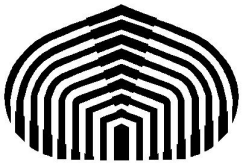

Universidad Simón Bolívar
CI3641: Lenguajes de Programación
Prof. Ricardo Monascal
Trimestre septiembre-diciembre 2023

Estudiante:
Jeamhowards Montiel
19-10234
Ing. Computación

Examen 1

1. Escoja algún lenguaje de programación de alto nivel y de propósito general cuyo nombre empiece con la misma letra que su nombre (por ejemplo, si su nombre es "Pedro", podría escoger "Perl", "PLI", "Python", etc.).

(a) De una breve descripción del lenguaje escogido.

i. Diga qué tipo de alcances y asociaciones posee, argumentando las ventajas y desventajas de la decisión tomada por los diseñadores del lenguaje, en el contexto de sus usuarios objetivos.

ii. Diga qué tipo de módulos ofrece (de tenerlos) y las diferentes formas de importar y exportar nombres.

iii. Diga si el lenguaje ofrece la posibilidad de crear aliases, sobrecarga y polimorfismo.

En caso afirmativo, dé algunos ejemplos.

iv. Diga qué herramientas ofrece a potenciales desarrolladores, como: compiladores, intérpretes, debuggers, profilers, frameworks, etc.

Respuesta:

a. - Lenguaje escogido: **Java**

Java es un lenguaje de programación de alto nivel, ampliamente utilizado en el desarrollo de aplicaciones de software, desde aplicaciones web y móviles hasta sistemas empresariales y embebidos. Fue creado por Sun Microsystems (ahora propiedad de Oracle) en la década de 1990 y se caracteriza por ser portátil y orientado a objetos, entre otras características.

i. Java es un lenguaje de programación que tiene alcance estático y asociación profuen.

- Alcance estático: Java utiliza la resolución de tipos en tiempo de compilación, lo que significa que las decisiones sobre qué método o campo acceder se toman en función de la declaración de tipos en tiempo de compilación. Esto proporciona ventajas en términos de detección temprana de errores y optimización del rendimiento, ya que el compilador puede realizar verificaciones de tipo más exhaustivas. Sin embargo, esto también significa que la resolución de tipos no puede cambiar en tiempo de ejecución, lo que puede limitar la flexibilidad en algunos casos.

- Asociación profunda: Java utiliza enlace profundo, lo que significa que el ambiente de referencia se calcula en el momento en que se construye la función. Esto permite la implementación de polimorfismo y flexibilidad en tiempo de ejecución, lo que es beneficioso para los usuarios que desean crear jerarquías de clases y aplicar herencia y polimorfismo.

El alcance estático con asociación profunda puede ser más difícil de usar para aplicaciones complejas y puede ser menos eficiente para aplicaciones que necesitan acceder a variables o métodos en diferentes ámbitos.

ii. . Java ofrece paquetes (packages) como la forma de modularizar. Los paquetes son una colección de clases y permiten controlar el acceso a las clases y miembros. Para importar y exportar nombres en Java, se utilizan declaraciones `import` y no se utiliza una declaración explícita para exportar nombres. Las clases dentro de un paquete pueden ser accedidas desde otros paquetes si se importan correctamente.

Ejemplo de importación en Java:

```
import java.util.ArrayList;
import paquete2.otropaquete.*;
```

iii. Java ofrece la posibilidad de crear **alias** referenciando a objetos a través de la declaración y uso de variables. Las referencias a objetos son variables que almacenan direcciones de memoria a objetos en lugar de los objetos en sí. Esto permite trabajar con objetos de manera flexible.

Ejemplo de alias:

```
public class ObjetoReferencia {
    public int valor;

    public ObjetoReferencia(int valor) {
        this.valor = valor;
    }

    public static void main(String[] args) {
        ObjetoReferencia obj1 = new ObjetoReferencia(10); // obj1 es alias al
            objeto
        ObjetoReferencia obj2 = obj1; // obj2 ahora es otro alias al mismo
            objeto que obj1

        System.out.println(obj1.valor); // Imprime 10
        System.out.println(obj2.valor); // Imprime 10

        obj2.valor = 20; // Cambiamos el valor a trav s de obj2

        System.out.println(obj1.valor); // Imprime 20, ya que ambos apuntan al
            mismo objeto
        System.out.println(obj2.valor); // Imprime 20
    }
}
```

Java admite la **sobrecarga**, lo que se traduce en java como que puede haber varios métodos con el mismo nombre en una clase, pero con diferentes parámetros.

Ejemplo de sobrecarga:

```
public class SobrecargaMetodos {
    public int suma(int a, int b) {
        return a + b;
    }

    public double suma(double a, double b) {
        return a + b;
    }

    public static void main(String[] args) {
        SobrecargaMetodos calculadora = new SobrecargaMetodos();

        int resultadoEntero = calculadora.suma(5, 7);
        double resultadoDouble = calculadora.suma(3.5, 2.7);

        System.out.println("Suma de enteros: " + resultadoEntero); // Imprime
            12
        System.out.println("Suma de doubles: " + resultadoDouble); // Imprime
            6.2
    }
}
```

El **polimorfismo** en java es común, es fácilmente visible a través de la herencia.

Ejemplo de polimorfismo:

```
class Figura {
    void dibujar() {
        System.out.println("Dibujando una figura");
    }
}

class Circulo extends Figura {
    void dibujar() {
        System.out.println("Dibujando un circulo");
    }
}

class Triangulo extends Figura {
    void dibujar() {
        System.out.println("Dibujando un triangulo");
    }
}

public class Main {
    public static void main(String[] args) {
        Figura[] figuras = new Figura[3];
```

```
        figuras[0] = new Circulo();
        figuras[1] = new Triangulo();
        figuras[2] = new Figura();

        for (Figura figura : figuras) {
            figura.dibujar(); // Polimorfismo
        }
    }
}
```

iv. Java ofrece una amplia gama de herramientas y recursos para desarrolladores, que incluyen:

- Compiladores: El compilador de Java (javac) se utiliza para compilar código fuente en archivos de clase ejecutables.
- Intérpretes: La máquina virtual de Java (JVM) actúa como un intérprete y ejecuta programas Java en tiempo de ejecución.
- Debuggers: Java proporciona herramientas de depuración como el Depurador de Java (jdb) y entornos de desarrollo integrados (IDEs) como Eclipse, NetBeans y IntelliJ IDEA que ofrecen capacidades de depuración.
- Profilers: Hay herramientas de perfiles disponibles como VisualVM y YourKit que permiten analizar el rendimiento de las aplicaciones Java.
- Frameworks: Java cuenta con una amplia variedad de marcos y bibliotecas para el desarrollo de aplicaciones, como Spring, Hibernate, JavaFX y muchos otros.

(b) Implemente los siguientes programas en el lenguaje escogido: i. Dada una cadena de caracteres w y un entero no–negativo k, calcular la rotación de k posiciones de la cadena w.

Por ejemplo:

- rotar(“hola”; 0) = “hola”
- rotar(“hola”; 1) = “olah”
- rotar(“hola”; 2) = “laho”
- rotar(“hola”; 3) = “ahol”
- rotar(“hola”; 4) = “hola”
- rotar(“hola”; 5) = “olah”

ii. Dada una matriz cuadrada A (cuya dimensión es $N \times N$), calcular el producto AA^T (donde A^T es la transpuesta de A).

Respuesta:

b.- El código fuente de la resolución del problema se encuentra en el siguiente repositorio de GitHub

Repositorio Examen 1

2. Considere el siguiente programa escrito en pseudo-código:

```
int a = Y + Z + 1, b = X + Y + 1, c = Z + Y + 1;

sub R (int b) {
    a := b + c - 1
}
sub Q (int a, sub r) {
    b := a + 1
    r(c)
}

sub P(int a, sub s, sub t) {
    sub R(int a) {
        b := c + a + 1
    }
    sub Q (int b, sub r) {
        c := a + b
        r(c + a)
        t(c + b)
    }
    int c := a + b
    if (a < 2 * (Y + Z + 1)) {
        P(a + 2 * (Y + Z + 1), s, R)
    } else {
        int a := c + 1
        s(c * a, R)
        Q(c * b, t)
    }
    print(a, b, c)
}

P(a, Q, R);
print(a, b, c)
```

Note que deberá reemplazar los valores para X, Y y Z como fue explicado en los párrafos de introducción del examen.

Diga qué imprime el programa en cuestión, si el lenguaje tiene:

(a) Alcance estático y asociación profunda

(b) Alcance dinámico y asociación profunda

(c) Alcance estático y asociación superficial

(d) Alcance dinámico y asociación superficial

Recuerde mostrar los pasos de su ejecución (por lo menos al nivel de cada nuevo marco de pila creado).

Respuesta:

(a) Alcance estático y asociación profunda - Tenemos $\mathbf{X = 2, Y = 3, Z = 4}$ por lo tanto, $\mathbf{a = Y + Z + 1 = 3 + 4 + 1 = 8}$, $\mathbf{b = X + Y + 1 = 2 + 3 + 1 = 6}$, $\mathbf{c = Z + Y + 1 = 4 + 3 + 1 = 8}$. A continuación se mostrará todo la pila de llamadas y los estados en cada momento.

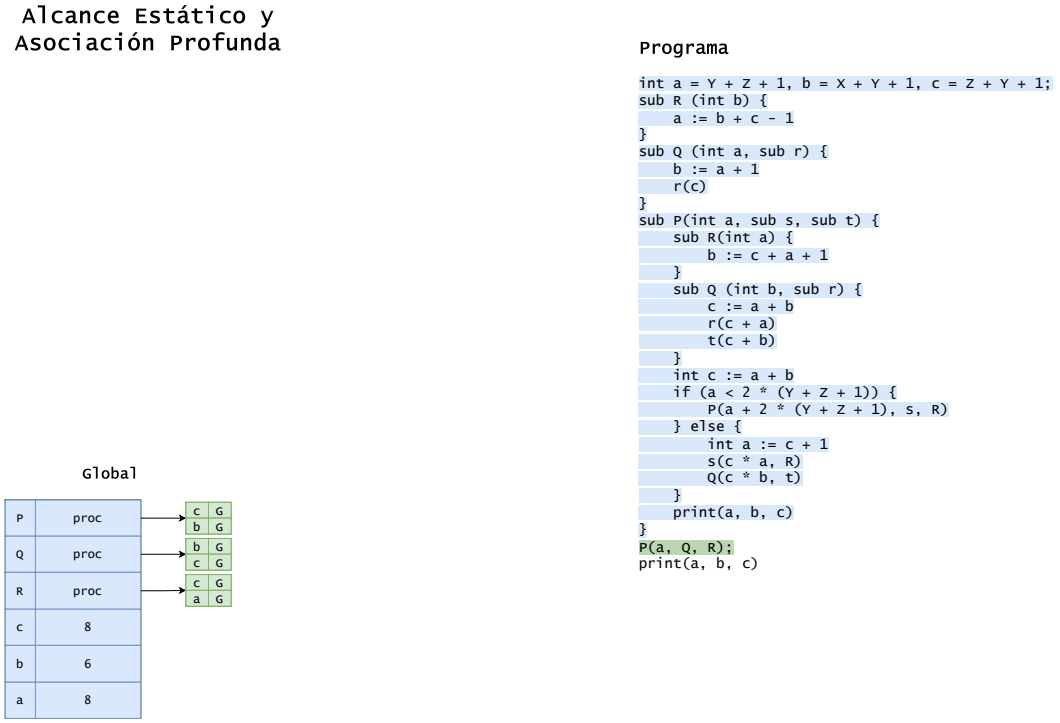

Figura 1: Alcance estático y asociación profunda - 1

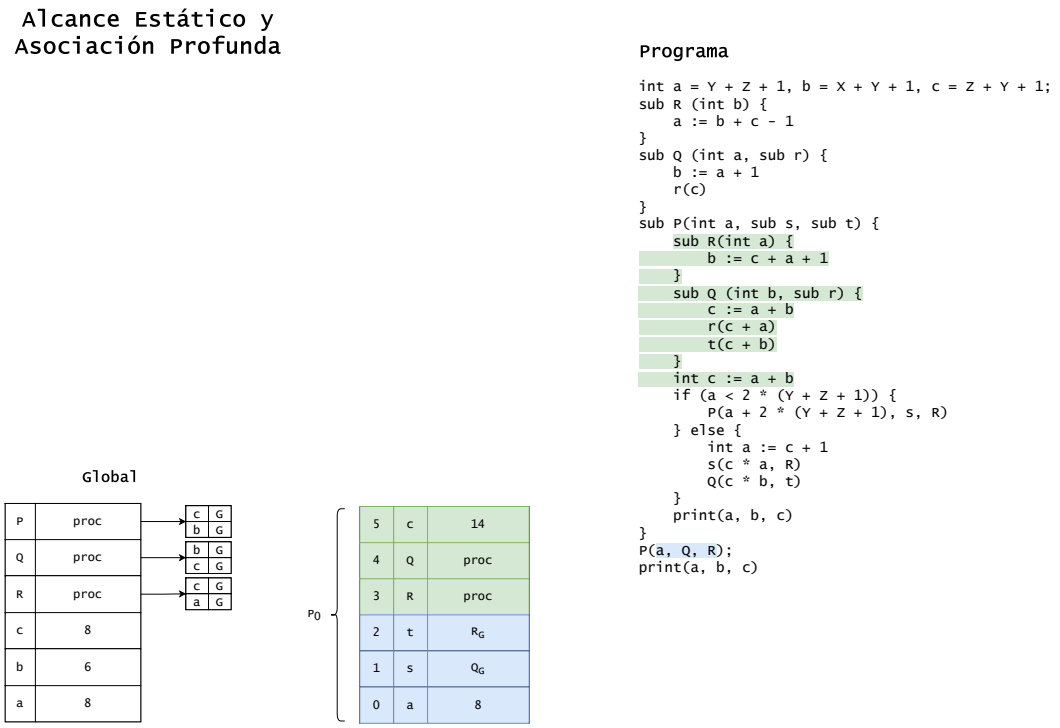

Figura 2: Alcance estático y asociación profunda - 2

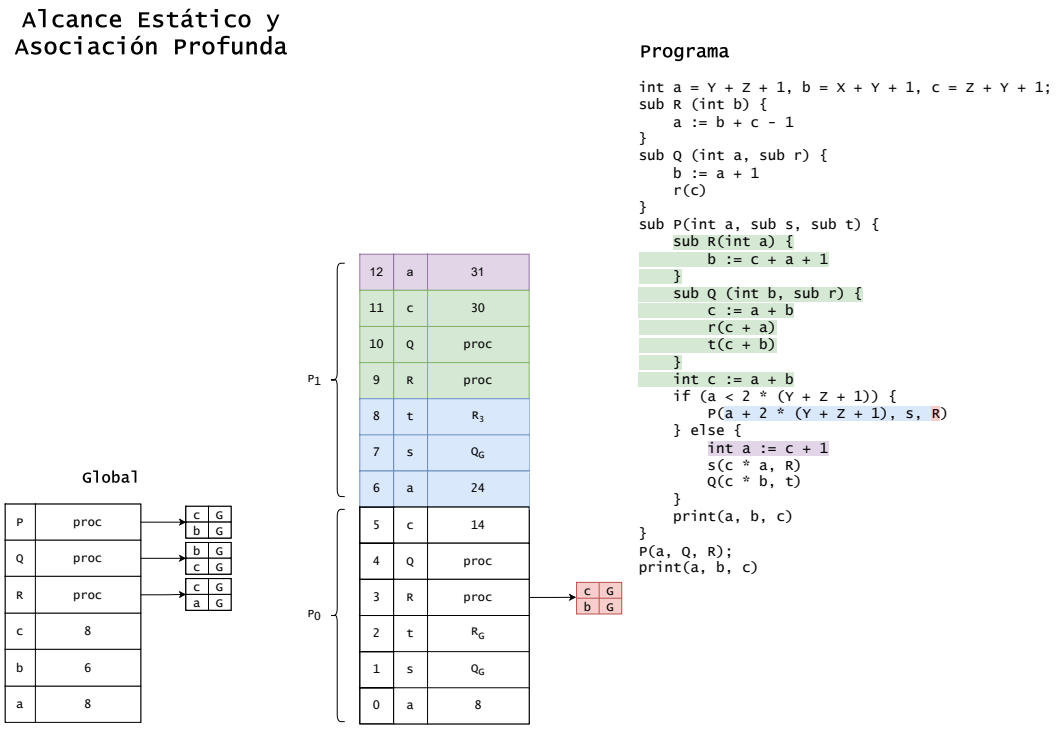

Figura 3: Alcance estático y asociación profunda - 3

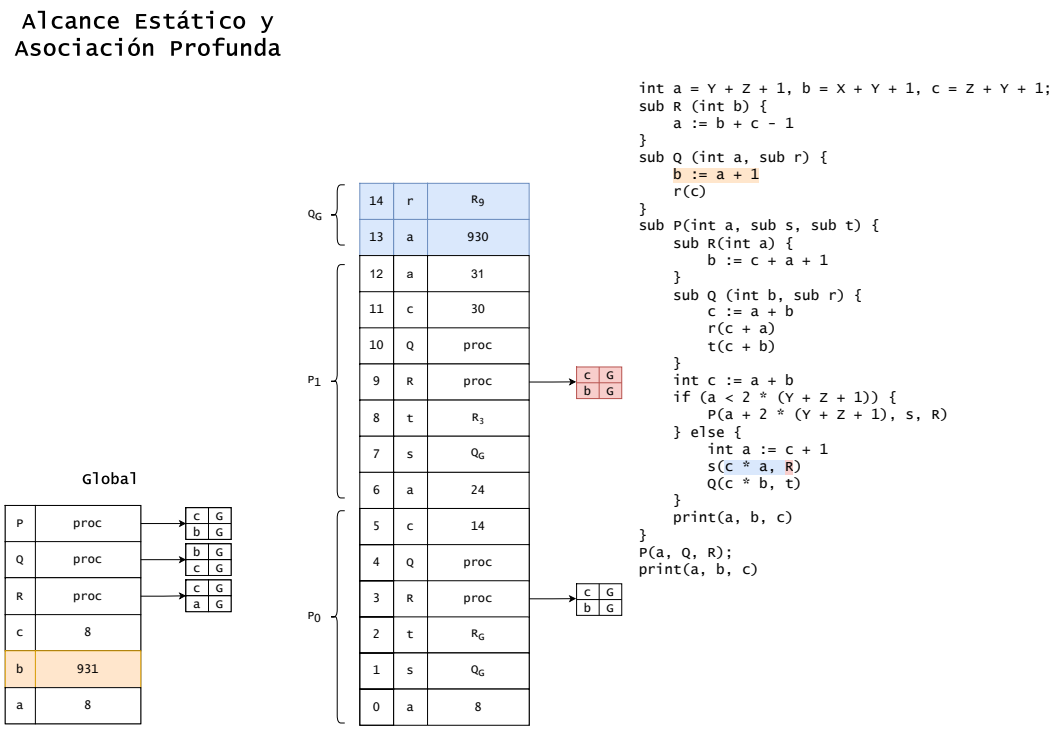

Figura 4: Alcance estático y asociación profunda - 4

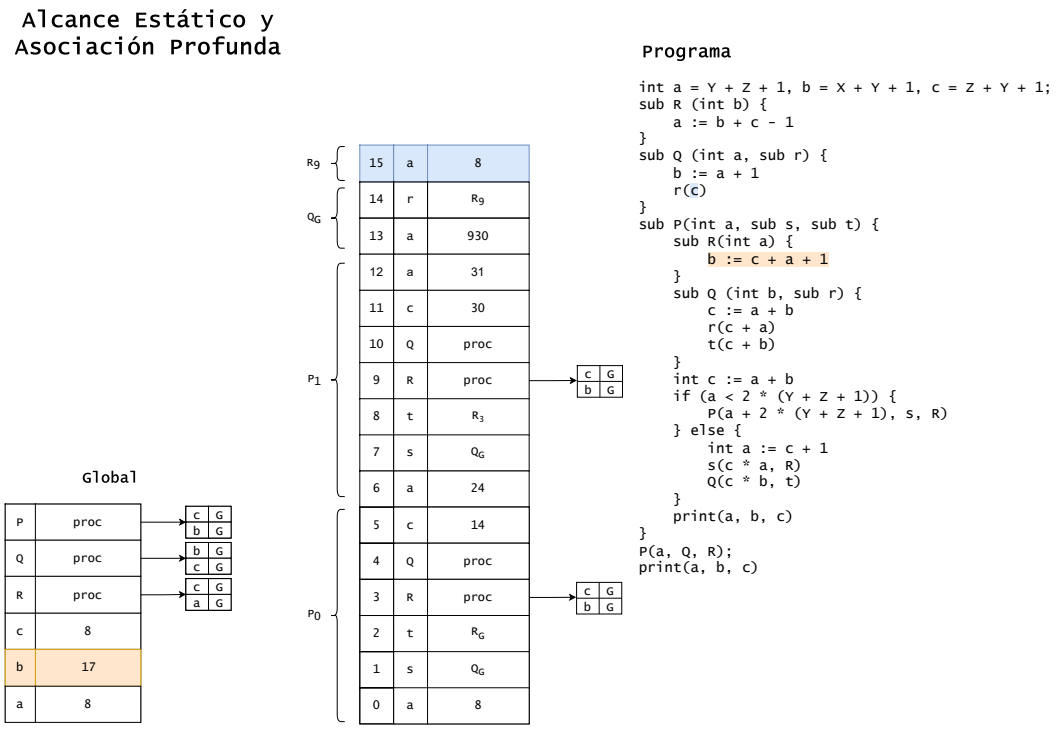

Figura 5: Alcance estático y asociación profunda - 5

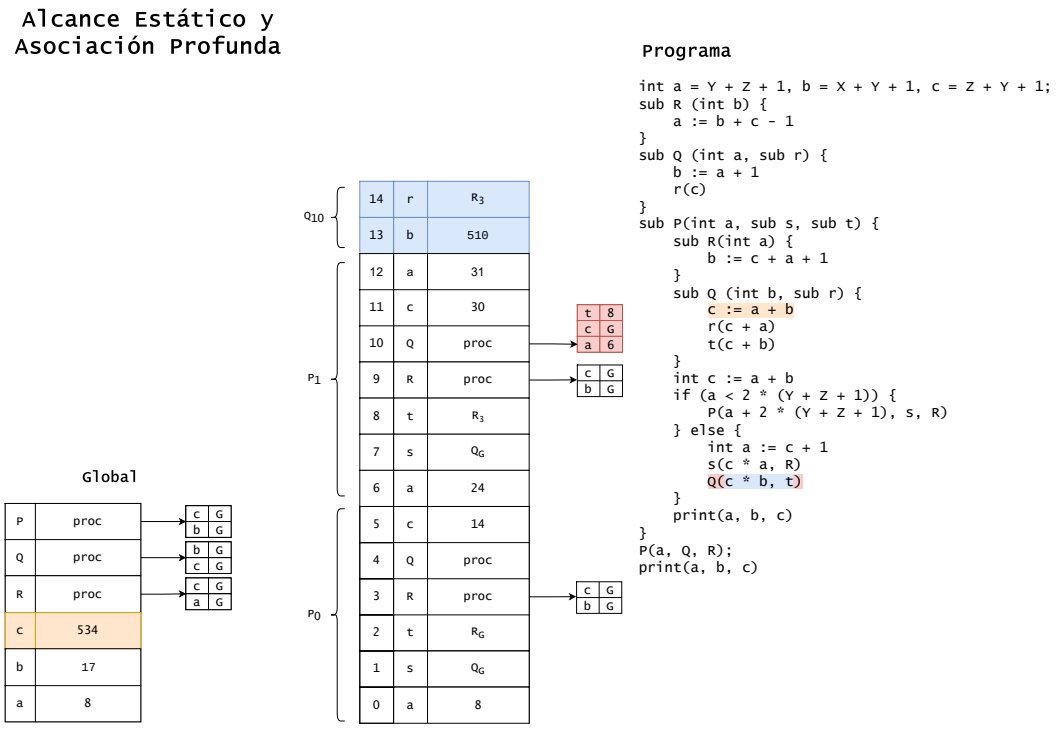

Figura 6: Alcance estático y asociación profunda - 6

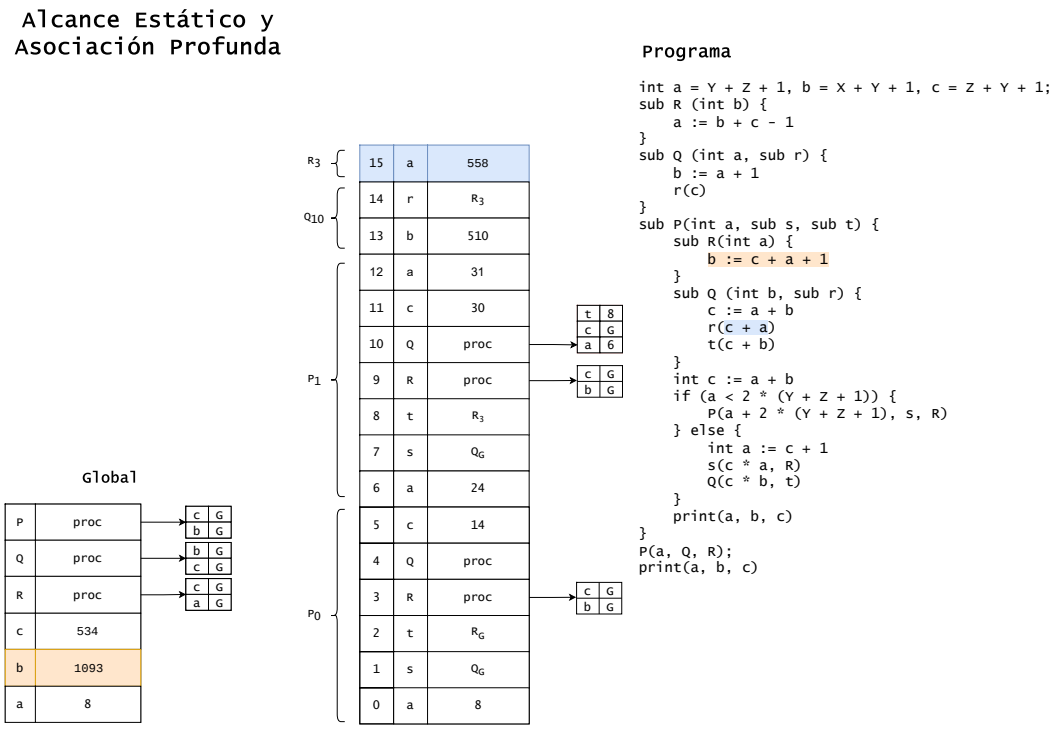

Figura 7: Alcance estático y asociación profunda - 7

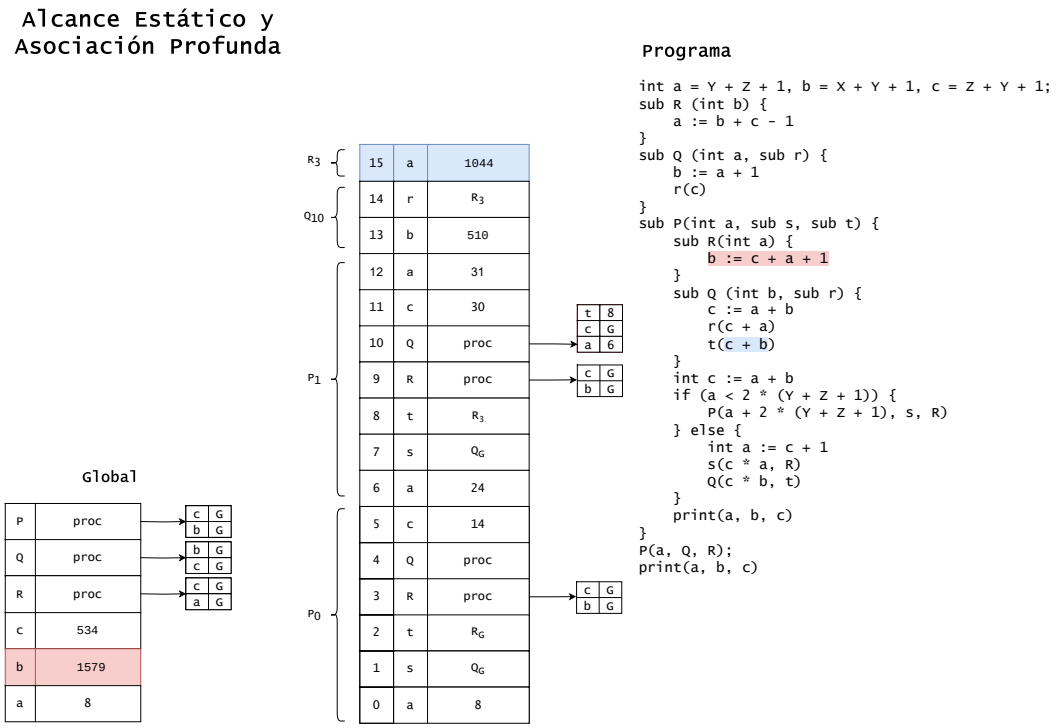

Figura 8: Alcance estático y asociación profunda - 8

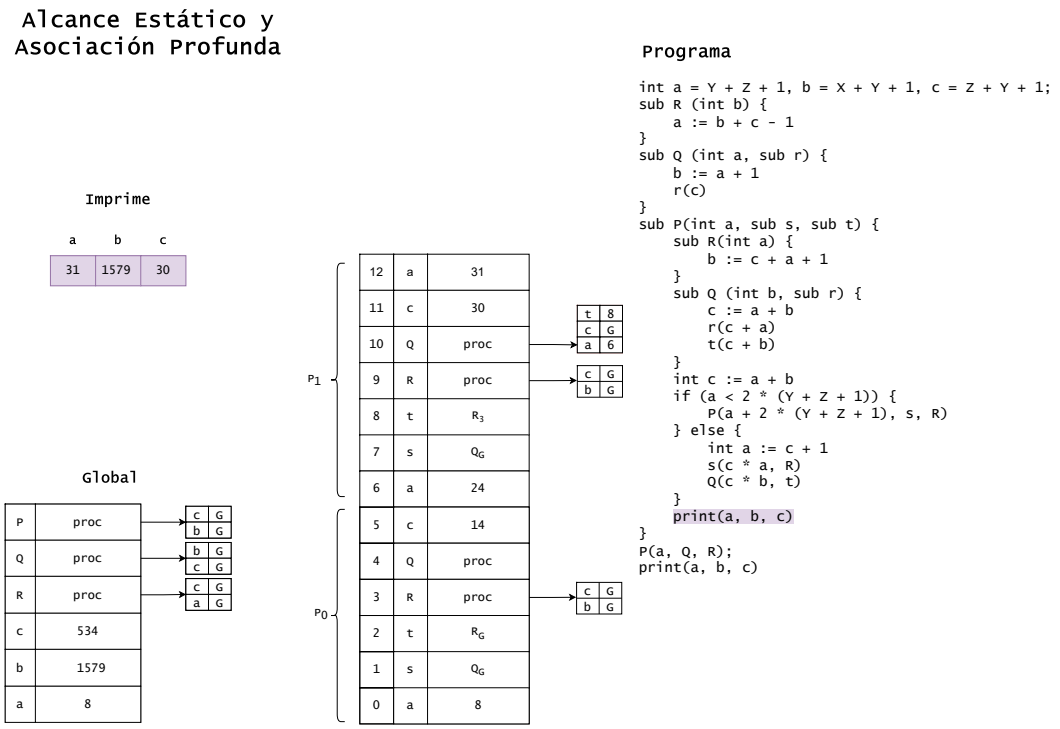

Figura 9: Alcance estático y asociación profunda - 9

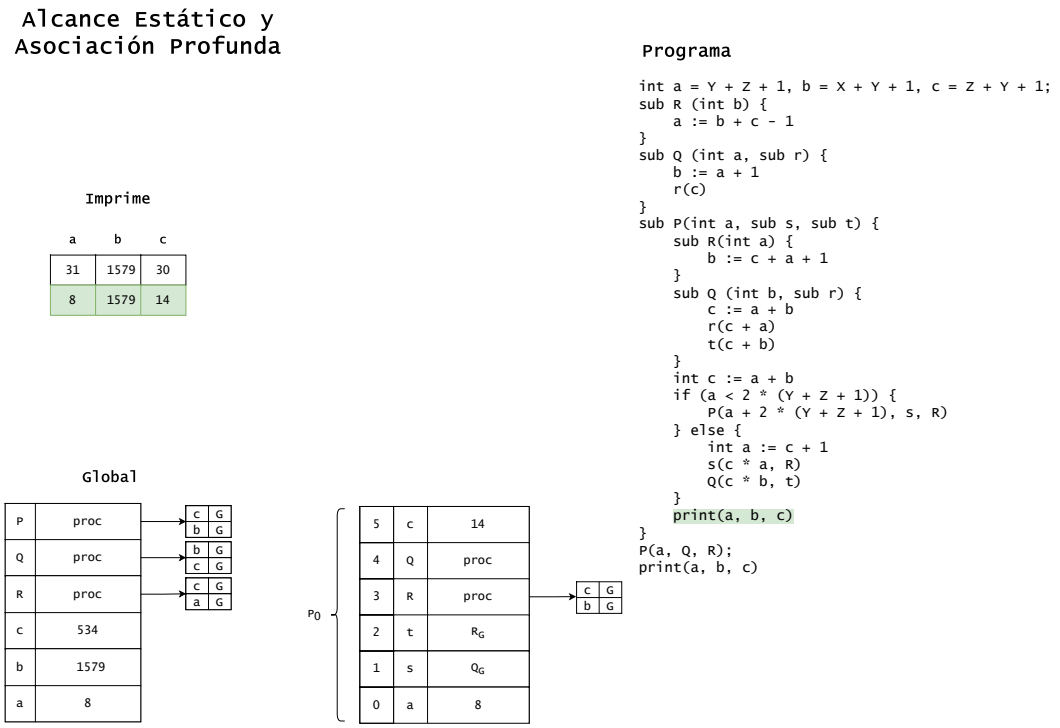

Figura 10: Alcance estático y asociación profunda - 10

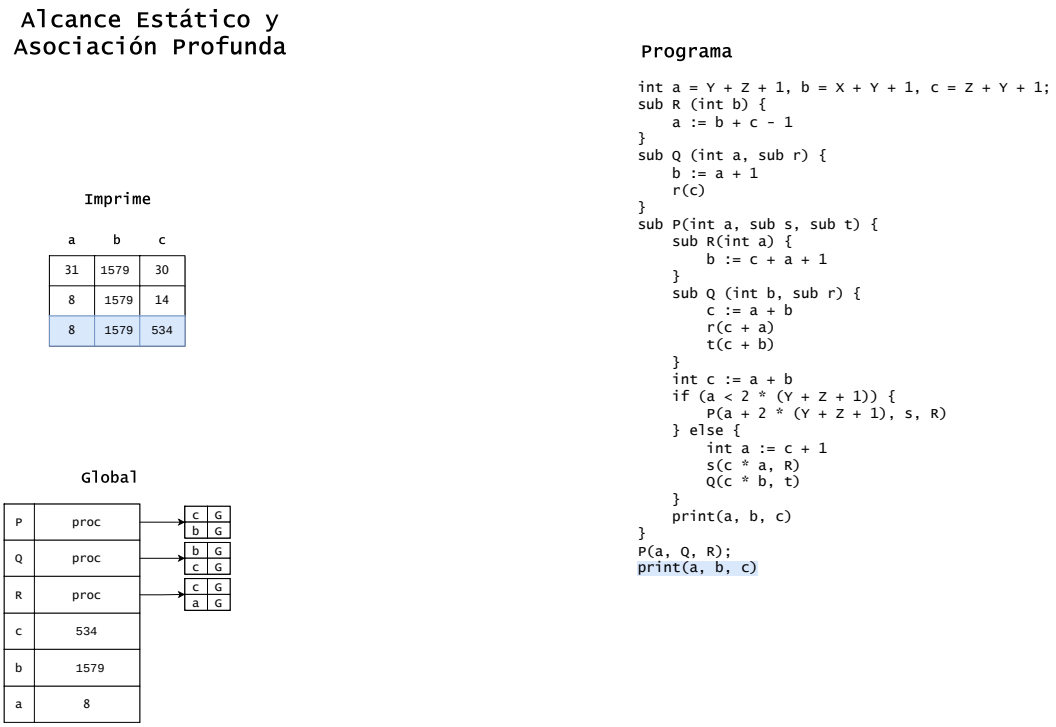

Figura 11: Alcance estático y asociación profunda - 11

(b) Alcance dinámico y asociación profunda - Tenemos $\mathbf{X = 2, Y = 3, Z = 4}$ por lo tanto, $\mathbf{a = Y + Z + 1 = 3 + 4 + 1 = 8}$, $\mathbf{b = X + Y + 1 = 2 + 3 + 1 = 6}$, $\mathbf{c = Z + Y + 1 = 4 + 3 + 1 = 8}$. A continuación se mostrará todo la pila de llamadas y los estados en cada momento.

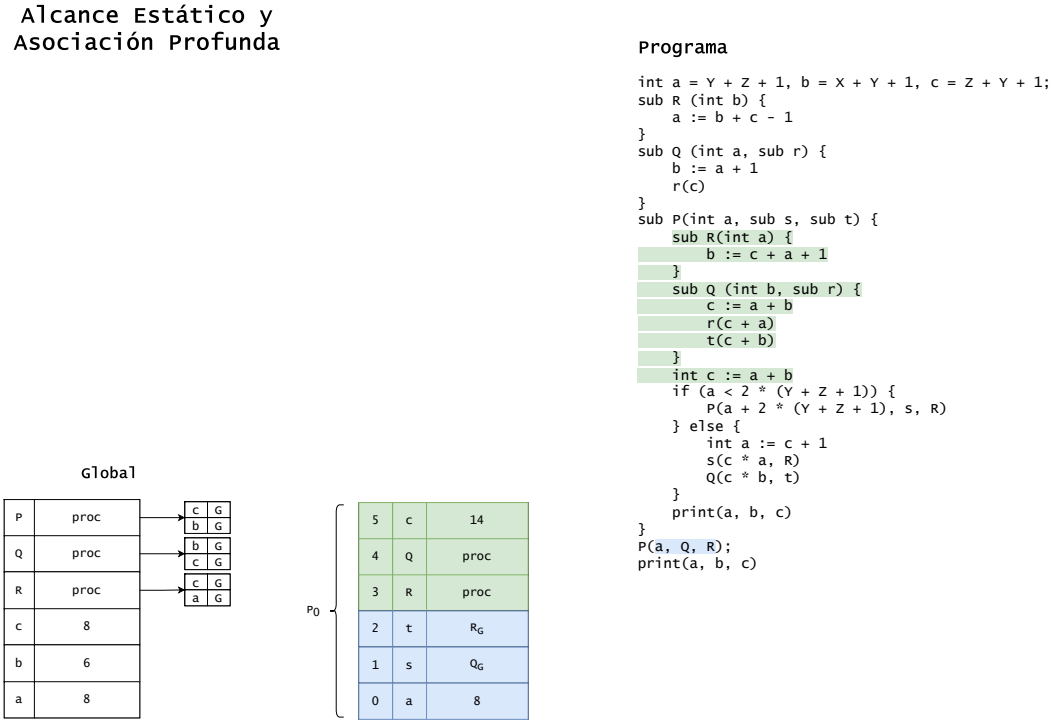

Figura 12: Alcance dinámico y asociación profunda - 1

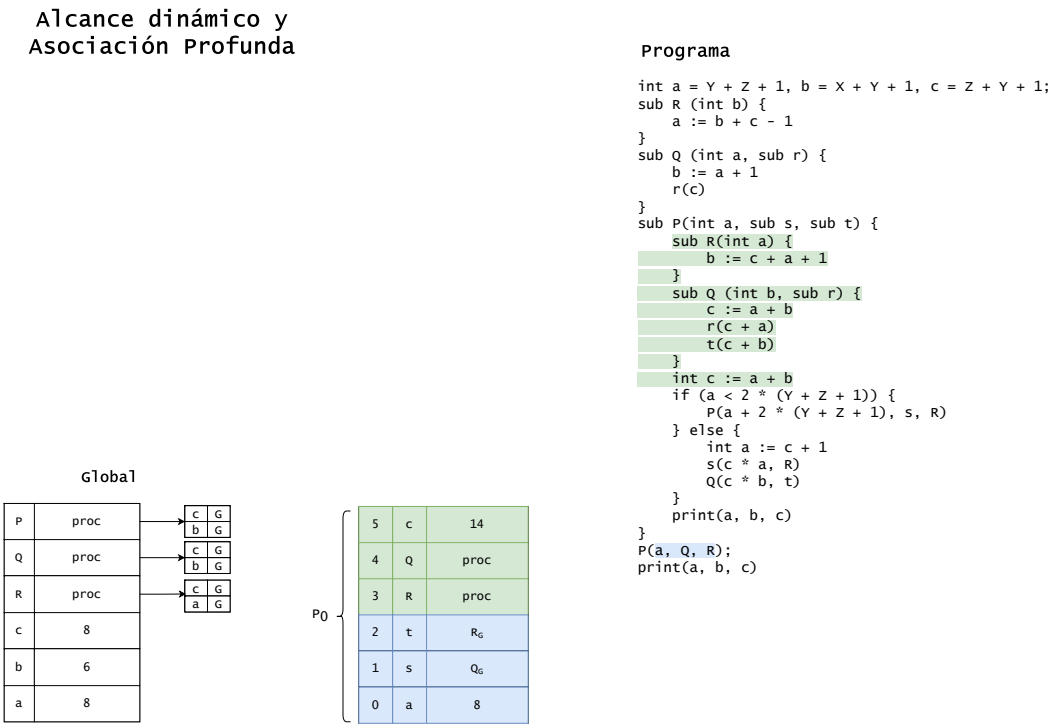

Figura 13: Alcance dinámico y asociación profunda - 2

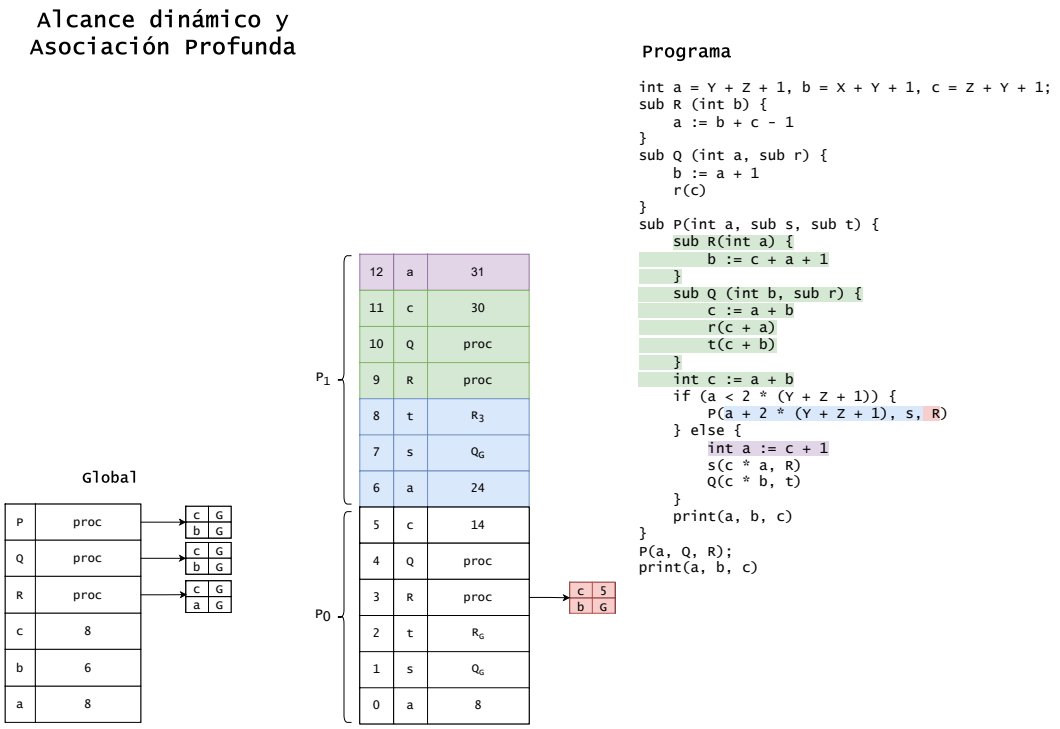

Figura 14: Alcance dinámico y asociación profunda - 3

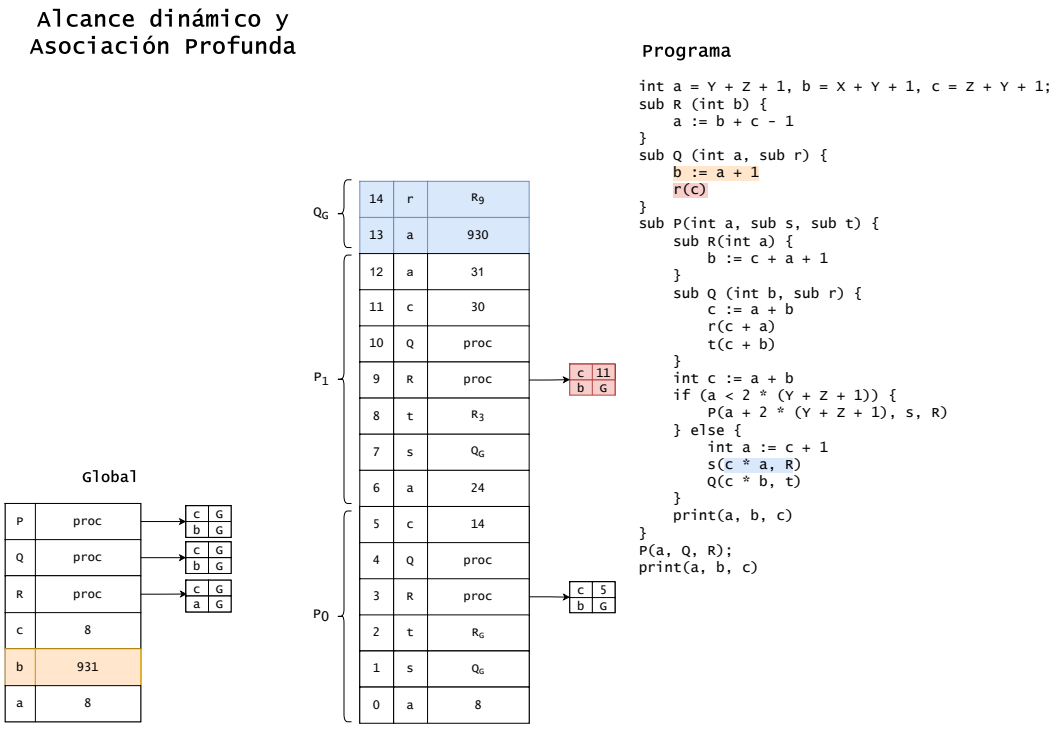

Figura 15: Alcance dinámico y asociación profunda - 4

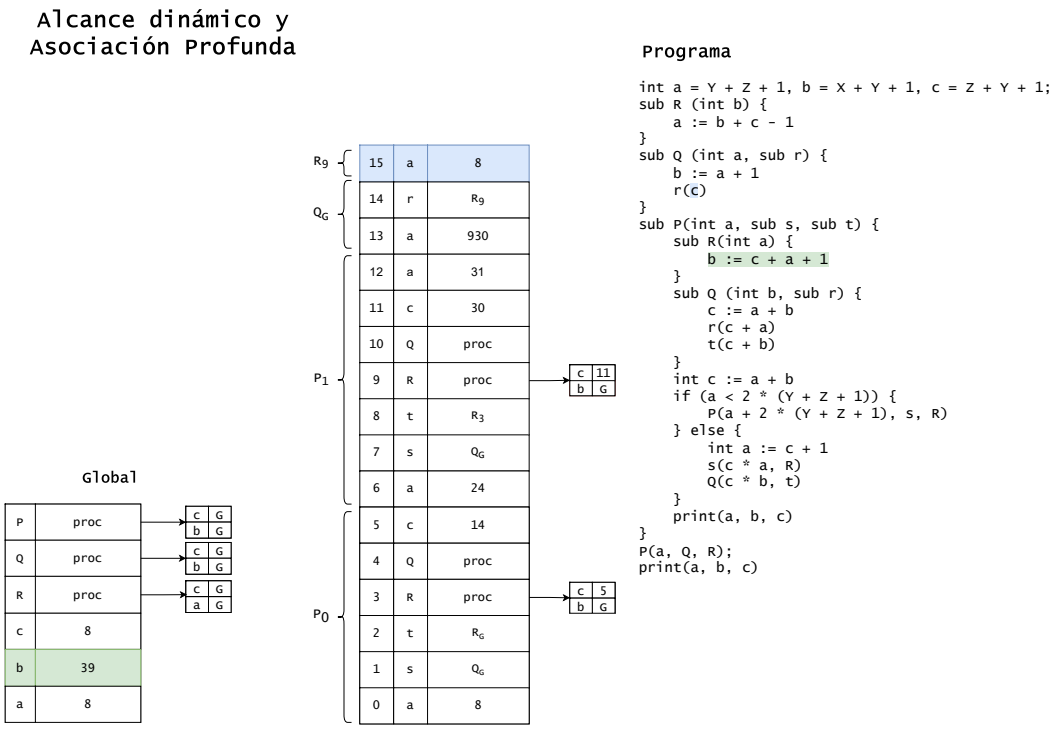

Figura 16: Alcance dinámico y asociación profunda - 5

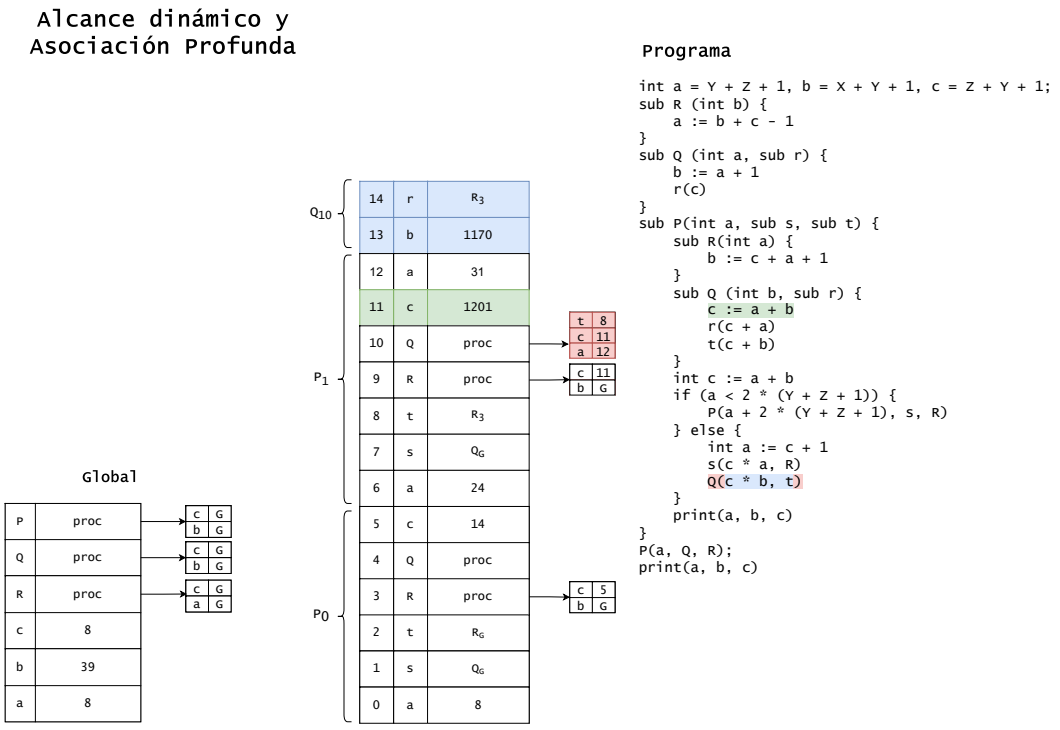

Figura 17: Alcance dinámico y asociación profunda - 6

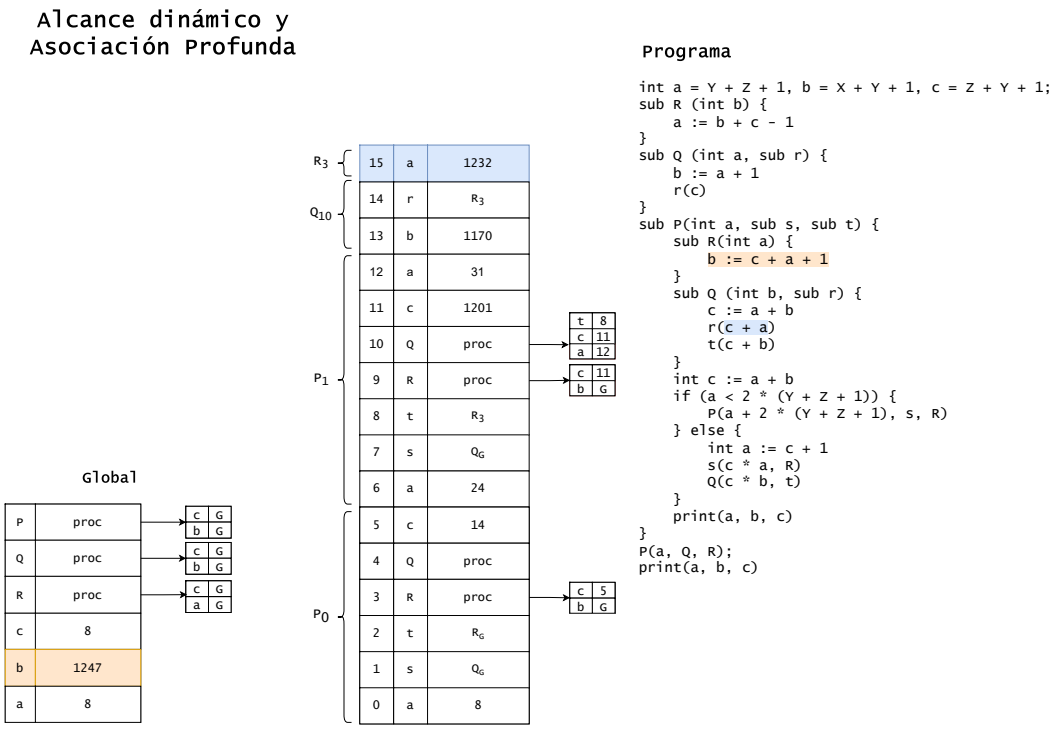

Figura 18: Alcance dinámico y asociación profunda - 7

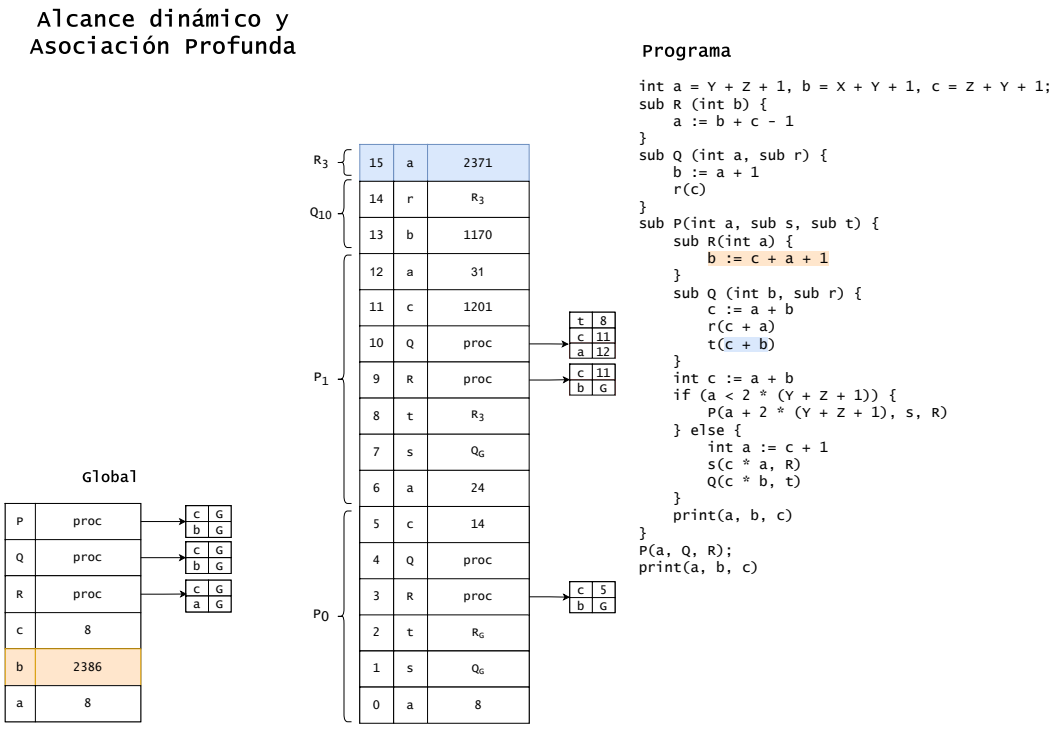

Figura 19: Alcance dinámico y asociación profunda - 8

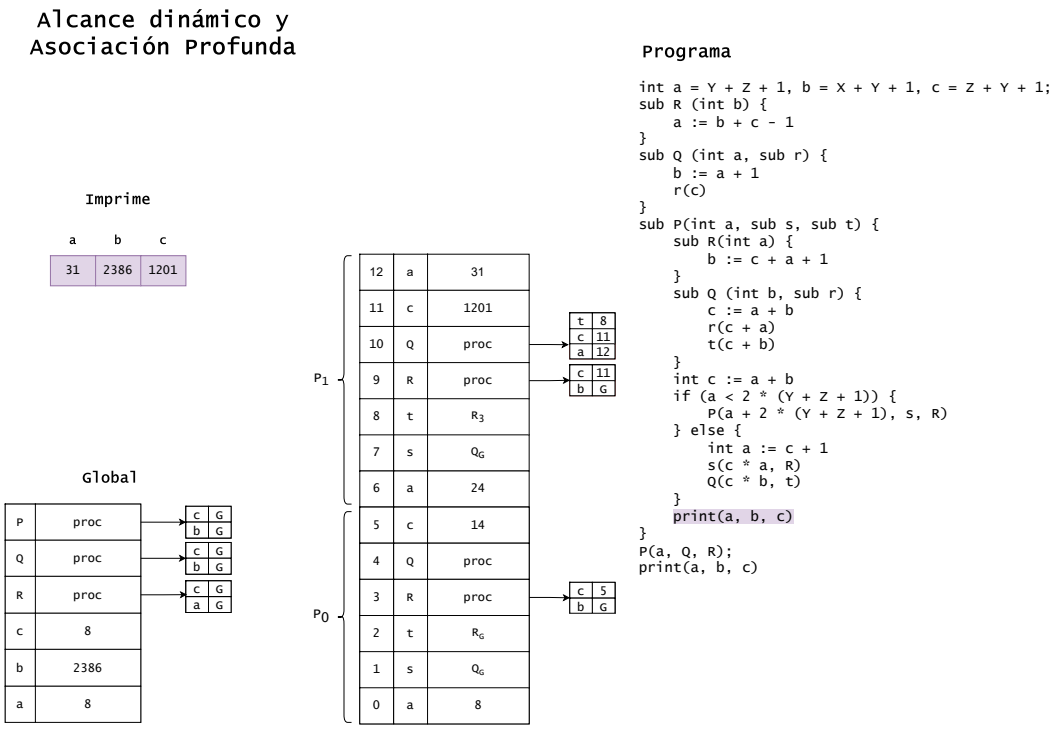

Figura 20: Alcance dinámico y asociación profunda - 9

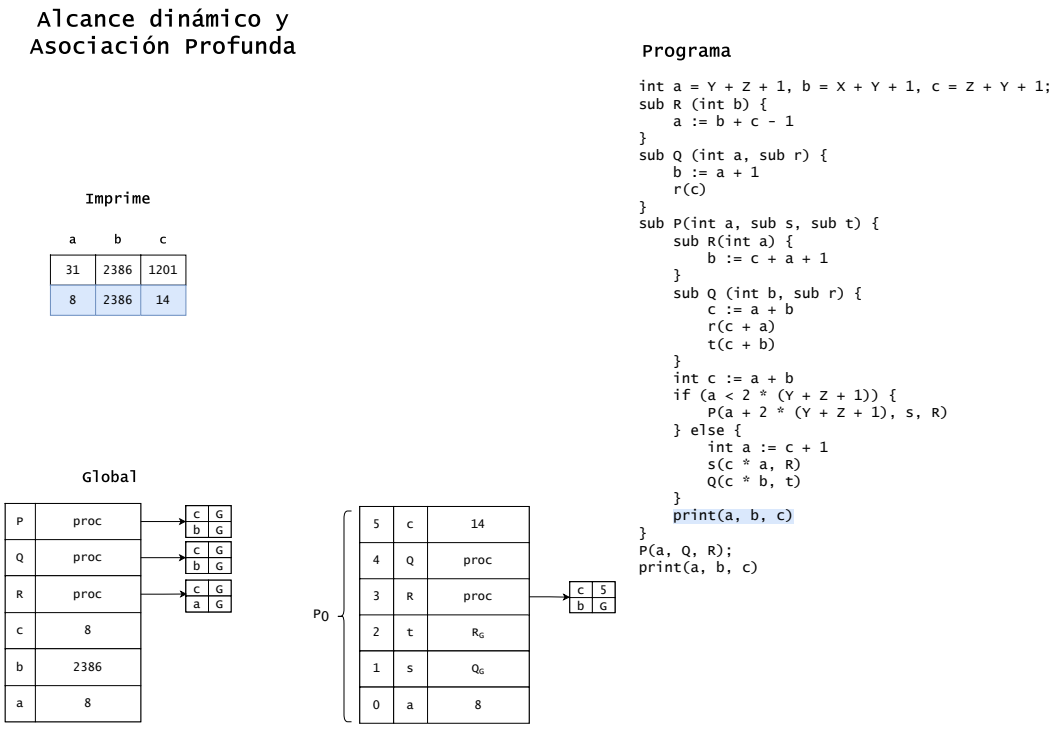

Figura 21: Alcance dinámico y asociación profunda - 10

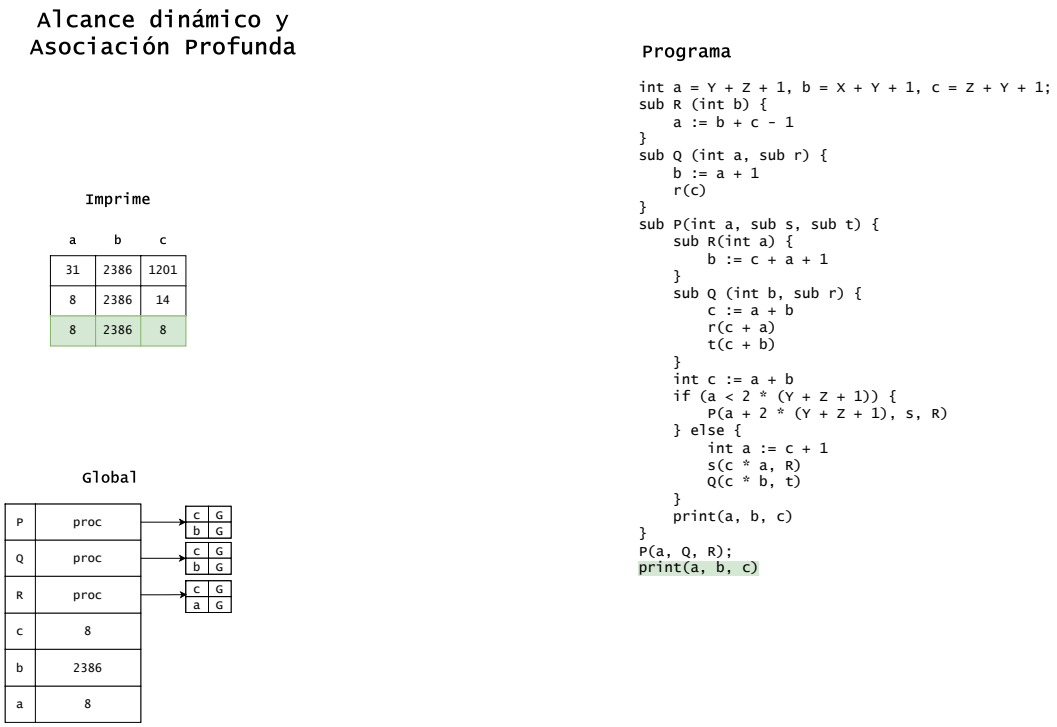

Figura 22: Alcance dinámico y asociación profunda - 11

(c) Alcance estático y asociación superficial - Tenemos $\mathbf{X = 2, Y = 3, Z = 4}$ por lo tanto, $\mathbf{a = Y + Z + 1 = 3 + 4 + 1 = 8}$, $\mathbf{b = X + Y + 1 = 2 + 3 + 1 = 6}$, $\mathbf{c = Z + Y + 1 = 4 + 3 + 1 = 8}$. A continuación se mostrará todo la pila de llamadas y los estados en cada momento.

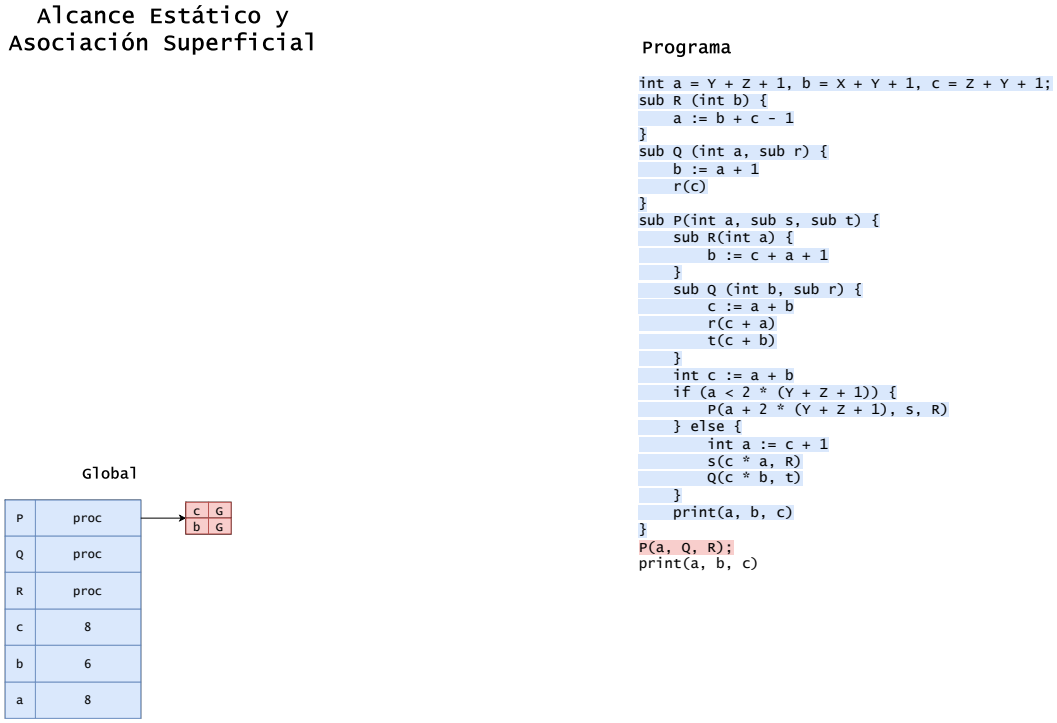

Figura 23: Alcance estático y asociación superficial - 1

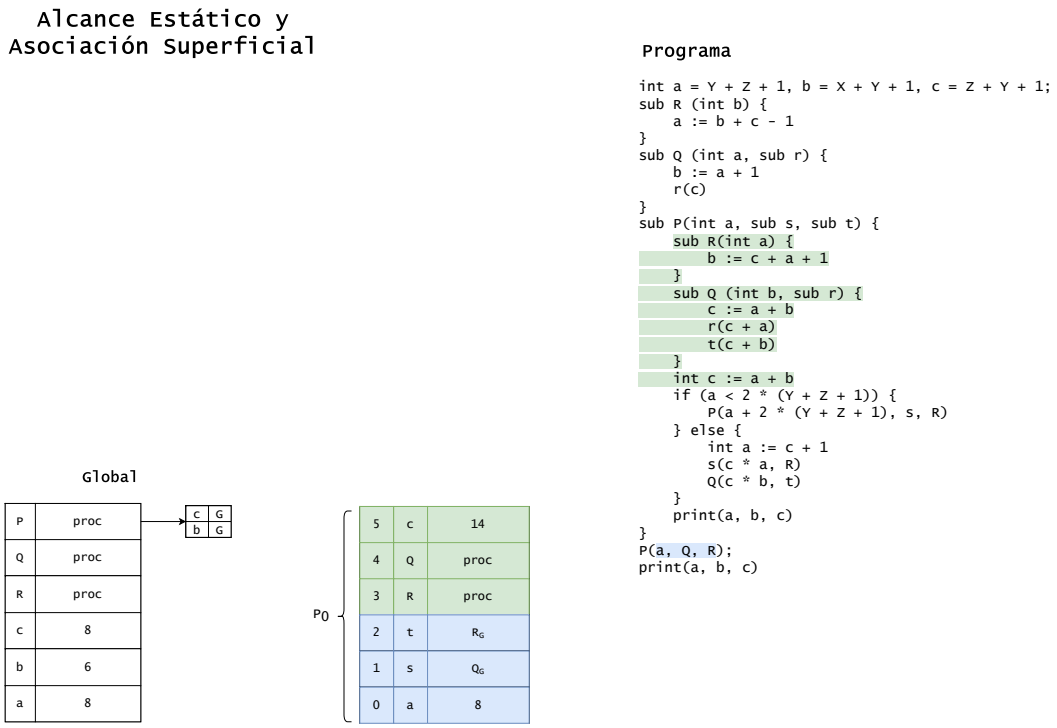

Figura 24: Alcance estático y asociación superficial - 2

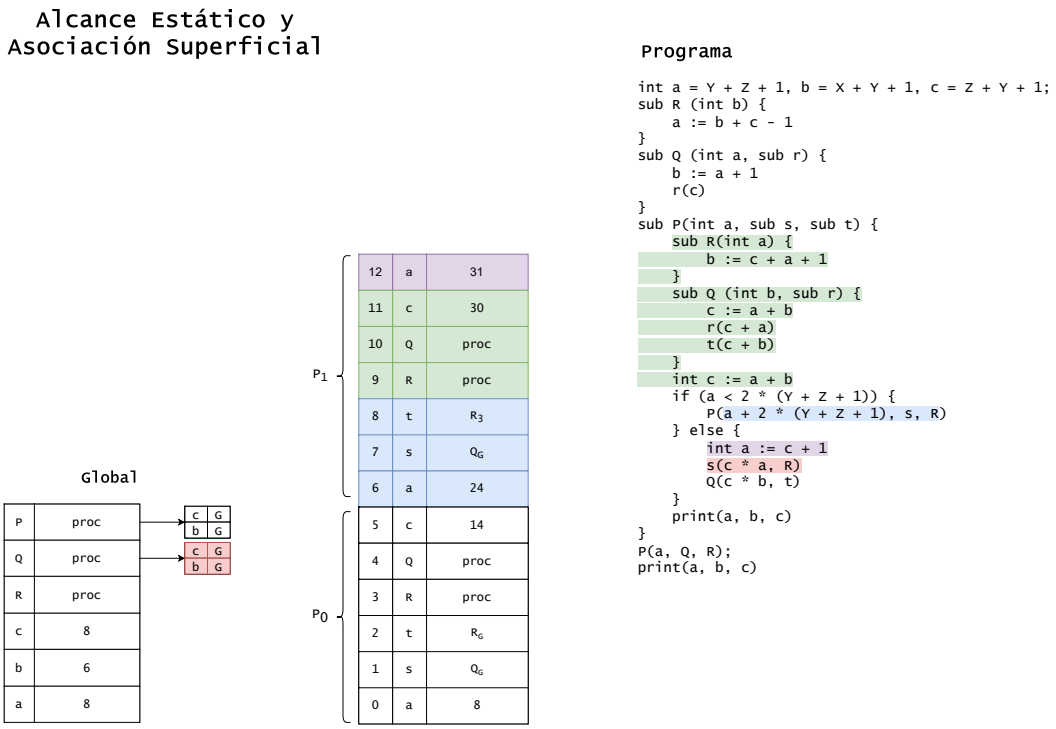

Figura 25: Alcance estático y asociación superficial - 3

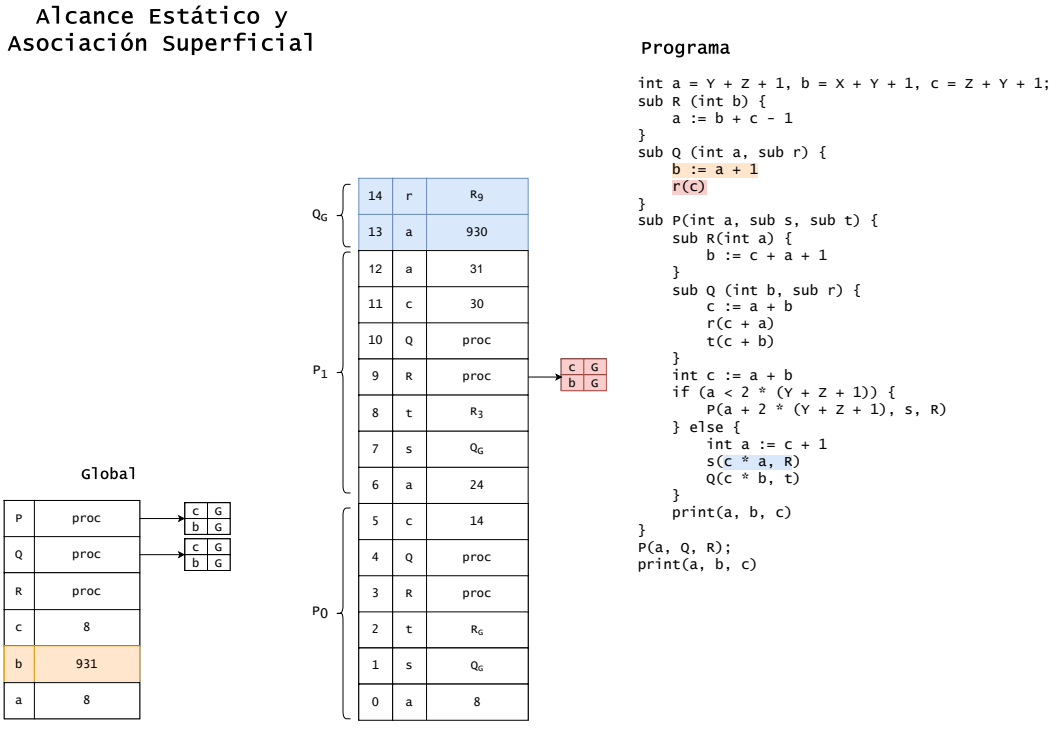

Figura 26: Alcance estático y asociación superficial - 4

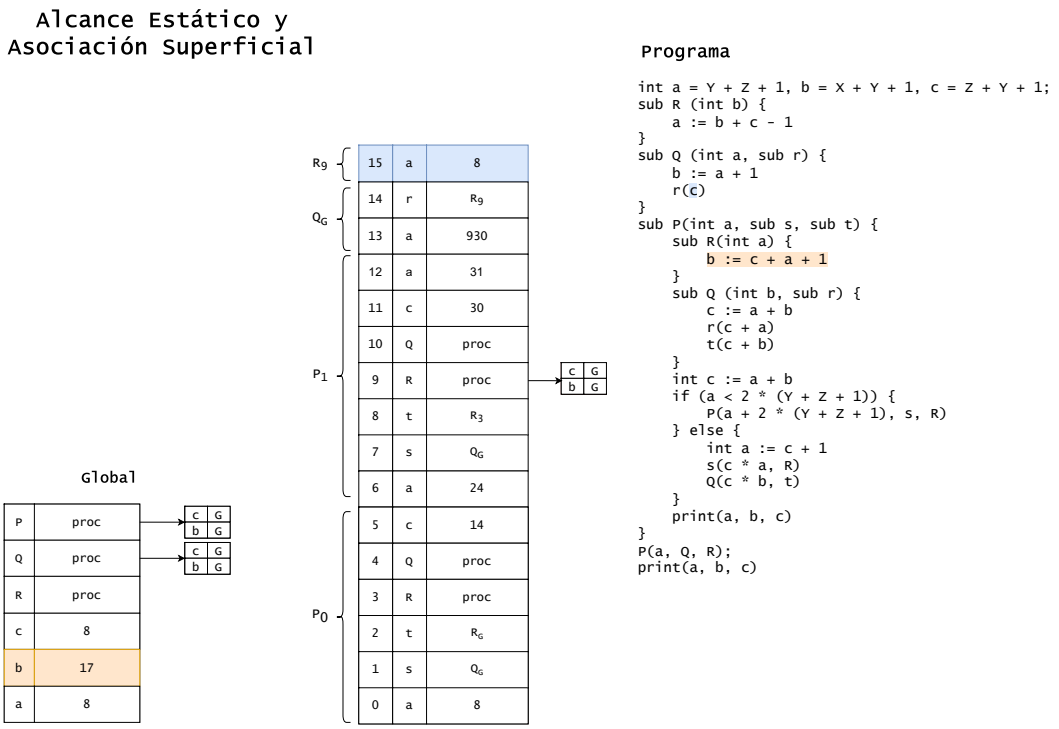

Figura 27: Alcance estático y asociación superficial - 5

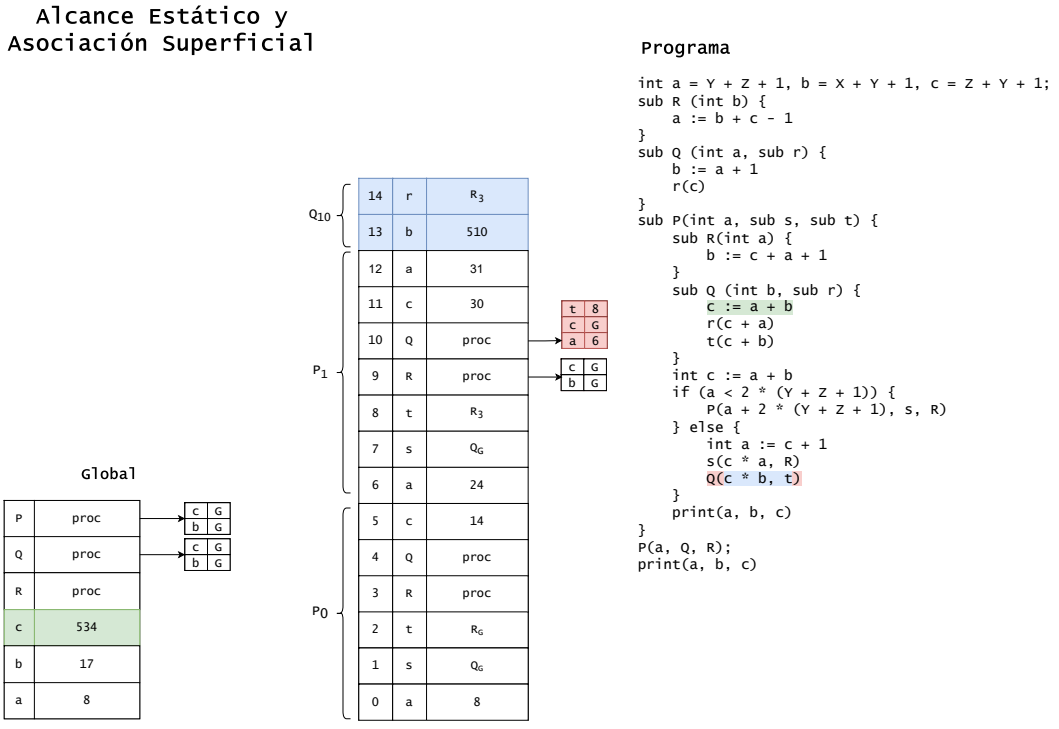

Figura 28: Alcance estático y asociación superficial - 6

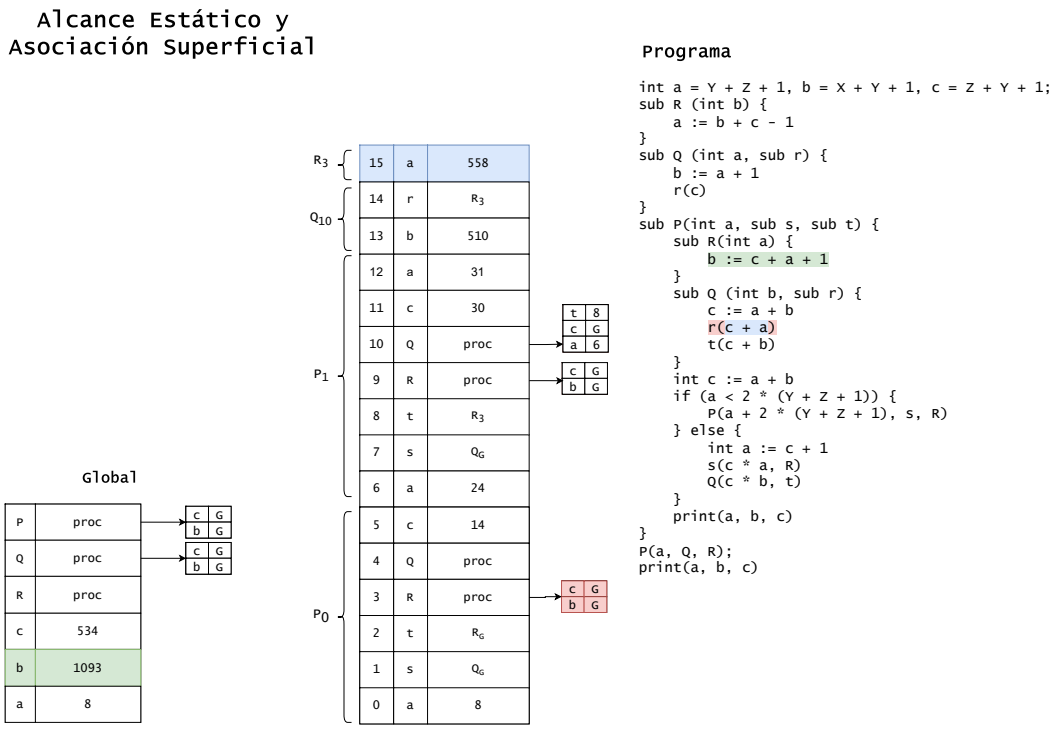

Figura 29: Alcance estático y asociación superficial - 7

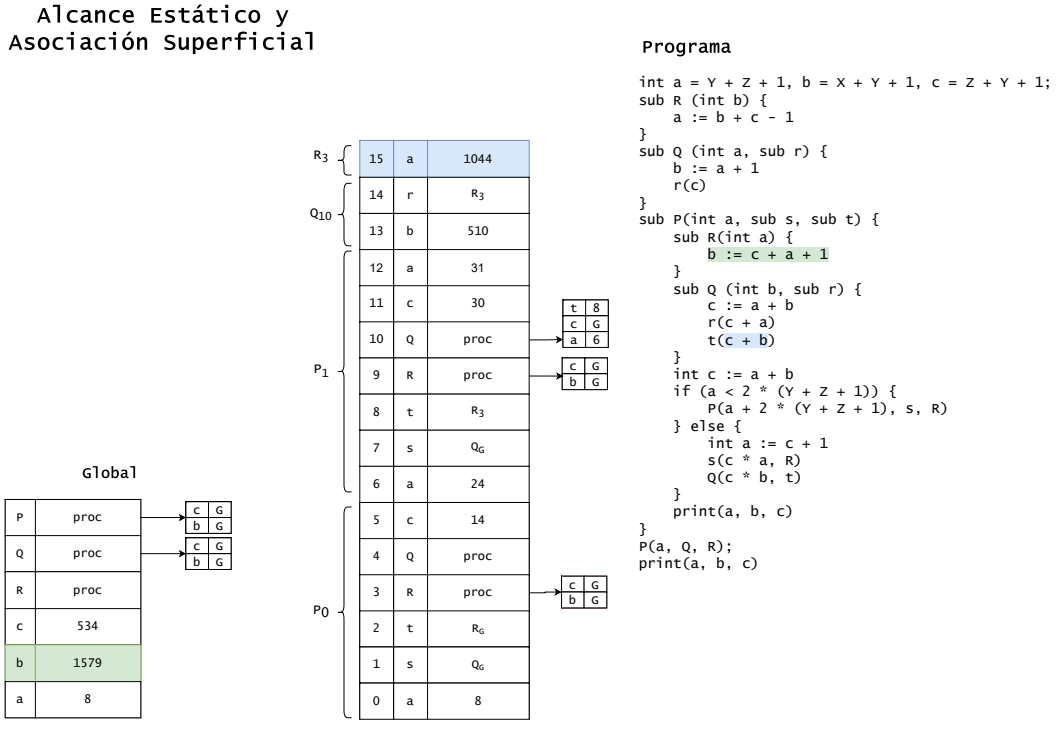

Figura 30: Alcance estático y asociación superficial - 8

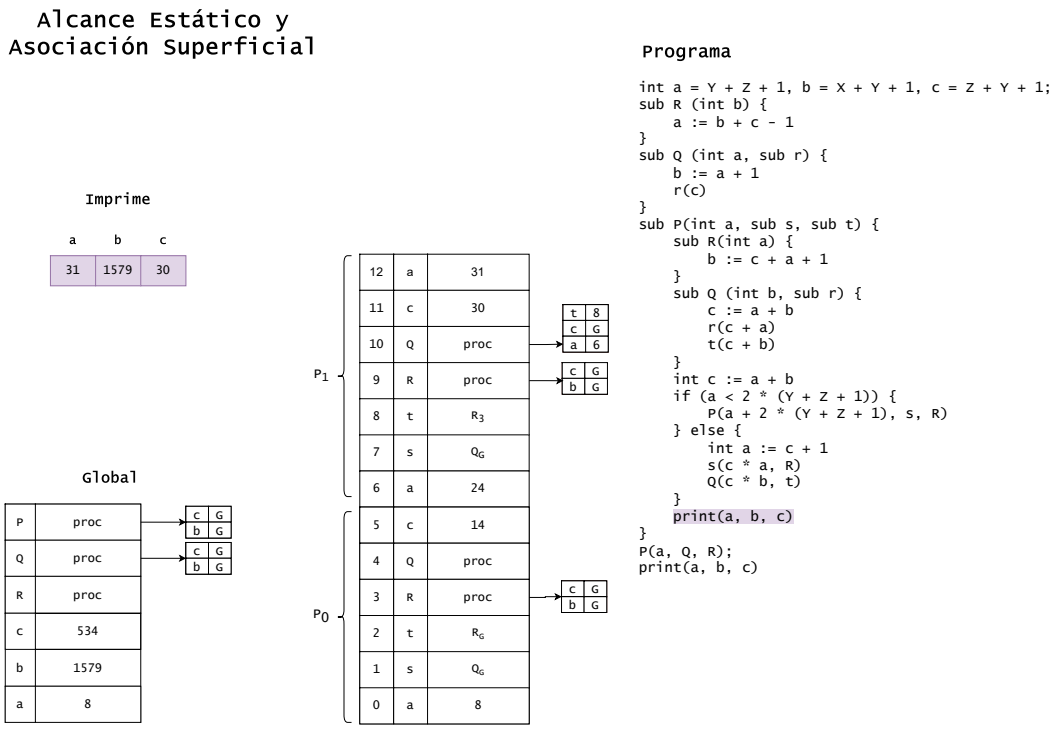

Figura 31: Alcance estático y asociación superficial - 9

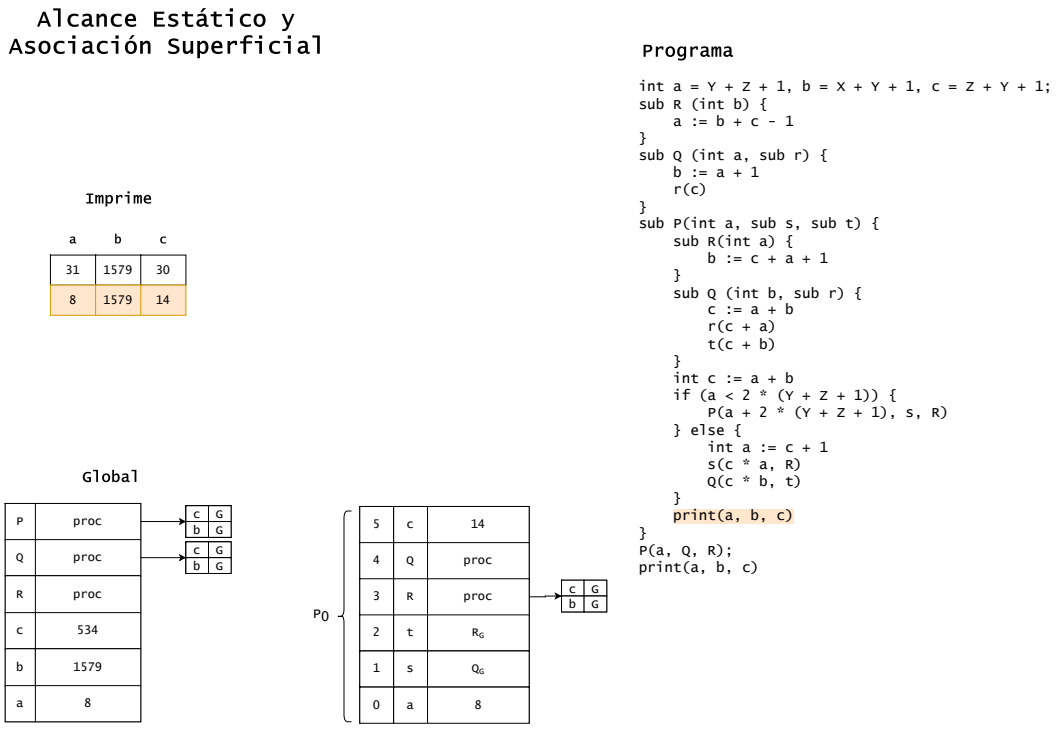

Figura 32: Alcance estático y asociación superficial - 10

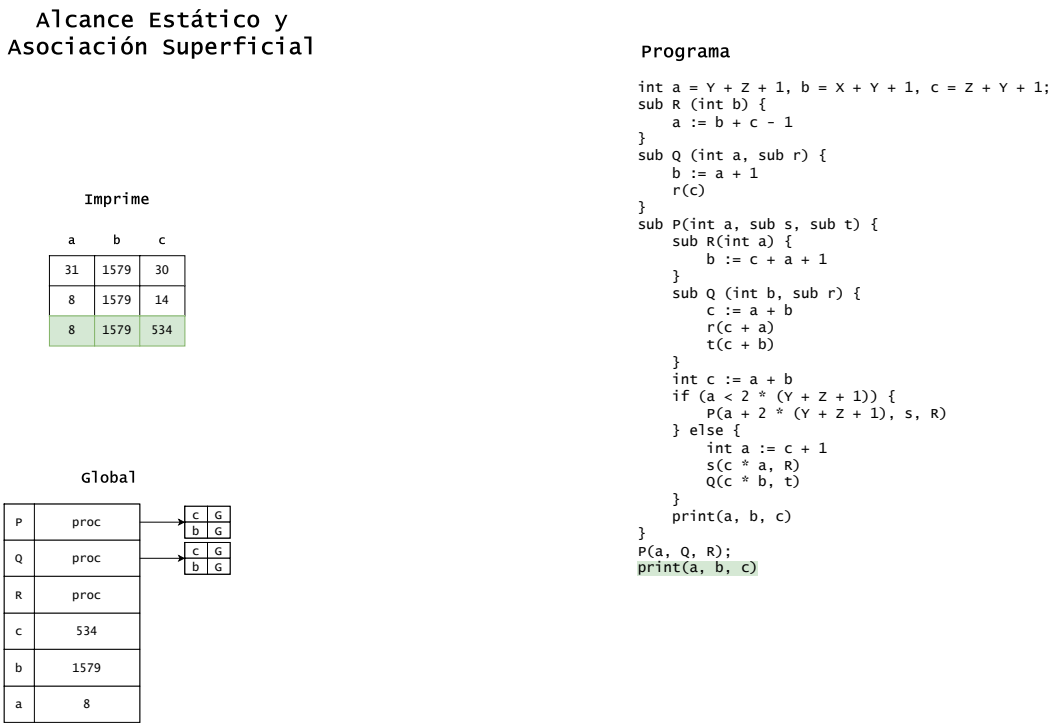

Figura 33: Alcance estático y asociación superficial - 11

(d) Alcance dinámico y asociación superficial - Tenemos $\mathbf{X = 2, Y = 3, Z = 4}$ por lo tanto, $\mathbf{a = Y + Z + 1 = 3 + 4 + 1 = 8}$, $\mathbf{b = X + Y + 1 = 2 + 3 + 1 = 6}$, $\mathbf{c = Z + Y + 1 = 4 + 3 + 1 = 8}$. A continuación se mostrará todo la pila de llamadas y los estados en cada momento.

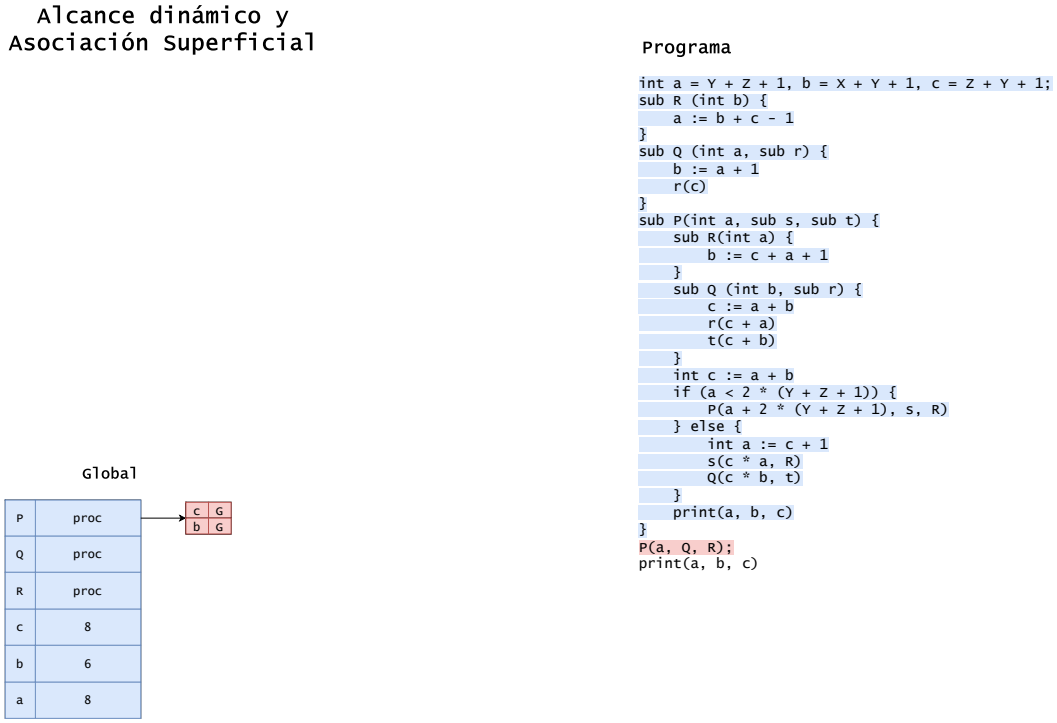

Figura 34: Alcance dinámico y asociación superficial - 1

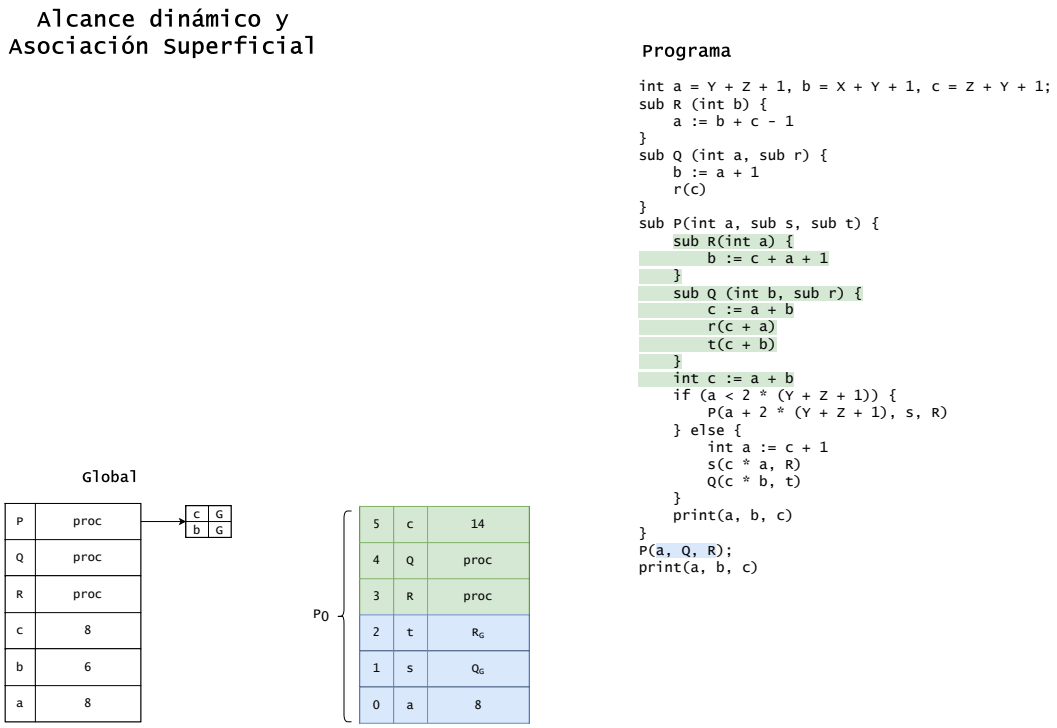

Figura 35: Alcance dinámico y asociación superficial - 2

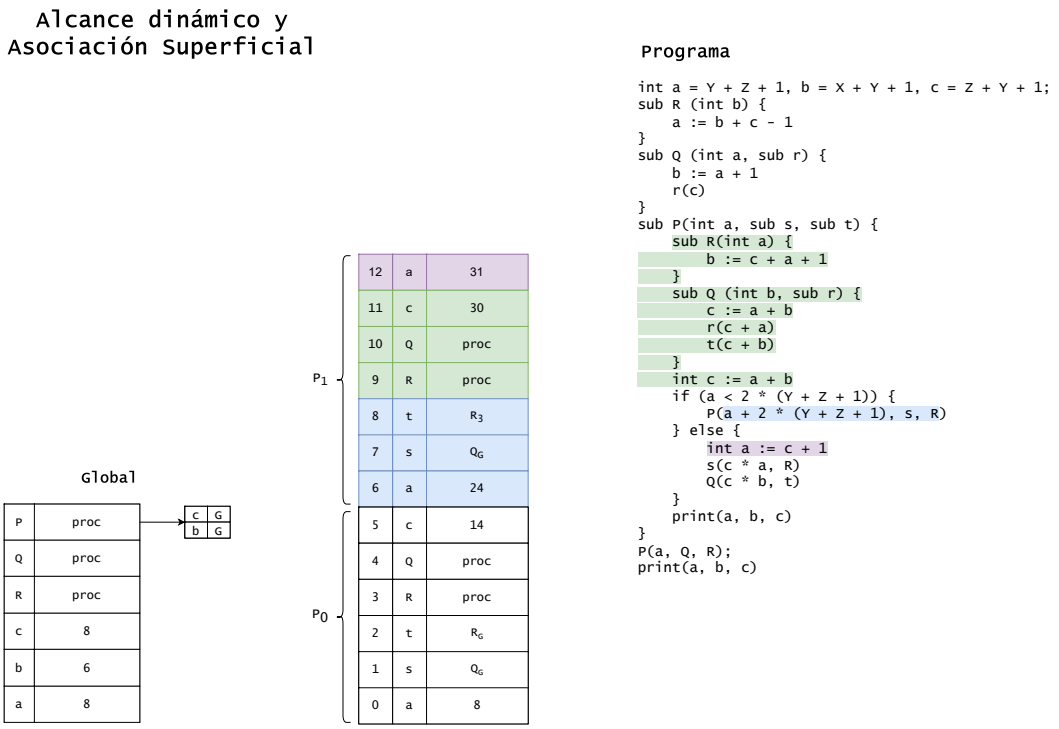

Figura 36: Alcance dinámico y asociación superficial - 3

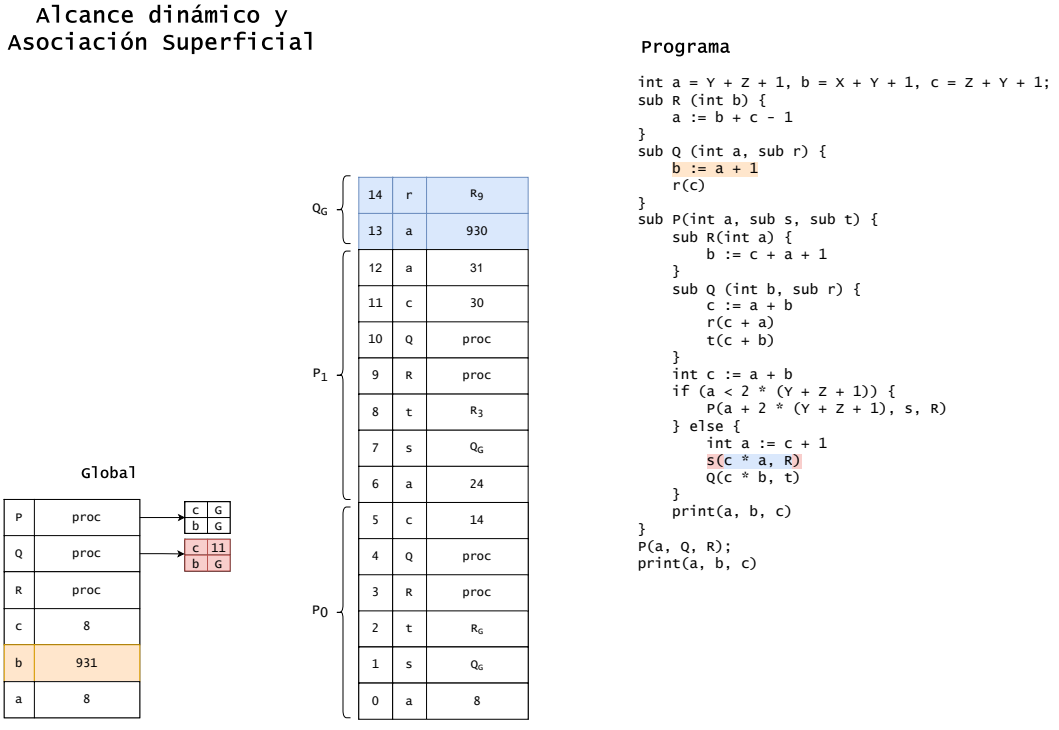

Figura 37: Alcance dinámico y asociación superficial - 4

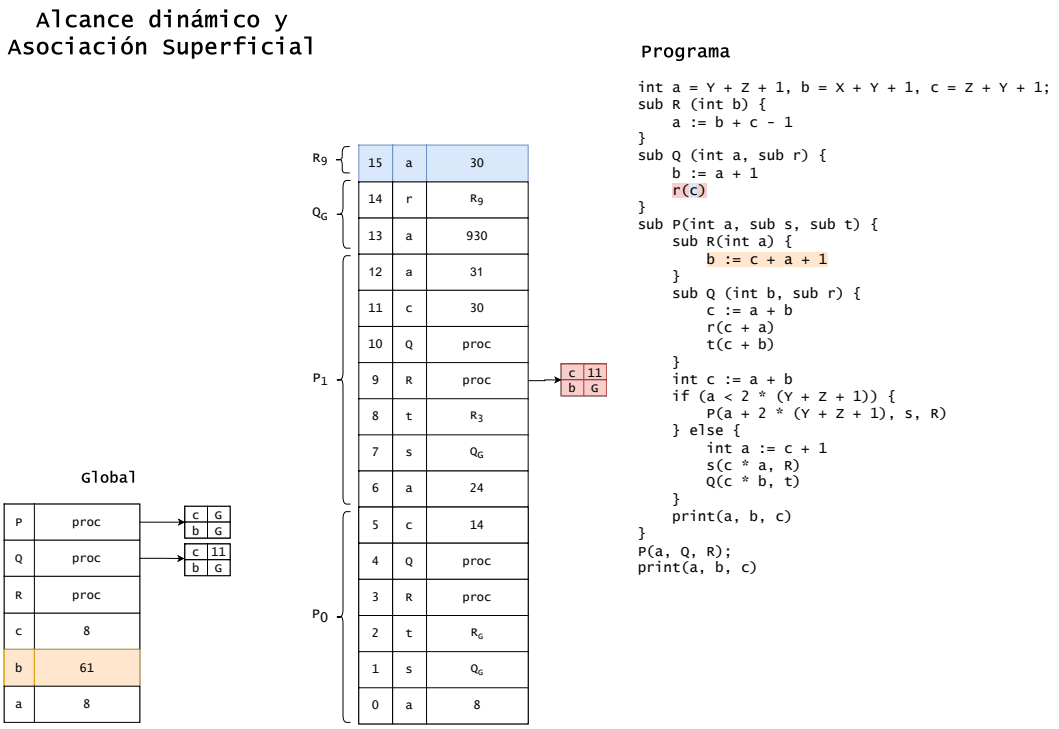

Figura 38: Alcance dinámico y asociación superficial - 5

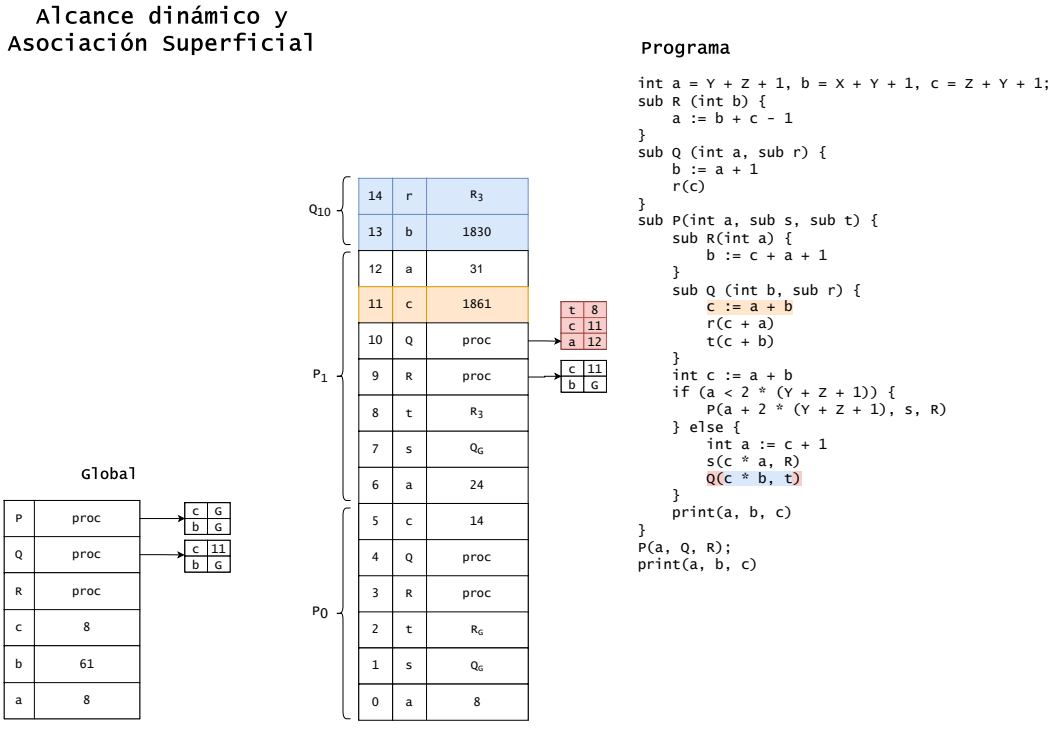

Figura 39: Alcance dinámico y asociación superficial - 6

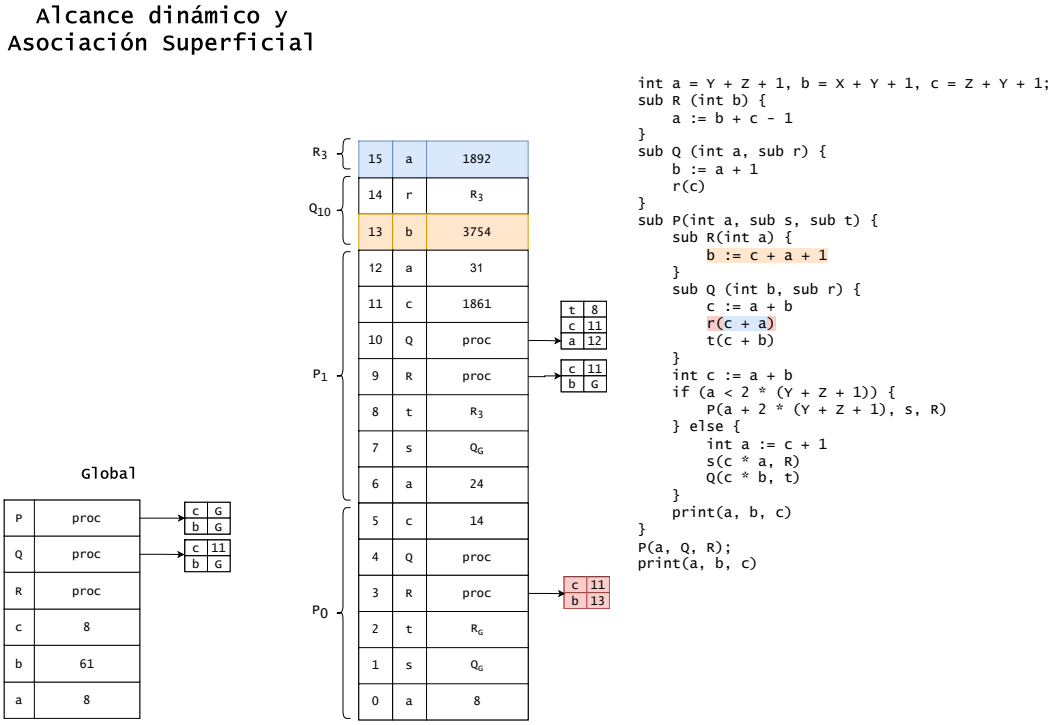

Figura 40: Alcance dinámico y asociación superficial - 7

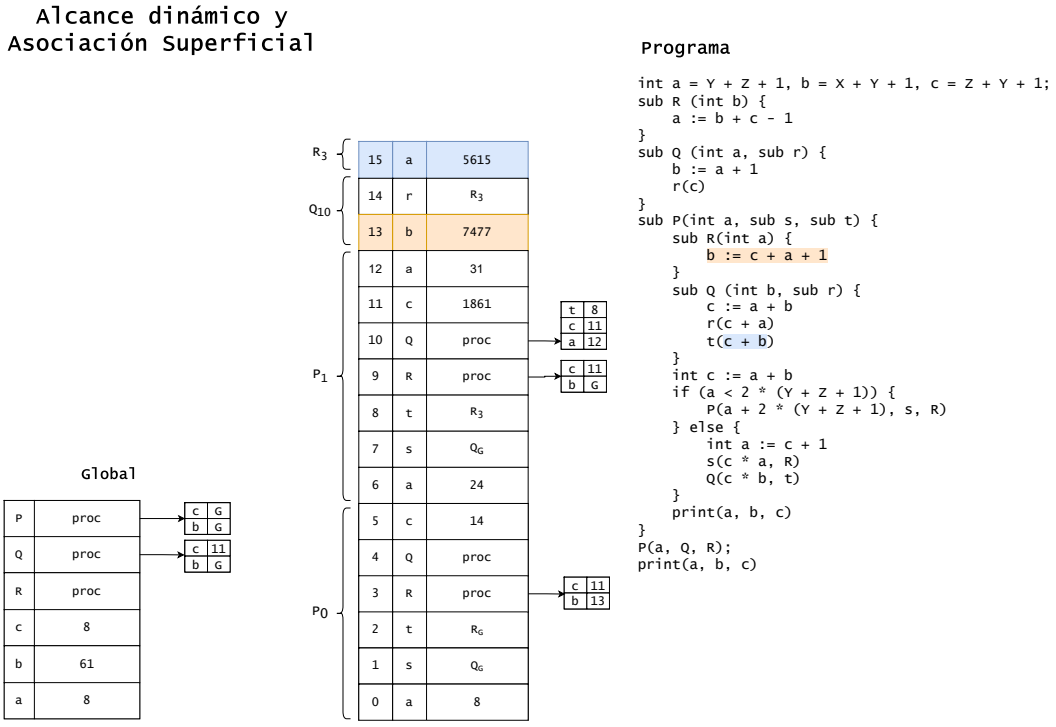

Figura 41: Alcance dinámico y asociación superficial - 8

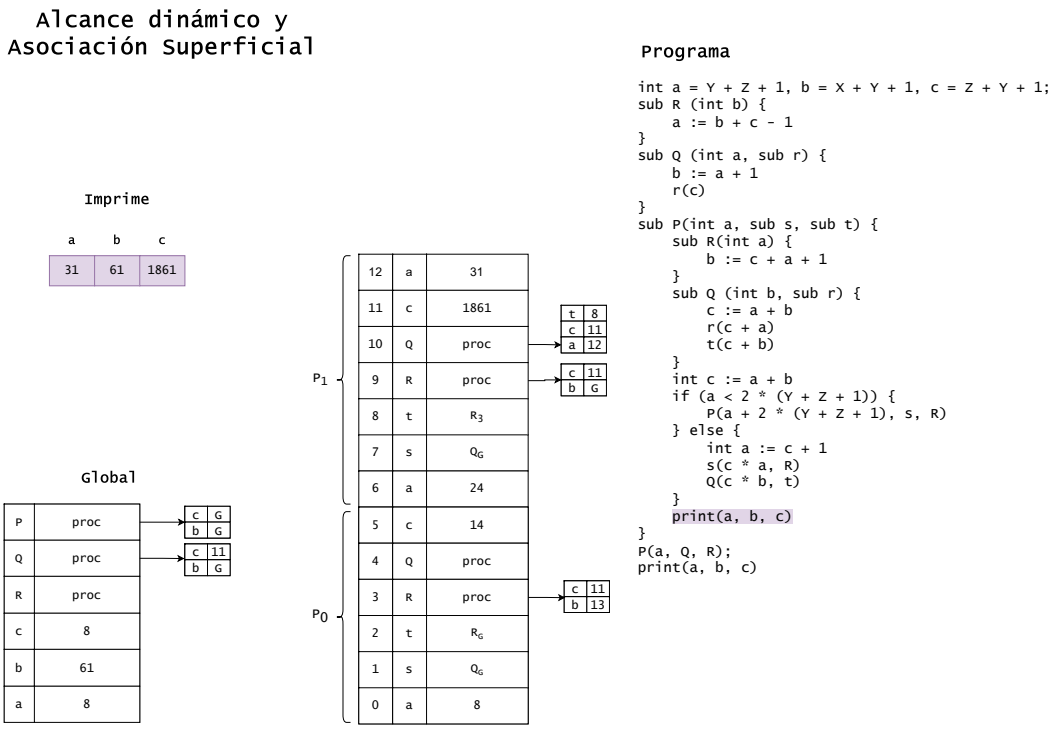

Figura 42: Alcance dinámico y asociación superficial - 9

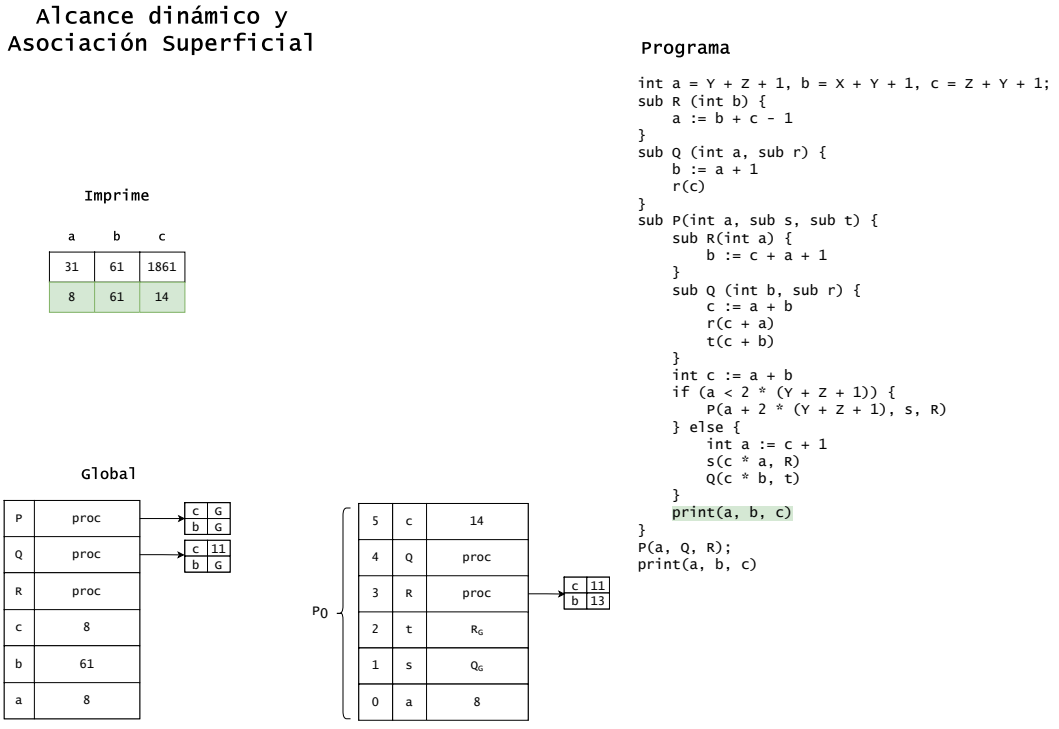

Figura 43: Alcance dinámico y asociación superficial - 10

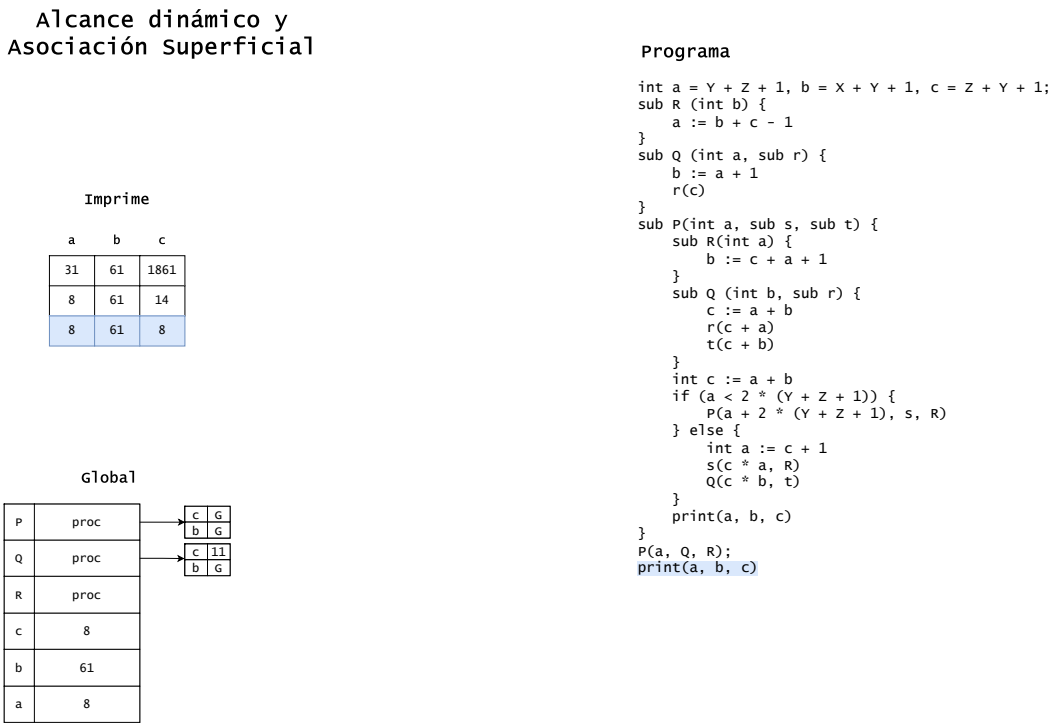

Figura 44: Alcance dinámico y asociación superficial - 11

3. Se desea que modele e implemente, en el lenguaje de su elección, un programa que simule un manejador de memoria que implementa el buddy system. Este programa debe cumplir con las siguientes características:

(a) Al ser invocado, recibirá como argumento la cantidad de bloques de memoria que manejará.

(b) Una vez iniciado el programa, pedirá repetidamente al usuario una acción para proceder. Tal acción puede ser:

i. RESERVAR <cantidad> ¡nombre> Representa una reserva de espacio de ¡cantidad¿bloques, asociados al identificador <nombre>.

El programa debe reportar un error e ignorar la acción si ¡nombre¿ya tiene memoria reservada o no hay un espacio libre contiguo suficientemente grande como para satisfacer la petición (en cualquier caso, el mensaje de error debe ser claro e informativo).

ii. LIBERAR <nombre> Representa una liberación del espacio que contiene el identificador <nombre>. El programa debe reportar un error e ignorar la acción si <nombre> no tiene memoria reservada (el mensaje de error debe ser claro e informativo).

iii. MOSTRAR Debe mostrar una representación gráfica (en texto) de las listas de bloques libres, así como la información de nombres y la memoria que tienen asociada a los mismos.

iv. SALIR Debe salir del simulador. Al finalizar la ejecución de cada acción, el programa deberá pedir la siguiente acción al usuario.

Investigue herramientas para pruebas unitarias y cobertura en su lenguaje escogido y agregue pruebas a su programa que permitan corroborar su correcto funcionamiento. Como regla general, su programa debería tener una cobertura (de líneas de código y de bifuración) mayor al 80 %.

Respuesta:

El código fuente de la resolución del problema se encuentra en el siguiente repositorio de GitHub

Repositorio Examen 1

En el repositorio se especifica como ejecutar el programa, además contiene el informe de cobertura derivado de los test realizados.

4. Se desea que modele e implemente, en el lenguaje de su elección, un módulo que defina el tipo de vectores de tres (3) dimensiones y operadores aritméticas sobre estos. La librería debe cumplir con las siguientes características: (a) Debe saber tratar las siguientes expresiones aritméticas:

- suma: representada por el símbolo +.
- resta: representada por el símbolo -.
- producto cruz: representada por el símbolo *.
- producto punto: representada por el símbolo %.
- norma: representada por el símbolo &.

(b) Los vectores deben poder operarse entre sí sin necesidad de que el usuario de la librería escriba llamadas adicionales a funciones. Por ejemplo, si a, b y c son vectores, las siguientes expresiones deben ser válidas para un usuario de la librería:

b + c

a * b + c

(b + b) * (c - a)

a % (c * b)

(c) Los vectores deben poder operarse también con números enteros o flotantes por la derecha (menos para el producto punto) sin necesidad de que el usuario de la librería escriba llamadas adicionales a funciones. Por ejemplo, si a, b y c son vectores, las siguientes instrucciones deben ser válidas para un usuario de la librería:

b + 3

a * 3.0 + &b

(b + b) * (c % a)

En este caso, el resultado debe ser el vector operado en sus 3 dimensiones con la operación propuesta. Esto es:

$(x; y; z) \oplus n = (x \oplus n; y \oplus n; z \oplus n)$

Investigue herramientas para pruebas unitarias y cobertura en su lenguaje escogido y agregue pruebas a su programa que permitan corroborar su correcto funcionamiento. Como regla general, su programa debería tener una cobertura (de líneas de código y de bifurcación) mayor al 80 %.

Respuesta:

El código fuente de la resolución del problema se encuentra en el siguiente repositorio de GitHub

Repositorio Examen 1

En el repositorio se especifica como ejecutar el programa, además contiene el informe de cobertura derivado de los test realizados.

5. Se desea que modele e implemente, en el lenguaje de su elección, un programa que simule programas, intérpretes y traductores como los vistos al estudiar los diagramas de T. Este programa debe cumplir con las siguientes características: (a) Debe poder manejar programas, intérpretes y traductores. Estos pueden ser:

- PROGRAMA <nombre> <lenguaje> Representa a un programa identificado por <nombre> escrito en <lenguaje>.
- INTERPRETE <lenguaje_base> <lenguaje> Representa a un intérprete para <lenguaje> escrito en <lenguaje_base>.
- TRADUCTOR <lenguaje_base> <lenguaje_origen> <lenguaje_destino> Representa a un traductor, desde <lenguaje_origen> hacia <lenguaje_destino>, escrito en <lenguaje_base>.

(b) Todos los lenguajes deben ser cadenas alfanuméricas y no tienen por qué corresponder a algún lenguaje que exista.

(c) Existirá una nombre especial LOCAL que se referirá al lenguaje que puede interpretar la máquina local.

(d) Una vez iniciado el programa, pedirá repetidamente al usuario una acción para proceder.

Tal acción puede ser:

i. DEFINIR <tipo> [<argumentos>]

Representa una definición de clase <tipo> con <argumentos> (ver arriba los tipos y argumentos que se deben soportar). El programa debe reportar un error e ignorar la acción si <nombre> ya tiene un programa asociado, en el caso de programas.

ii. EJECUTABLE <nombre>

Representa una consulta de la posibilidad de ejecutar el programa de nombre ¡nombre¿. Su programa debe imprimir si es posible construir lo que se pide, usando únicamente las definiciones hechas hasta el momento.

El programa debe reportar un error e ignorar la acción si ¡nombre¿no tiene un programa asociado.

iii. SALIR

Debe salir del simulador. Al finalizar la ejecución de cada acción, el programa deberá pedir la siguiente acción al usuario

Respuesta:

El código fuente de la resolución del problema se encuentra en el siguiente repositorio de GitHub

Repositorio Examen 1

En el repositorio se especifica como ejecutar el programa, además contiene el informe de cobertura derivado de los test realizados.