R.1

* Búsqueda exhaustiva con todas las posibilidades o alternativas, de soluciones parciales que conducen a la solución del problema.
* Vuelta atrás, si una solución parcial no conduce a la solución global del problema se vuelve atrás para probar con otra solución parcial.

R.2

R.3

* Única solución
* Mejor solución
* Todas las soluciones

EJERCICIO 6

Array aEntrada of char [1...n];

Array aSolu of char [1...n];

Array aMark of boolean [1...n];

void Permutar (char cP)

if ()

EJERCICIO 8

Array aA of integer [1...iN];

Array aSolu of integer [1...];

integer iN;

integer iTamSol;

void Backtra (integer iAcum, integer iTotal)

if (iAcum == iTotal) then

print (aSolu);

else

if (iAcum > iTotal) then

return;

else

for integer iInd = 1 to iN do

iAcum = iAcum + aA[iInd];

iTamSol ++;

aSolu [iTamSol] = aA[iInd];

Backtra (iAcum, iTotal);

iAcum == aA [iInd];

aSolu [iTamSol - -] = 0;

End

End

End

End

|  |  |  |  |
| --- | --- | --- | --- |
| + | - | \* | / |

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |

EJERCICIO 9

Array Simbol of char [1...4];

Array Sol of char [1...iM];

Function ExpArit (integer iN, integer iR)

if (iResul == iR) then

return iResul;

else

for integer iI = 0 to iN do

if (Simbol [i]