

1. Dar una definición de la función cambio utilizando la técnica de programación dinámica a partir de la siguiente definición recursiva (backtracking):

$$\text{cambio}(i, j) = \begin{cases} 0 & j = 0 \\ \infty & j > 0 \wedge i = 0 \\ \min_{q \in \{0, 1, \dots, j \div d_i\}} (q + \text{cambio}(i - 1, j - q * d_i)) & j > 0 \wedge i > 0 \end{cases}$$

```
fun cambio(d: array[1..n] of nat, monto: nat) ret cantidad: nat
  var tabla: array[0..n, 0..monto] of nat  {- tabla[i,j] = cambio(i,j) -}
  var minimo: nat

  {- Caso 1 -}
  for i := 0 to n do
    tabla[i, 0] := 0
  od

  {- Caso 2 -}
  for j := 1 to monto do
    tabla[0, j] := ∞
  od

  {- Caso 3 -}
  for i := 1 to n do
    for j := 1 to monto do
      minimo := ∞
      for q := 0 to (j ÷ d[i]) do
        minimo := minimo `min` q + tabla[i-1, j-q*d[i]]
      od
      tabla[i, j] := minimo
    od
  od

  cantidad := tabla[n, monto]
end fun
```