == False, bandera = 1;

Break[]

];

];

If[bandera==1, Print[“La relación no es reflexiva”];

EsReflexiva = False,

Print[“La relación es reflexiva”]];

bandera = 0;

For[$i = 1, i \leq$ Length[$B$], $i$++,

If[MemberQ[$B$, {$B$[[$i$, 2]], $B$[[$i$, 1]]}] == False,

bandera = 1;

Break[]

];

];

If[bandera==1, Print[“La relación no es simétrica”];

EsSimetrica = False,

Print[“La relación es simétrica”]];

bandera = 0;

```
For[i = 1, i ≤ Length[B], i++,
For[j = 1, j ≤ Length[B], j++,
If[B[[i, 2]] == B[[j, 1]]&&MemberQ[B, {B[[i, 1]], B[[j, 2]]}] == False,
bandera = 1;
Break[];
Break[];
];
];
];

If[bandera==1, Print["La relación no es transitiva"];
EsTransitiva = False,
Print["La relación es transitiva"]];
bandera = 0;

If[EsReflexiva == True&&EsSimetrica == True&&EsTransitiva == True,

Print["La relación es de equivalencia y sus clases de equivalencia corresponde a"];
For[i = 1, i ≤ Length[A], i++,
Cl_i = {};
For[j = 1, j ≤ Length[A], j++,
If[MemberQ[B, {A[[i]], A[[j]]}] == True,
Cl_i = Append [Cl_i, A[[j]]]] ;
];
```

```
Print ["[", i, "]", "=", Cl_i] ;

];,


Print["La relación no es de equivalencia"];

];


];

A = {1,2,3};

B = {{1,1},{1,2},{2,2},{3,3},{2,1}};



Ejercicio3[A, B]
```

La relación cumple las siguientes propiedades a partir de su gráfico

La relación es reflexiva

La relación es simétrica

La relación es transitiva

La relación es de equivalencia y sus clases de equivalencia corresponde a

$[1] = \{1, 2\}$

$[2] = \{1, 2\}$

$[3] = \{3\}$

5. **[40 Puntos]** Del libro de Enrique Vílchez que se encuentra en mediación virtual desde iniciado el semestre, del capítulo 4, página 110 realizar los ejercicios: 4.4, 4.5, 4.6, 4.7, 4.8, 4.9, 4.10 y 4.11

   4.4) Sea R la relación: a$R$b $\Leftrightarrow$ a $\geq$ , siendo a, b $\in \{2, 4, 6, ..., 100\}$, construya un programa en Mathematica para determinar si: 84 $R$ 96 y 24 $R$ 2.

```
(*Ejercicio 4.4*)

Eje1[B_List]:=Module[{A = {}, R = {}},
```

```
For[i = 2, i ≤ 100, A = Append[A, i]; i+=2];

ProdCart = CartesianProduct[A, A];

For[i = 1, i ≤ Length[ProdCart], L = ProdCart[[i]];

If[(L[[2]])^2 ≥ L[[1]], R = Append[R, L]]; i++];


If[MemberQ[R, B] == True, Return[True],

Return[False]]]


Eje1[{84, 96}]

Eje1[{24, 2}]
```

True

False

4.5) Halle con Mathematica el dominio y rango de la relación $R$ dada por: a$R$b si y solo el mínimo común múltiplo entre a y b es igual a 300, con a $\in \{1, 3, , 99\}$ y b $\in \{2, 4, 100\}$ Cual es el máximo valor del mínimo común múltiplo para que relación $R$ sea distinta de vació? Sugerencia: utilice el comando LCM que calcula el mínimo común múltiplo.

```
(*Ejercicio 4.5*)

Eje2[m_, t_, h_, x_]:=

Module[{i  = m, n, j = h, A = {}, B = {}, Grafico = {},

Dominio = {}, Rango = {}, Maximo = {}},

While[i ≤ t,

A = Append[A, i];

i+=2;

];
```

```
While[j ≤ x,
B = Append[B, j];
j+=2;
];
For[i = 1, i ≤ Length[A], i++,
For[j = 1, j ≤ Length[B], j++,
If[LCM[A[[i]], B[[j]]] == 300,
Grafico = Append[Grafico, {A[[i]], B[[j]]}];
Dominio = Append[Dominio, A[[i]]];
Rango = Append[Rango, B[[j]]];
];
];
];
Maximo = Rango;
For[i = 1, i ≤ Length[Dominio], i++,
For[j = 1, j ≤ Length[Maximo], j++,
If[Dominio[[i]] > Maximo[[j]],
Maximo = Dominio[[j]];
];
];
];
Dominio = DeleteDuplicates[Dominio];
Rango = DeleteDuplicates[Rango];
Print[“Gráfico es: ”, Grafico, “, el dominio es: ”,
Dominio, “, el rango es: ”, Rango,
```

```
" y el máximo valor para que R sea distinto al vacío es: ",
Maximo];
];
```

4.6) Encuentre a través del software Mathematica, la matriz de la relación binaria $R$ de los ejercicios 4.4 y 4.5

```
(*Ejercicio 4.6*)
Eje3[x_, z_]:=
Module[{i = x, j, Grafico = {}, A = {}, Salida = 0, n, M},
While[i ≤ z,
A = Append[A, i];
i+=2;
];
For[i = 1, i ≤ Length[A], i++,
For[j = 1, j ≤ Length[A], j++,
If[A[[i]] ≤ ((A[[j]])^2),
Grafico = Append[Grafico, {A[[i]], A[[j]]}];
];
];
];
M = ConstantArray[0, {Length[A], Length[A]}];
For[i = 1, i ≤ Length[A], i++,
For[j = 1, j ≤ Length[A], j++,
If[MemberQ[Grafico, {A[[i]], A[[j]]}] == True,
```

$M[[i, j]] = 1;$

];

];

];

Print[“La matriz es: ”, MatrixForm[$M$]];

];

Eje3[2, 100]

$$
\begin{pmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1
\end{pmatrix}
$$

```
Eje33[m_, t_, h_, x_]:=
Module[{i  = m, n, j = h, A = {}, B = {}, Grafico = {},
Z =  {}, Maximo, ñ, M},
While[i ≤ t,
A = Append[A, i];
i+=2;
];
While[j ≤ x,
B = Append[B, j];
j+=2;
];
For[i = 1, i ≤ Length[A], i++,
For[j = 1, j ≤ Length[B], j++,
If[LCM[A[[i]], B[[j]]] == 300,
Grafico = Append[Grafico, {A[[i]], B[[j]]}];
];
];
];
M = ConstantArray[0, {Length[A], Length[B]}];
For[i = 1, i ≤ Length[A], i++,
For[j = 1, j ≤ Length[B], j++,
If[MemberQ[Grafico, {A[[i]], B[[j]]}] == True,
M[[i, j]] = 1;
];
];
```

```
];
Print["La matriz es: ", MatrixForm[M]];
];
Eje33[2, 100, 1, 99]
```

$$
\begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}
$$

4.7) Represente por medio de un dígrafo la relación: $R = \{(a,b) | a - b \leq 7\}$, definida sobre

$A = \{l, 2, ..., 10\}$

(*Ejercicio 4.7*)

Eje4[A_]:=Module[$\{i, j, M,$ DimA$\}$,

DimA = Dimensions[$A$];

$M$ = ConstantArray[0, {DimA[[1]], DimA[[1]]}];

For[$i = 1, i \leq$ DimA[[1]], $i$++,

For[$j = 1, j \leq$ DimA[[1]], $j$++,

If[($A$[[$i$]] $-$ $A$[[$j$]]) $\leq 7$,

$M$[[$i, j$]] = 1;

];

];

];

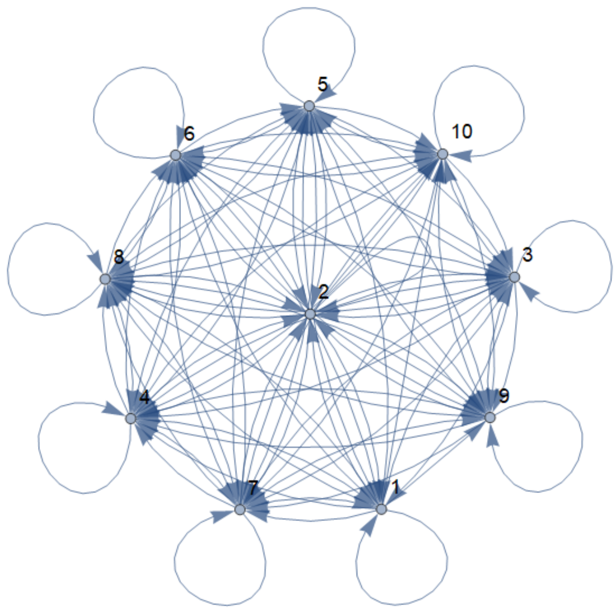
Print[" Su grafo es: "];

AdjacencyGraph[$M$,

VertexLabels $\rightarrow$ Table[$r \rightarrow 1r$, {$r$, Length[$A$]}],

ImagePadding $\rightarrow$ 10]

];




4.8) Determine con ayuda de Mathematica las relaciones unión e intersección entre $R_1$ e de $R_2$, si $R_1 = \{(1,1),(1,2),(2,1),(2,2),(3,1),(4,2)\}$y $R_2 = (a,b)|a + b \leq 20$, con a,$b \in \{1,2,3,...,100\}$. Halle además $\overline{R_1}$.

```
(*Ejercicio 4.8*)
Eje8[m_, k_, Grafico_]:=
Module[{i = m, n, A = {}, j, Grafico2 = {}, UnionRelaciones,
Interjeccion, Produc = {}, Complemento},
While[i ≤ k,
A = Append[A, i];
i+=1;
];
For[i = 1, i ≤ Length[A], i++,
For[j = 1, j ≤ Length[A], j++,
Produc = Append[Produc, {A[[i]], A[[j]]}];
If[(A[[i]] + A[[j]]) ≤ 20,
Grafico2 = Append[Grafico2, {A[[i]], A[[j]]}];
];
];
];
UnionRelaciones = Union[Grafico, Grafico2];
Interjeccion = Intersection[Grafico, Grafico2];
Complemento = Complement[Produc, Grafico];
Print ["R1 ∪ R2 = ", UnionRelaciones] ;
Print ["R1 ∩ R2 = ", Interjeccion] ;
Print ["R̄1 = ", Complemento] ;
];
G = {{1, 1}, {1, 2}, {2, 1}, {2, 2}, {3, 1}, {4, 2}};
Eje8[1, 100, G]
```

4.9) Con respecto al ejercicio anterior, encuentre por medio de software: $R_1^{-1}$ y $R_2^{-1}$. Utilizando operaciones con matrices boleanas, halle explícitamente:

$R_1^{-1} \cup R_2^{-1}$, $R_1^{-1} \cap R_2^{-1}$, $\overline{R_1^{-1} \cap R_2^{-1}}$ y $R_1^{-1} o R_2^{-1}$

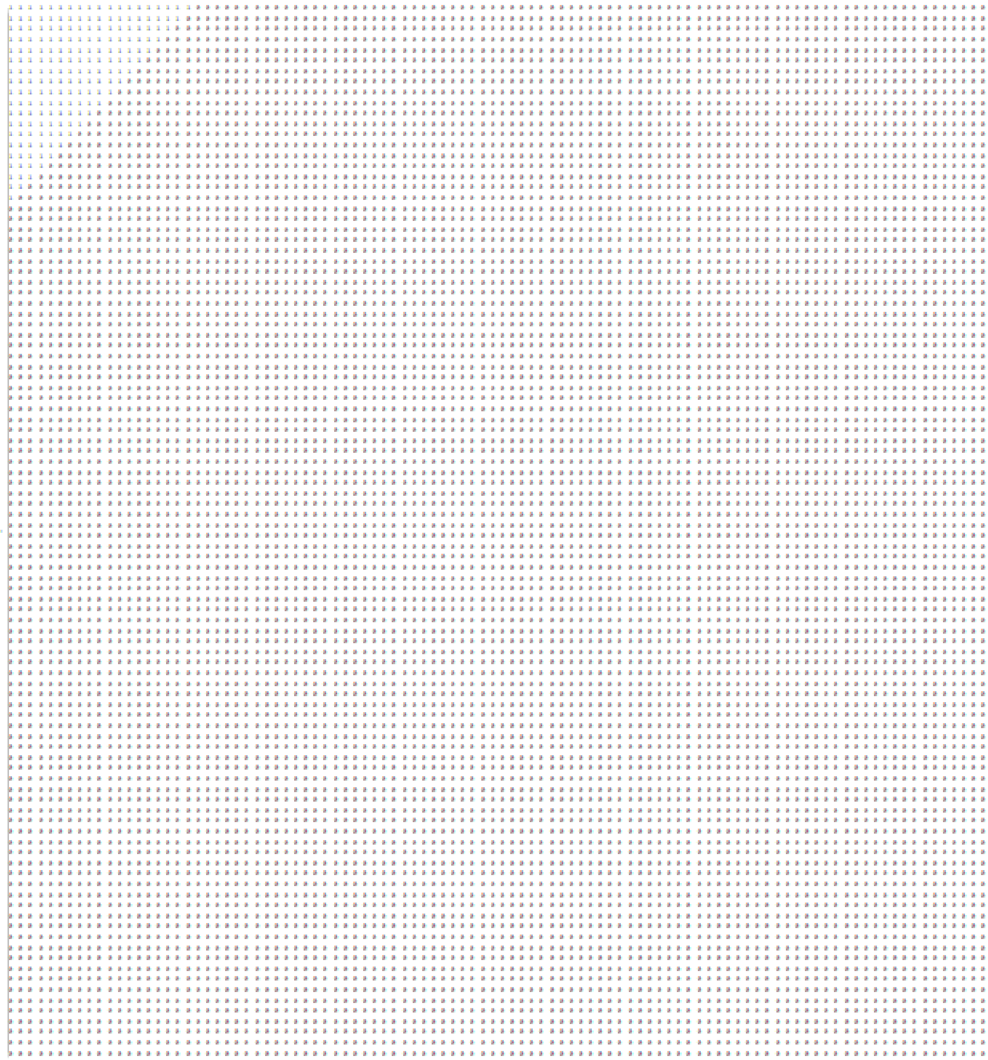
```
(*Ejercicio 4.9*)

Ejercicio9[]:=Module[{ProdCart, Universo = {}, RMUno = {}, RMdos = {}, R1 = {}, i, R2 = {}, L, P

UNImatrices, INTERmatrices, MatrizComplemento},

For[i = 1, i ≤ 100, Universo = Append[Universo, i]; i++];

ProdCart = CartesianProduct[Universo, Universo];

For[i = 1, i ≤ 20, i++,

For[j = 1, j ≤ 20, j++,

If[i + j ≤ 20, R2 = Append[R2, {i, j}]];

];

];

R1 = {{1, 1}, {1, 2}, {2, 1}, {2, 2}, {3, 1}, {4, 2}};


For[i = 1, i ≤ Length[R1], i++,

P = Append[P, R1[[i, 2]]];

P = Append[P, R1[[i, 1]]];

RMUno = Append[RMUno, P];

P = {};

];

For[i = 1, i ≤ Length[R2], i++,

P = Append[P, R2[[i, 2]]];

P = Append[P, R2[[i, 1]]];

RMdos = Append[RMdos, P];

P = {};

];
```

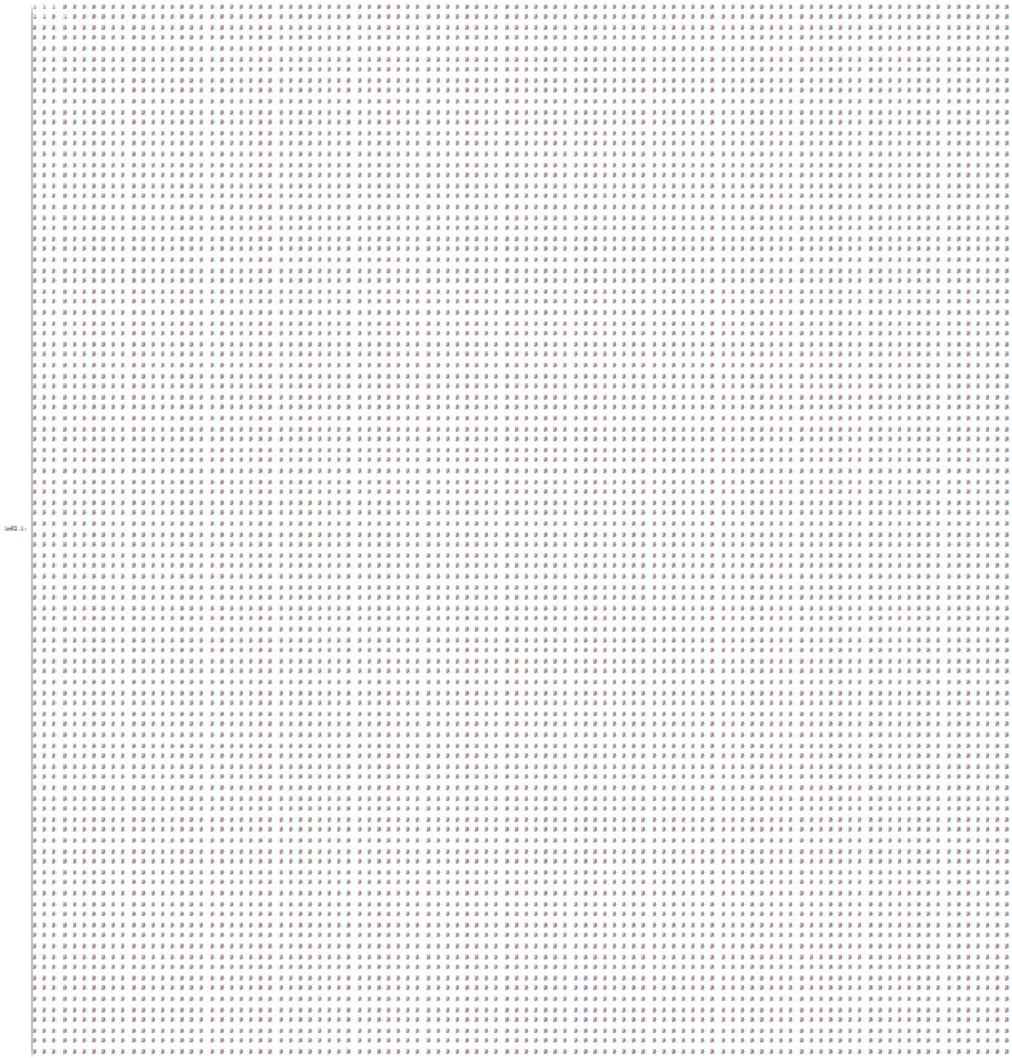
```
MatrizR1 = ConstantArray[0, {Length[Universo], Length[Universo]}];
For[i = 1, i ≤ Length[Universo], i++,
For[j = 1, j ≤ Length[Universo], j++,
If[MemberQ[RMUno, {Universo[[i]], Universo[[j]]}] == True,
MatrizR1[[i, j]] = 1;
];
];
];
MatrizR2 = ConstantArray[0, {Length[Universo], Length[Universo]}];
For[i = 1, i ≤ Length[Universo], i++,
For[j = 1, j ≤ Length[Universo], j++,
If[MemberQ[RMdos, {Universo[[i]], Universo[[j]]}] == True,
MatrizR2[[i, j]] = 1;
];
];
];

UNImatrices = ConstantArray[0, {Length[Universo], Length[Universo]}];
For[i = 1, i ≤ Length[Universo], i++,
For[j = 1, j ≤ Length[Universo], j++,
If[(MatrizR1[[i, j]] == 1)||(MatrizR2[[i, j]] == 1),
UNImatrices[[i, j]] = 1;
];
];
];
```

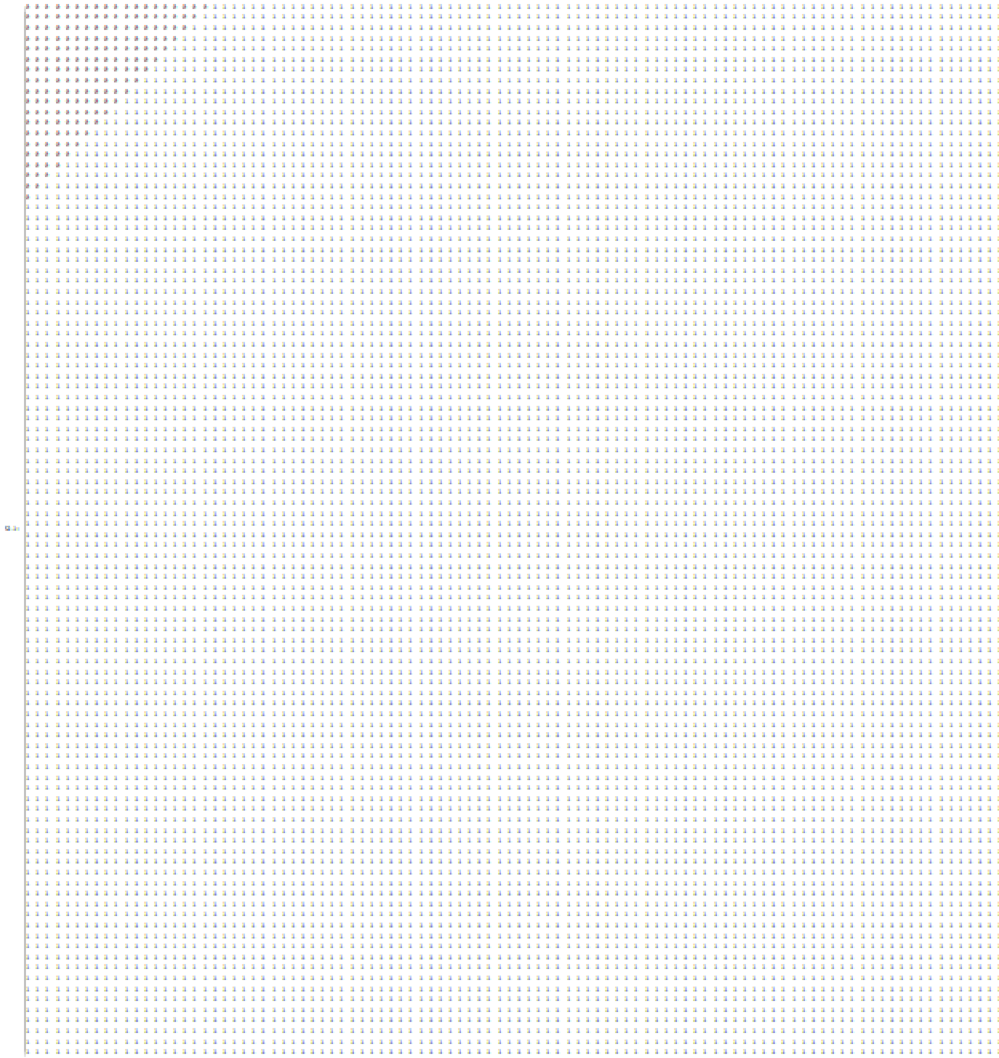
```
INTERmatrices = ConstantArray[0, {Length[Universo], Length[Universo]}];
For[i = 1, i ≤ Length[Universo], i++,
For[j = 1, j ≤ Length[Universo], j++,
If[(MatrizR1[[i, j]] == 1)&&(MatrizR2[[i, j]] == 1),
INTERmatrices[[i, j]] = 1;
];
];
];
MatrizComplemento = ConstantArray[0, {Length[Universo], Length[Universo]}];

For[i = 1, i ≤ Length[Universo], i++,
For[j = 1, j ≤ Length[Universo], j++,
If[UNImatrices[[i, j]] == 0,
MatrizComplemento[[i, j]] = 1;
];
];
];

Print["Inverso de R1: ", RMUno];
Print["Inverso de R2: ", RMdos];
Print["La matriz de la union de R1-1UR2-1: ", MatrixForm[UNImatrices]];
Print["La matriz de la interseccion de R1-1nR2-1: ", MatrixForm[INTERmatrices]];
Print["La matriz de la complemento de R1-1UR2-1: ", MatrixForm[MatrizComplemento]];
];
```

**Ejercicio9[]**

```
Inverso de R1: {{1, 1}, {2, 1}, {1, 2}, {2, 2}, {1, 3}, {2, 4}}
Inverso de R2: {{1, 1}, {2, 1}, {3, 1}, {4, 1}, {5, 1}, {6, 1}, {7, 1}, {8, 1}, {9, 1}, {10, 1}, {11, 1}, {12, 1}, {13, 1}, {14, 1
  {18, 1}, {19, 1}, {1, 2}, {2, 2}, {3, 2}, {4, 2}, {5, 2}, {6, 2}, {7, 2}, {8, 2}, {9, 2}, {10, 2}, {11, 2}, {12, 2}, {13, 2}, {14
  {17, 2}, {18, 2}, {1, 3}, {2, 3}, {3, 3}, {4, 3}, {5, 3}, {6, 3}, {7, 3}, {8, 3}, {9, 3}, {10, 3}, {11, 3}, {12, 3}, {13, 3}, {14
  {17, 3}, {1, 4}, {2, 4}, {3, 4}, {4, 4}, {5, 4}, {6, 4}, {7, 4}, {8, 4}, {9, 4}, {10, 4}, {11, 4}, {12, 4}, {13, 4}, {14, 4}, {15
  {2, 5}, {3, 5}, {4, 5}, {5, 5}, {6, 5}, {7, 5}, {8, 5}, {9, 5}, {10, 5}, {11, 5}, {12, 5}, {13, 5}, {14, 5}, {15, 5}, {1, 6}, {2,
  {5, 6}, {6, 6}, {7, 6}, {8, 6}, {9, 6}, {10, 6}, {11, 6}, {12, 6}, {13, 6}, {14, 6}, {1, 7}, {2, 7}, {3, 7}, {4, 7}, {5, 7}, {6,
  {9, 7}, {10, 7}, {11, 7}, {12, 7}, {13, 7}, {1, 8}, {2, 8}, {3, 8}, {4, 8}, {5, 8}, {6, 8}, {7, 8}, {8, 8}, {9, 8}, {10, 8}, {11,
  {2, 9}, {3, 9}, {4, 9}, {5, 9}, {6, 9}, {7, 9}, {8, 9}, {9, 9}, {10, 9}, {11, 9}, {1, 10}, {2, 10}, {3, 10}, {4, 10}, {5, 10}, {6
  {8, 10}, {9, 10}, {10, 10}, {1, 11}, {2, 11}, {3, 11}, {4, 11}, {5, 11}, {6, 11}, {7, 11}, {8, 11}, {9, 11}, {1, 12}, {2, 12}, {
  {5, 12}, {6, 12}, {7, 12}, {8, 12}, {1, 13}, {2, 13}, {3, 13}, {4, 13}, {5, 13}, {6, 13}, {7, 13}, {1, 14}, {2, 14}, {3, 14}, {4
  {6, 14}, {1, 15}, {2, 15}, {3, 15}, {4, 15}, {5, 15}, {1, 16}, {2, 16}, {3, 16}, {4, 16}, {1, 17}, {2, 17}, {3, 17}, {1, 18}, {2
```







4.10) Clasifique con apoyo de software, las relaciones de los ejercicios 4.4,4.7 y 4.8 en: reflexivas, simétricas, antisimétricas, transitivas, de equivalencia o de orden parcial.

```
(*Ejercicio 10*)

Eje10:=Module[{A, G, indicador = 0, i, j, EsReflexiva, EsSimetrica, EsAntisimetrica, EsTransitiva, Uni

For[i = 1, i ≤ 100, Universo = Append[Universo, i]; i++];

ProdCart = CartesianProduct[Universo, Universo];
```

```
For[i = 1, i ≤ 20, i++,
For[j = 1, j ≤ 20, j++,
If[i + j ≤ 20, R2 = Append[R2, {i, j}]];
];
];
A = Universo;
G = R2;
EsReflexiva = 1;
EsSimetrica = 1;
EsAntisimetrica = 1;
EsTransitiva = 1;
For[i = 1, i ≤ Length[G], i++,
If[MemberQ[G, {G[[i, 2]], G[[i, 1]]}] == False, indicador = 1;
Break[];
];
];
If[indicador==1, Print["La relación ", G, " no es simétrica"];
EsSimetrica = 0, Print["La relación ", G, " es simétrica"];
];
indicador = 0;
For[i = 1, i ≤ Length[A], i++,
If[MemberQ[G, {A[[i]], A[[i]]}] == False, indicador = 1;
Break[];
];
];
```

```
If[indicador==1, Print[“La relación ”, G, “ no es reflexiva”];
EsReflexiva = 0, Print[“La relación ”, G, “ es reflexiva”];
];
indicador = 0;
For[i = 1, i ≤ Length[G], i++,
If[(MemberQ[G, {G[[i, 2]], G[[i, 1]]}] == True)&&(G[[i, 2]] ≠ G[[i, 1]]), indicador = 1;
Break[];
];
];
If[indicador==1, Print[“La relación ”, G, “ no es antisimétrica”];
EsAntisimetrica = 0, Print[“La relación ”, G, “ es antisimétrica”];
];
indicador = 0;
For[i = 1, i ≤ Length[G], i++,
If[indicador==1, Break[]; ];
For[j = 1, j ≤ Length[G], j++,
If[G[[i, 2]] == G[[j, 1]]&&MemberQ[G, {G[[i, 1]], G[[j, 2]]}] == False,

indicador = 1; Break[]; ];
];


];
If[indicador==1, Print[“La relación ”, G, “ no es transitiva”];
EsTransitiva = 0, Print[“La relación ”, G, “ es transitiva”]; ];
If[EsReflexiva == 1&&EsSimetrica == 1&&EsTransitiva == 1,
```

```
Print["La relacion es de equivalencia"], Print["La relacion no es de equivalencia"]];
If[EsReflexiva == 1&&EsAntisimetrica == 1&&EsTransitiva == 1,
Print["La relacion es de orden parcial"], Print["La relacion no es de orden parcial"]];
];

Eje10[]
```

4.11) Construya con ayuda del comando **KSetPartitions**, las relaciones de equivalencia que se obtienen de las particiones de longitud tres sobre el conjunto $A = \{1, 2, 3, 4, 5\}$.

```
(* Ejercicio 4.11 *)
<< Combinatorica`
CP = KSetPartitions[4, 3];
For [i = 1, i ≤ Length[CP], C_i = CP[[i]]; i++]
(* Relaciones de equivalencia *)
For [i = 1, i ≤ Length[CP], R_i = {};
For [j = 1, j ≤ Length [C_i] , Pt_j = CartesianProduct [C_i[[j]], C_i[[j]]] ;
R_i = Union [R_i, Pt_j] ;
j++];
Print [R, i, "=", R_i] ;
i++]
```

*Tener esa capacidad de entender el comportamiento del mundo desde la óptica de lo que estudiamos, es una buena señal de nuestro proceso de aprendizaje...*
**MaLu**