```
6
```

```
Pregunta 5
```

- a) i) orden de evaluación normal
- misteriosa "abc" (gen 1)
 - folder what const[] "abc" (gen 1)
- => what "a" \$ foldr what (const[]) "bc" (1:gen 1)
- = ("a",1) \$ +ddr what (wnst[]) "bc" (g on 2)
- =)
- ("a",1): what "b" \$ fddr what (const [])
 "c" (gen 2)
- => ("a",1): what "b" \$ folder what (const [])
 "("(2-gen3))
- => ("a",1):("b",2) \$ foldr what (const[]) "c" (gen 3)
- ('a",1): ("b",2) what "c" & folder what (const [])
- ("a",1): ("b",2) what "c" \$ fold what (const []) ")
 "" (3:9 an 4)

("a"; 1): ("b", 2): ("c",3) \$ folder what (const []) ""(gen4)

=> ("a",1): ("b", 1): ("c":3) : (const[]) ;" (gen 4)

= ("a",1): ("b",2): ("c",3): []

=> ("a",1); ("b",2) :[("c",5)]

=> ("a",1): [("b",2), ("c",3)]

=> [('a',1), ('b',2), ('e',3)]

ii) Orden aplication
misteriosa "abc" (g on 1)

misteriosa "abc" (1: g on 2)

misteriosa "abc" (1:2: gar 3)

Entra en evaluación infinta por que gen n no tiene limite superior.

b) foldA f b Hoga = b foldA f b Rama v Romal RamaR = Rama (+ value) (+old A + b RamaL) (sold + b Ramab) c) il Orden de evaluación normal sospectusa arbolito (gont 1) IddA what F (const Hoja) arbolito (gan A 1) IddA what F (const Hoga)
Rama 'a' (Rama b' Hogo (Rama 'c' Haja Hoga) Hoga) (gen A 1) what F 'a' (toldA what IF (const Hoga)

what F 'o' (told A what F (const Hoga)

(Rama' b'

(Rama' c' Hoga Hope)

) (told A what F (const Hoga) Hoga)

(gan A 1)

```
what IF 'a' ( fold A. what IF (const Hoga)
                          ( Rama 'b'
                             Hoga (c' Hoga Hoga)
                 ) (Jolah whatTF (const Hoga) Hoga)
                  (Rama
                      (gon (1+1))
   Rama ('a,1) ((fold A what TF (const Hopa)
                              (Rama 'b'
                                  (Rama 'c' Hoja Hoja)
               ) (gen A (141))
               1(fold A what IF (const Hoga) Hoga)
                  (gen A (1 x 2)))
Rama ('a', 1) ( what IF 'b' (fold A what IF (const Huga) Hoga)
            ( fold A what IF (const Hope)
                            (Roma 'c' Hoja Hoja ) |
              (gen (1+1)))
```

((fold A what IF (const Hopa) Hopa)

(gon (1+2)1)

=)

=)

(5)

```
Rama ('a', 1) ( whottf 'b'

(told A whattf (const Hogal Hoga)

1 told A whattf (const Hoga)

(Rama 'c' Hoga Hoga))

(Rama 2 (gon (2+1)) (gen (2*2)))

(I fold A whattf (const Hoga) Hoga)

(gen (1*2))).
```

Rama ('a', 1) (Rama ('b', 2)

(told A what IF (const Hoga) Hoga)

(fold A what IF (const Hoga)

(Rama 'c' Hoga Hoga)

(gen (2+2))

) (told A what IF (const Hoga) Hoga)

(gen A (1+2))

=> Rama ('a', 1) (
Rama ('b', 2)

((const Huga) (gen A (2+1))) (
(+old A what F (const Huga)

(Rama c) Huga

(gen (2+2)))

(gen (2+2)))

) ((toldA whatTF (const Hoga) Hoga)
(gen (1×2))

```
Rama ('a', 1)
          ( Rama (b',2)
                  Hoja
                  (solds what IF (const Hoga)
                                 (Rama 'c' Hoga Hoga)
                                  (gen (2*2)))
          ) (1 tolds what TF (const Hoga) Hoga)
(gen A (1+2))
   => Rama ('a', 1) (
               Rama (6,2)
                    Hoga
                     I what IF 'c' ( IddA what I Const Hogal
                                    Hogo)
                                   ( fold what TF ( const Hopa)
                                    Hoga) (gen (2x2)))
             ) ( (fold) what I (const Hopa) Hopa)
Rama ('d',1) (
             Rama (b, 2)
                   Hoga
           (what TF 'c' (fild A what TF (const Hoga)
Hoga)
                                  I toldA what TF (const Hopa)
                                   (Hoga) Rama 4,
                                           (gen (4+1)) (gen (4+2)))
```

) ((toldA whatTF const Hoga) (yen (1+2)))

```
Rama ('a', 1) (
       Rama (b, 2)
            Hoga
             ( Rama ('c', 4)
                [ I told A what I (const Hoga) Hoga)
                        (gon (4+1))
                ((told A whattf (const Hoga) Hoga)
gen (4x2))
    (gen (1+2)) (const Hoja) Hoja)
Rama ('a', 1) (
           Rama (b', 2)
                HOJA
                ( Rama ('c', 4)
                 ((folds whattf (const Hoga) Hoga)
(gen (4+1)))
) 11 told A what TF (const Hoga) Hoga)
         (gen A (+2))
```

=)

```
Rama ('a', 1)(
               Rama(16 12)
                  Hona
                   ( Rama ( 'c', 4)
                        4000
                        (1 folds what TF (const Hoga ) Hoga)
(gen (4*2))
         ) 11 toldA whatTF (const Hopa) Hoga)
(gen (1+2)1)
    => Rama ( 'a', 1) (
                 Rama (b',2) (
                      Hopa
                      ( Rama ('c', 4)
                             Maga
                            (const Hoja )(gen (4x2))
           ) ((toldA whatTF (anst Hoga) Hoga)
 =)
     Rama ('a', 1) (
               Rama (b', 1) Hoga Hoga Hoga
        (1 toldA whattf (const floga) Hoga)
(gen (42))
=)
```

```
Rama (6,1)
              Rama ('b', 2)
                  Hoga
                  Rama ('c', 4)
                        Hoga
           ) (( const Hopa) (gen (1+2))
       Rama ('a',1)
                Rama ('b', 2)
                     Hoga
Rama ('c', 4)
                            Hoga
                             Hoga
                Hoja
 (i)
    s ospechosa arbolito (genA 1)
    sospechosa arbolito Rama 1
                                (gon A (2))
(gon A (2))
=)
   sospectosa arbolito Rama 1
                                  (Rama 1
                                        (gen (3))
                                       (cgen (4))
                                  (gon A 2)
```

6

Entra en recursión infinita ya que gen An no tiene condición de parada.