

Ejercicios

Programación con restricciones – Fdl -UCM – RafaC

1) Datos

```
par int:M=100;  
par int:N=5;  
array [1..N,1..N] of var int: a;
```

establecer restricciones para que:

- a) La suma de valores en a sea m
- b) Todos los valores en a estén entre 1 y N
- c) Para todo valor de i de 1 a N , el valor i aparece al menos una vez en a .
- d) Los valores en a sean lo más pequeños posibles; si llamamos c al máximo de los valores en a , se trata de minimizar c .

Para mostrar un array de $N \times N$ utilizar:

```
output(  
  [show(a[i,j])++ " " ++  
    (if (j==N) then "\n" else " " endif) | i in 1..N, j in 1..N]);
```

- 2) Queremos encontrar un array t de tamaño $N \times N$ en los que cada casilla puede tomar los valores 0, 2, 3 o 5. En particular, la casilla $t[i,j]$ contendrá k con k en $\{2,3,5\}$ si $i+j$ es divisible entre k , o 0 en caso contrario.
- 3) Queremos encontrar un array de tamaño $N \times N$ con valores de $1..N$ en el que:
 - Si una casilla de una fila >1 contiene el valor N , la casilla de arriba (misma columna, fila una menos) tiene que ser un 1.
 - Si una casilla de una columna c , $1 < c < N$ contiene un valor par p , entonces las casillas de la izquierda y de la derecha contienen los valores $p-1$ y $p+1$, respectivamente
 - Todos los números $1..N$ aparecen al menos 2 veces en el array
- 4) Queremos encontrar un array de tamaño $N \times N$ con valores de $1..N$ en el que
 - a) Todas las filas son diferentes
 - b) Todas las columnas son diferentes
 - c) No hay ninguna fila ordenada de forma creciente
 - d) No hay ninguna columna ordenada de forma creciente.
- 5) Queremos encontrar vectores de tamaño N con valores entre $2..N$, tales que existe una posición i del array tal que la suma de los elementos en las restantes $N-1$ posiciones es $3*N-1$. Por ejemplo, para $N=4$, el array $[4, 4, 3, 2]$ verifica que, quitando el último elemento, $4+4+3 = 11 = 3*4-1$.