

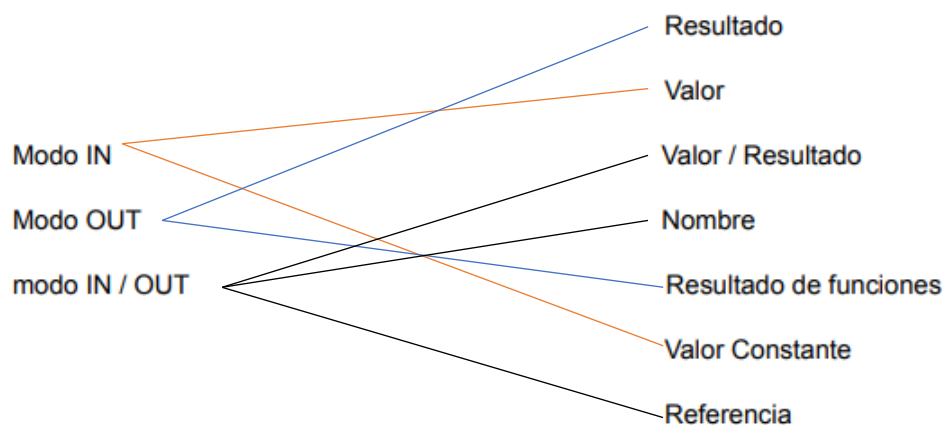
Práctica 6

Ejercicio 1:

a- Explique brevemente los siguientes conceptos

- **Parámetro:** es una forma de compartir datos entre diferentes unidades. Es la más flexible y permite la transferencia de diferentes datos en cada llamada. Proporciona ventajas en legibilidad y modificabilidad. Nos permiten compartir los datos en forma abstracta ya que indican con precisión qué es exactamente lo que se comparte
- **Parámetro real:** es un valor u otra entidad utilizada para pasar a un procedimiento o función. Están en la parte de la invocación
- **Parámetro formal:** es una variable utilizada para recibir valores de entrada en una rutina, subrutina etc. Se ponen en la parte de la declaración. Es una variable local a su entorno.
- **Ligadura posicional:** los parámetros formales y reales se ligan según la posición en la llamada y en la declaración.
- **Ligadura por palabra clave o nombre:** los parámetros formales y reales se ligan por el nombre. Se debe conocer los nombres de los parámetros formales.

Ejercicio 2: Unir los siguientes puntos según corresponda y de una definición y un ejemplo de cada par.



Ejercicio 3:

a- Complete el siguiente cuadro según lo correspondiente a cada lenguaje:

Tipo de pasaje de parámetros	Lenguaje
<ul style="list-style-type: none">• Por defecto con copia IN• Por resultado OUT• IN OUT	ADA

<ul style="list-style-type: none"> ○ Para los tipos primitivos indica por valor-resultado ○ Para los tipos no primitivos, datos compuestos (arreglo , registro) se hace por referencia. 	
<ul style="list-style-type: none"> • Por valor (Si se necesita por referencia se usa punteros) • Permite pasaje por valor constante, agregando const 	C
<ul style="list-style-type: none"> • Por valor, pero si se pasa un objeto "mutable", no se hace una copia sino que se trabaja sobre él. 	Ruby
<ul style="list-style-type: none"> • El único mecanismo contemplado es el paso por copia de valor. Pero como las variables de tipo no primitivos son todas referencias a variables anónimas en la heap, el paso por valor de una de estas variables son en realidad un paso por referencia de las variables 	JAVA
<ul style="list-style-type: none"> • Se puede pasar de dos formas: <ul style="list-style-type: none"> ○ Inmutables: actuara como por valor ○ Mutables: No se hace una copia sino que se trabaja sobre él. 	Python

b- Ada es más seguro que Pascal, respecto al pasaje de parámetros en las funciones. Explique por qué.

Ada es más seguro que Pascal en cuanto al pasaje de parámetros en las funciones debido a su sistema de tipos más estricto y su sintaxis más clara y explícita para especificar los modos de paso de parámetros.

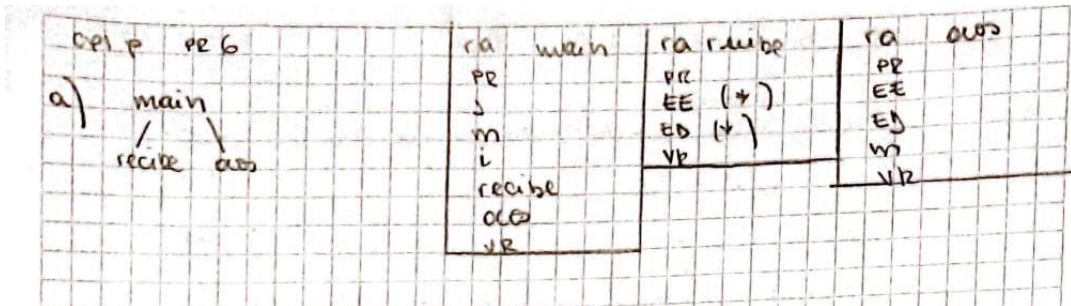
c- Explique cómo maneja Ada los tipos de parámetros in-out de acuerdo al tipo de dato

Ada maneja los parámetros in-out de acuerdo al tipo de dato que se está utilizando, utilizando una técnica de paso por referencia para tipos de datos simples, y una técnica de copia y devolución para tipos de datos más complejos. Esto permite un manejo seguro y eficiente de los parámetros in-out en los programas Ada.

Ejercicio 4: Sea el siguiente programa escrito en Pascal-like

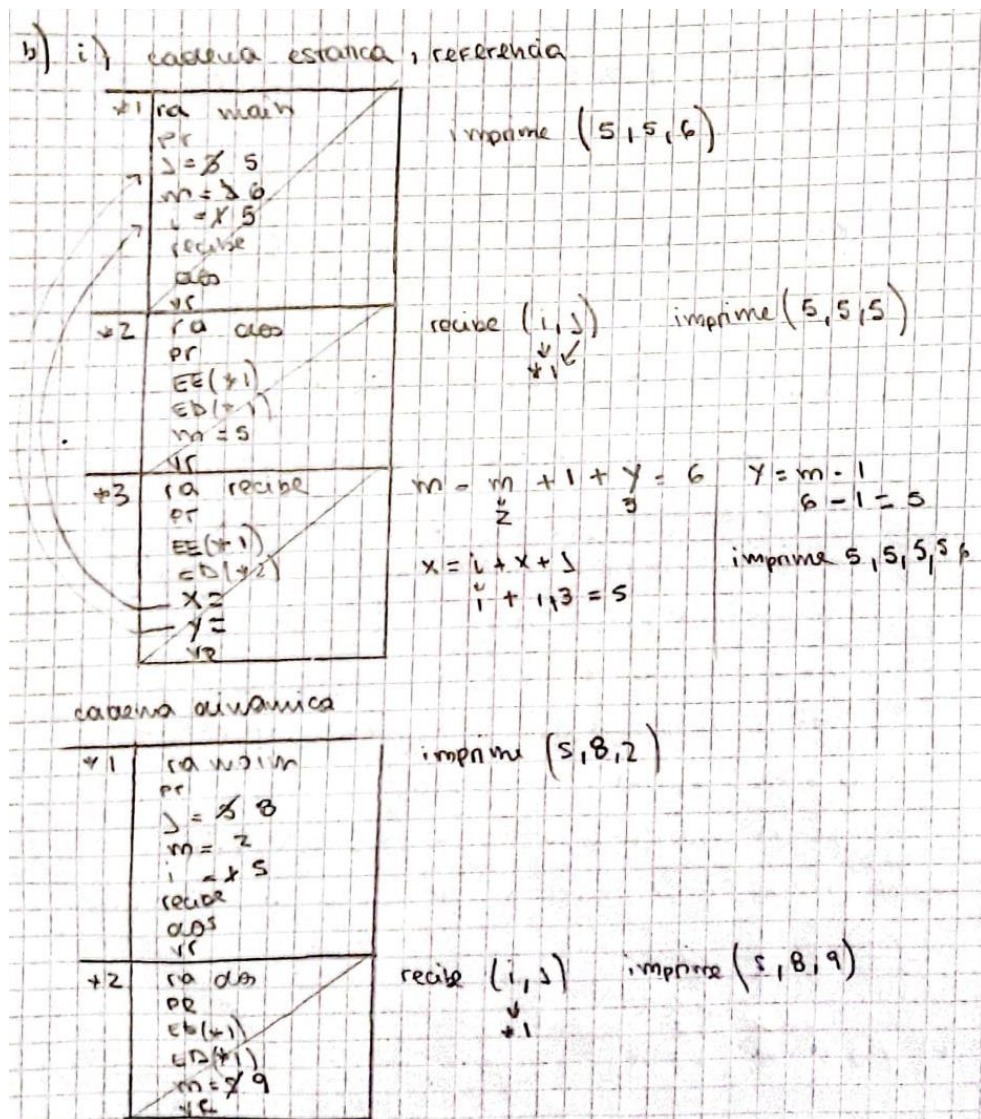
<pre> Procedure Main; var j, m, i: integer; Procedure Recibe (x:integer; y:integer); begin m:= m + 1 + y; x:=i + x + j; y:=m - 1; write (x, y, i, j, m); end; </pre>	<pre> Procedure Dos; var m:integer; begin m:= 5; Recibe(i, j); write (i, j, m); end; begin m:= 2; i:=1; j:=3; Dos; write (i, j, m); end. </pre>
---	--

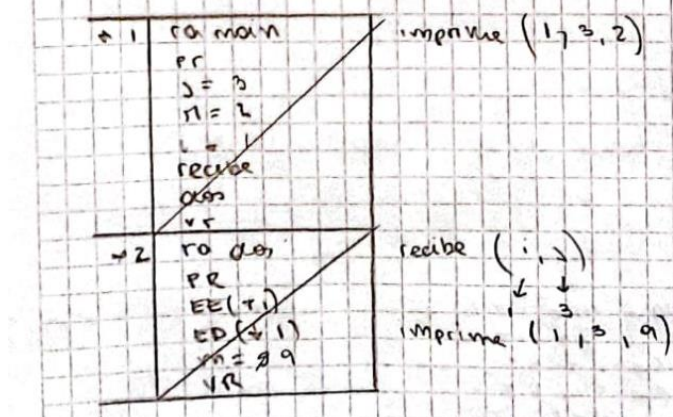
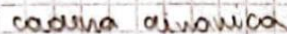
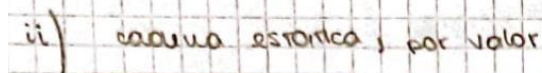
- a- Arme el árbol de anidamiento sintáctico y el registro de activación de cada una de las unidades.



- b- Decir qué imprime el programa suponiendo que para todas las variables que se pasan el pasaje de parámetros es por: (Deberá hacer la pila estática y dinámica para cada caso)

i- Referencia. ii- Valor. iii- Valor Resultado. iv- Nombre. v- Resultado.





v3 ra recube
 PR
 EE(+1)
 ED(+2)
~~X = X 5~~
~~Y = Y 8~~
 VR

$m = 5 + 1 + 3 = 9$
 $x = i + X + 3 = 5$
 $1 + 1 + 3$
 $y = m - 1 = 8$
 $9 - 1$
 imprime (5, 8, 1, 3, 9)

iii) cadeia estatica, valor resultados

v1 ra main
 PR
~~J = J 5~~
~~M = M 6~~
~~I = I 5~~
 recube
 dos
 VR

imprime (5, 5, 6)

v2 ra dos
 PR
 EE(+1)
 ED(+1)
~~m = 5~~
~~VR 5, 5~~

recube (i, j) imprime (5, 5, 5)
 $\begin{matrix} \downarrow & \downarrow \\ 1 & 5 \end{matrix}$

v3 ra recube
 PR
 EE(+1)
 ED(+2)
~~X = (VR:1) = 5~~
~~Y = (VR:2) = 8~~
 VR

$m = m + 1 + Y = 6$
 $2 + 1 + 3$
 $x = i + X + 3 = 5$
 $1 + 1 + 3$
 $y = m - 1 = 5$
 $6 - 1$
 imprime (5, 5, 1, 3, 6)

cadeia dinamica

v1 ra main
 PR
~~J = J 8~~
~~M = 2~~
~~I = I 5~~
 recube
 dos
 VR

imprime (5, 8, 2)

v2 ra dos
 PR
 EE(+1)
 ED(+1)
~~m = 8, 9~~
~~VR 5, 8~~

recube (i, j) imprime (5, 8, 9)
 $\begin{matrix} \downarrow & \downarrow \\ 1 & 3 \end{matrix}$

v3 ra recube
 PR
 EE(+1)
 ED(+2)
~~X = (VR:1) = 5~~
~~Y = (VR:2) = 8~~
 VR

$m = m + 1 + Y = 9$
 $5 + 1 + 3$
 $x = i + X + 3 = 5$
 $1 + 1 + 3$
 $y = m - 1 = 8$
 $9 - 1$
 imprime (5, 8, 1, 3, 9)

cadena estática, nombre

HUJIN

FECHA

v1	<div>ra main</div> <div>pr</div> <div>$j = 8$</div> <div>$m = 2$</div> <div>$i = 5$</div> <div>recibe</div> <div>des</div> <div>vr</div>
v2	<div>ra des</div> <div>pr</div> <div>EE(+1)</div> <div>ED(+1)</div> <div>$m = 5$</div> <div>vr</div>
v3	<div>ra recibe</div> <div>pr</div> <div>EE(+1)</div> <div>ED(+2)</div> <div>$X = (\uparrow i)$</div> <div>$Y = (\uparrow j)$</div> <div>vr</div>

imprime (s, s, 6)

imprime (s, s, s)

$$m = m + 1 + y = 6$$

$$2 + 1 + 3 = 6$$

$$\begin{cases} X = i + X_{des} + 2 = 5 \\ i = 1 + 1 + 3 = 5 \\ y = m - 1 = 5 \\ j = 6 - 1 = 5 \end{cases}$$

imprime (s, s, s, s)

cadena dinámica

v1	<div>ra main</div> <div>pr</div> <div>$j = 8$</div> <div>$m = 2$</div> <div>$i = 5$</div> <div>recibe</div> <div>des</div> <div>vr</div>
v2	<div>ra des</div> <div>pr</div> <div>EE(+1)</div> <div>ED(+1)</div> <div>$m = 9$</div> <div>vr</div>
v3	<div>ra recibe</div> <div>pr</div> <div>EE(+1)</div> <div>ED(+2)</div> <div>$X = (\uparrow i)$</div> <div>$Y = (\uparrow j)$</div> <div>vr</div>

imprime (s, 8, 2)

imprime (s, 8, 9)

$$m = m + 1 + y = 9$$

$$5 + 1 + 3 = 9$$

$$X = i + X_{des} + 1 = 5$$

$$i = 1 + 1 + 3 = 5$$

$$y = m - 1 = 8$$

$$j = 9 - 1 = 8$$

imprime (s, 8, s, 8, 9)

c- ¿Existió algún caso que no pudo realizarlo porque saltó algún tipo de error? Diga cuál y por qué.

Si, en la cadena estática y dinámica resultado hay error en $m = m + 1 + y$ ya que y no esta inicializada

d- ¿Dará el mismo resultado si se trata de un lenguaje que sigue la cadena dinámica? Justifique la respuesta realizando las pilas de activación

No, no será el mismo resultado, se puede observar en el punto b como no tienen los mismos resultados e impresiones.

Ejercicio 5: Suponiendo que se está ejecutando un programa con el siguiente registro de activación en memoria y se llama al procedimiento rutina(iter,vec,a). Determine el tipo de parámetro que se deben utilizar en el llamado para que los resultados sean los siguientes:

a) (4,6,7),(4,6,7), 2, 2

b) (3,5,6),(4,6,7), 2, 2

c) (3,5,6),(5,5,6), 0, -1

PR
LD
LE
Iter: true
Vec:[3,5,6]
a: -1
Rutina()
VR

.....

procedura rutina(tipoParam iteracion,tipoParam vector,tipoParam vit):

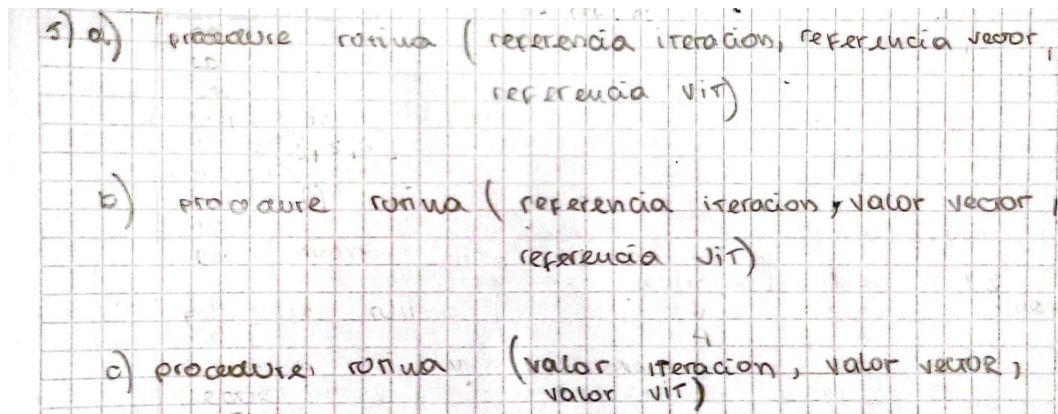
```

while iteracion begin
    vit = a+1
    vector[vit] = vector[vit]+1
    iteracion = (vector[vit] mod 2)==0
end
print vec
print vector
print vit
print a

```

.....

rutina(iter,vec,a)



Ejercicio 6: Indique con un ejemplo el comportamiento del parámetro por nombre (en el parámetro formal) para los siguientes casos de parámetros reales:

- **Un valor entero:** Un único valor se comporta exactamente igual que el pasaje por referencia
- **Una constante:** Si es una constante es equivalente a por valor
- **Un elemento de un arreglo:** Si es un elemento de un arreglo puede cambiar el subíndice entre las distintas referencias
- **Una expresión:** Si es una expresión se evalúa cada vez

Ejercicio 7: Realice la pila de ejecución del siguiente programa: a) siguiendo la cadena estática b) siguiendo la cadena dinámica

<pre> Procedure Uno; y, z: integer; r1:array[1..6] of integer; r2:array[1..5] of integer; Procedure Dos(nombre x, t:integer; var io:integer; valor-resultado y:integer); Procedure Dos(nombre t1:integer); Procedure Tres; begin y:= y + 1; z:= z + 1; end; begin t1:= t1 + 1; t:= t + 1; Tres; t1:= t1 + 2; t:= t + 2; end; </pre>	<pre> begin x:= x + 1; t:= t + 1; io:= io + 1; x:= x + 2; if z =2 then Dos (t); end; begin for y:= 1 to 6 do r1(y):= 2; for y:= 1 to 5 do r2(y):= 1; z:= 2; y:= 1; Dos(r1(y + r2(y)), r2(z), y, z); for y:= 1 to 6 do write (r1(y)); for y:= 1 to 5 do write (r2(y)); end. </pre>
---	--

4) io es modo in valor

uno
/
dos
/
dos
/
tres

cadena estatica y cadena dinamica iguales

r1 ra uno
pr
 $y = t + 1 = 1 + 1 = 2$
 $z = x + 3 = 2 + 3 = 5$
 $r1(1) = 2$
 $r1(2) = 2 + 3 = 5$
 $r1(3) = 2$
 $r1(4) = 2$
 $r1(5) = 2$
 $r1(6) = 2$
 $r2(1) = 1$
 $r2(2) = 2 + 3 = 5$
 $r2(3) = 1 + 2 = 3$
 $r2(4) = 1$
 $r2(5) = 1$
dos
vr 3

dos
imprime 7
2
2
2
2
1
4
4
1
1

r2 ra dos
pr
 $EE(+1)$
 $ED(+1)$
 $X = [r1(y + r2(y))]$
 $t = [r2(z)]$
 $io = t + 2$
 $y = (x + z) = 2 + 3 = 5$
dos
vr

r3 ra dos
pr
 $EE(+2)$
 $ED(+2)$
 $t1 = t + 1$
tres
vr

r4 ra tres
pr
 $EE(+3)$
 $ED(+3)$
vr

es el y del wpar del cual se invoca

$r1(y + r2(y)) = r1(2) + 1 = 2 + 1 = 3$
 $t = t + 1$
 $r2(z) = r2(5) + 1 = 5 + 1 = 6$
 $io = io + 1 = 2 + 1 = 3$
 $X = r1(y + r2(y)) = r1(3) + 2 = 3 + 2 = 5$
 $t1 = t1 + 1$
 $t = t + 1$
 $r2(z) = r2(z) + 1 = 6 + 1 = 7$
 $t = t + 1$
 $r2(z) = r2(z) + 1 = 7 + 1 = 8$
 $y = y + 1 = 5 + 1 = 6$
 $z = z + 1 = 6 + 1 = 7$

Siempre en por nombre el ambiente de la variable que sale en el nombre es así que realiza la invocación