P3: Problema 2

Fiind dat un numar n pozitiv, se cere sa se determine toate descompunerile sale ca suma de numere prime distincte.

# MODEL RECURSIV

A white paper with black text

Description automatically generated

A white paper with black text

Description automatically generated

# COD

%nrDivizori(N:Integer, Div:Integer, Rez:Integer)

%Rez va contine numarul de divizori al numarului n

nrDivizori(N, Div, 0):-

Div > N,

!.

nrDivizori(N, Div, Rez):-

N mod Div =:= 0,

!,

Div2 is Div + 1,

nrDivizori(N, Div2, Rez2),

Rez is Rez2 + 1.

nrDivizori(N, Div, Rez):-

Div2 is Div + 1,

nrDivizori(N, Div2, Rez).

%prim(N:Integer)

%Verifica daca n este sau nu prim (se returneaza true/false)

prim(N):-

nrDivizori(N, 1, 2).

%candidat(N:Integer, Rez:Integer)

%Rez va lua valori din intervalul [1,N]

candidat(N, N):-

N >= 1.

candidat(N, Rez):-

N1 is N - 1,

N1 >= 1,

candidat(N1, Rez).

%sumPrimAux(N:Integer, Sum:Integer, C:list, Rez:list)

%in Rez se vor genera toate sumele de numere prime care dau N

sumPrimAux(N, N, C, C).

sumPrimAux(N, Sum, [C1|T], Rez):-

candidat(N,E),

prim(E),

Sum1 is Sum + E,

Sum1 =< N,

E < C1,

sumPrimAux(N, Sum1, [E|[C1|T]], Rez).

%sumPrim(N:Integer, Rez:list)

%in Rez se vor genera toate sumele de numere prime care dau N

%(i,i), (i,o)

sumPrim(N, Rez):-

candidat(N,E),

prim(E),

sumPrimAux(N,E,[E],Rez).