

3. PRAKTIKA:

Multzoak, aplikazioak, zatigarritasuna eta konbinatoria

1. Multzoak

Union[zerrenda1, zerrenda2, ...] zerrenda1, zerrenda2, ... adierazpenetan agertzen diren elementu desberdin guztiak agertuko dira.

Intersection[zerrenda1, zerrenda2, ...] zerrenda1, zerrenda2, ... adierazpenetan agertzen diren elementu komunak agertuko dira.

Complement[zerrenda1, zerrenda2] zerrenda2-n ez dauden zerrenda1-eko elementuak agertuko dira.

Subsets[zerrenda] zerrenda-ko azpimultzo posibleak agertuko dira.

Length[zerrenda] zerrenda-ko elementuen kopurua agertuko da.

2. Funtzioak

- **Parametrorik ez badago**

Solve[sistema,{ezezagunak}] Sistemaren soluzioa lortuko du, existitzen bada.

NSolve[sistema,{ezezagunak}] Sistemaren soluzio hurbildua lortuko du, existitzen bada.

- **Parametroak badaude:**

Reduce[*sistema*, {*ezezagunak*}] Sistemaren ebazpenerako parametroen arabera aukera guztiak aztertuko ditu.

If[*baldintza*, *t*, *f*] *t* erantzuna emango du baldintza egiaztatzen bada eta *f* emango du baldintza egiaztatzen ez bada.

3. Zatigarritasuna

GCD[*n*₁, *n*₂, ...] *n*₁, *n*₂, ... balioen z.k.h. kalkulatu du.

LCM[*n*₁, *n*₂, ...] *n*₁, *n*₂, ... balioen m.k.t. kalkulatu du

PrimeQ[*adierazpena*] True agertuko da adierazpena zenbaki lehena bada eta False konposatua bada.

CoprimeQ[*n*₁, *n*₂] True agertuko da *n*₁ eta *n*₂ elkar lehenak badira eta False aurkako kasuan.

NextPrime[*n*] *n* zenbakiaren hurbilen dagoen zenbaki lehena adieraziko du (*n* baino handiagoa).

FactorInteger[*n*] *n* zenbakiaren faktore lehenen zerrenda kalkulatu du, bakoitza bere berretzailearekin.

Divisors[*n*] *n* zenbakiaren zatitzaileen zerrenda kalkulatu du.

4. Konbinatoria

Binomial[*n*, *m*] $\binom{n}{m}$ zenbaki binomikoa kalkulatu du.

