Exercici 35. Per a cadascun dels nombres enters $2 \leq N \leq 25$, calculeu una arrel primitiva mòdul N, o bé comproveu que no existeix.

Solució 35. Sabem que solament tenen arrels primitives les $N \in [2, 4, p^\alpha, 2p^\alpha]$, amb p primer i $\alpha \in \mathbb{N}, \alpha \geq 1$. Això vol dir que si $N \in [8, 12, 15, 16, 20, 21, 24]$, no hi hauran arrels primitives.
Anomenem ordre de g $(\#[g]), g \in (\frac{\mathbb{Z}}{N\mathbb{Z}})^*$, al primer nombre $d : g^d \equiv 1(modN)$.
Definim una arrel primitiva de $(\frac{\mathbb{Z}}{N\mathbb{Z}})^*$ com aquell element $g : \#[g] = \#(\frac{\mathbb{Z}}{N\mathbb{Z}})^*$.És a dir, g és arrel primitiva $(modN) \iff \#[g] = \varphi(N)$.

$\underline{N = 2 :}$
Anomenem al candidat per ser arrel primitiva: $a \in (\frac{\mathbb{Z}}{2\mathbb{Z}})^* \Rightarrow a \in [1]$. També sabem que $\varphi(2) = 1$.

$$1^1 \equiv 1^{\varphi(2)} \equiv 1(mod2) \Rightarrow 1 \text{ és arrel primitiva (mod 2).}$$

$\underline{N = 3 :}$
$a \in (\frac{\mathbb{Z}}{3\mathbb{Z}})^* \Rightarrow a \in [1, 2]$. També sabem que $\varphi(3) = 2$.

$$1^1 \equiv 1(mod3), \text{ però} \neq \varphi(3) \Rightarrow \text{No és arrel primivita (mod 3)}$$
$$2^1 \equiv 2(mod3)$$
$$2^2 \equiv 1(mod3) \Rightarrow 2 \text{ és arrel primitiva (mod 3)}$$

$\underline{N = 4 :}$
$a \in (\frac{\mathbb{Z}}{4\mathbb{Z}})^* \Rightarrow a \in [1, 3]$. També sabem que $\varphi(4) = 2$.
1 no és arrel primitiva, ja que $1 \neq \varphi(4)$

$$3^1 \equiv 3(mod4)$$
$$3^2 \equiv 1(mod4) \Rightarrow 3 \text{ és arrel primitiva (mod 4)}$$

$\underline{N = 5 :}$
$a \in (\frac{\mathbb{Z}}{4\mathbb{Z}})^* \Rightarrow a \in [1, 2, 3, 4, 5]$. També sabem que $\varphi(5) = 4$.
1 no és arrel primitiva, ja que $1 \neq \varphi(5)$

$$2^1 \equiv 2(mod5)$$
$$2^2 \equiv 4(mod5)$$
$$2^3 \equiv 3(mod5)$$
$$2^4 \equiv 1(mod5) \Rightarrow 2 \text{ és arrel primitiva (mod 5)}$$

I així seguim fins a $N = 25$. Podem resoldre-ho també amb el Mathematica. La instrucció *PrimitiveRoot[N]*, ens donarà una arrel primitiva (mod N), i la instrucció *PrimitiveRootList[N]*, ens donarà totes les arrels primitives (mod N).
Fem doncs una llista amb el que ens dona si busquem les arrels des de 2 fins a 25:

N	2	3	4	5	6	7	8	9	10	11	12	13
Arrel	1	2	3	2	5	3	{}	2	7	2	{}	2

N	14	15	16	17	18	19	20	21	22	23	24	25
Arrel	3	{}	{}	3	11	2	{}	{}	13	5	{}	2