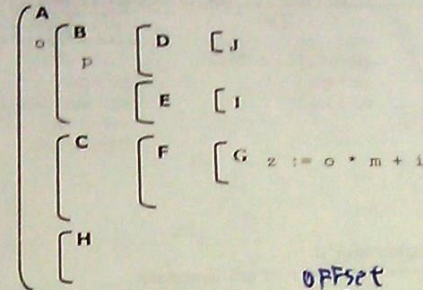


OFFSET O = 16  
OFFSET M = 4  
OFFSET ~~X~~ = 10  
OFFSET Z = 16

### Ejercicio Nro. 1

¿En qué lugar se encuentran definidas las variables presentes en el siguiente fragmento de código, teniendo en cuenta la definición de la cadena estática?



Offset, nivel de anidamiento desde donde está la expresión

```
MOV R2, BP
MOV BP, BP+4
MOV BP, BP+4
MOV R1, BP + 16
MOV BP, R2
MOV BP, BP+4
MOV BP, BP+4
MOV BP, BP+4
ADD R1, BP + 10
MOV BP, R2
MOV BP, BP+4
MOV BP + 16, R1
MOV BP, R2
```

$\rightarrow (\#16, 2) = O \rightarrow O$  está en C ✓

$\rightarrow (\#4, 0) = M \rightarrow M$  está en G ✓

$\rightarrow (\#10, 3) = i \rightarrow i$  está en A ✓

$\rightarrow (\#16, 1) = Z \rightarrow Z$  está en F ✓

### Ejercicio Nro. 2

z: integer;

a: dimension [1..4] of integer;

```
procedure test (x,w,y)
begin
```

```
  j: integer;
  z := z + 1;
  x := 5;
  a[3] := a[1] + 1;
  w := x - 2;
  y := x + w;
  j := y - x;
  a[j] := a[j] - 1;
```

```
end;
```

```
begin
```

```
  a[] := [2,3,4,5];
```

```
  z := 1;
```

```
  test (a[z], a[z], a[4]);
```

```
  print (a[1], a[2], a[3], a[4]);
```

```
end.
```

Determinar qué resultados se imprimirán si los parámetros son pasados  
a) por referencia b) por copia valor

Justificar la respuesta

### Ejercicio Nro. 3

Suponga un lenguaje tipo Algol (LTA) que utiliza el algoritmo de "Marcado y escaneado" como Garbage Collector (GC). A cada porción de memoria solicitada por el programador, el lenguaje le adiciona 2 enteros, utilizados para almacenar tamaño y marca, siendo utilizados por el GC.

Se tienen las siguientes rutinas:

```
Proc A
(
  int *pa
  int *qa
  int val
  pa=malloc(100)
  qa=malloc(50)
  call(b)
)

Proc B
(
  int *pb
  int *qb
  int vbl
  pb=malloc(100)
  qb=malloc(50)
  call(c)
)

Proc C
(
  int *pc
  int *qc
  int vcl
  pc=malloc(100)
  qc=qa
  call(d)
)

Proc D
(
  int *pd
  int *qd
  int *rd
  int *vdl
  pb=pc
  qd=malloc(100)
  rd=aval
  free(qd)
)
```

Nota: Los enteros ocupan 2 bytes. El comienzo de la memoria es en la dirección: 0

- ¿Cómo queda el heap luego que la rutina "D" devuelve el control? Representar gráficamente.
- En el caso que exista garbage en al retornar, utilice el GC indicando las direcciones de comienzo de cada variable en el heap. Representar gráficamente.
- Mencione cuando y donde aparecen referencias desactivadas si corresponde.

### Ejercicio Nro. 4

Sean las siguientes definiciones en ADA :

```
type SUP is new INTEGER
type CM is new INTEGER
type UNID is new INTEGER
A,B : CM
S : SUP
Z : UNID
X : INTEGER
```

- Sea A la base de un triángulo y B su altura. Calcular X como su superficie.
- Sea Z la cantidad de unidades cuya superficie individual es S. Calcular X como la superficie total.

### Ejercicio Nro. 5

Algunos compiladores utilizan como notación intermedia aquella llamada "polaca inversa" o notación postfija. Esta notación fue inventada por el matemático polaco Jan Lukasiewicz. Para las expresiones aritméticas, esta notación no requiere paréntesis.

Por ejemplo

$12 + 8 * 25$

$(12 + 8) * 25$

Se escriben en polaca inversa como

$12\ 8\ 25\ * +$

$12\ 8 +\ 25\ *$

Defina una gramática en notación BNF para describir expresiones de suma y multiplicación en notación polaca inversa.

Considere las constantes como terminales (cte)



2) a) Referencia  $(a[1], a[2], a[3], a[4])$   
 $x, y, z$   
 ①  $z := z + 1;$   
 ②  $x := 5;$   
 ③  $a[1] := a[1] + 1;$   
 ④  $w := x - z;$   
 ⑤  $y := x + w;$   
 ⑥  $j := y - x;$   
 ⑦  $a[j] := a[j] - 1;$

$z$	$a[1]$	$a[2]$	$a[3]$	$a[4]$
2	2	3	1	2
3	3	3	2	2
4	4	3	3	2

Se imprime  $a[1] = 3, a[2] = 3, a[3] = 3, a[4] = 2$   
 en pasaje por referencia, los parámetros formales  
 no existen en la función llamada, son alias de los parámetros  
 reales.

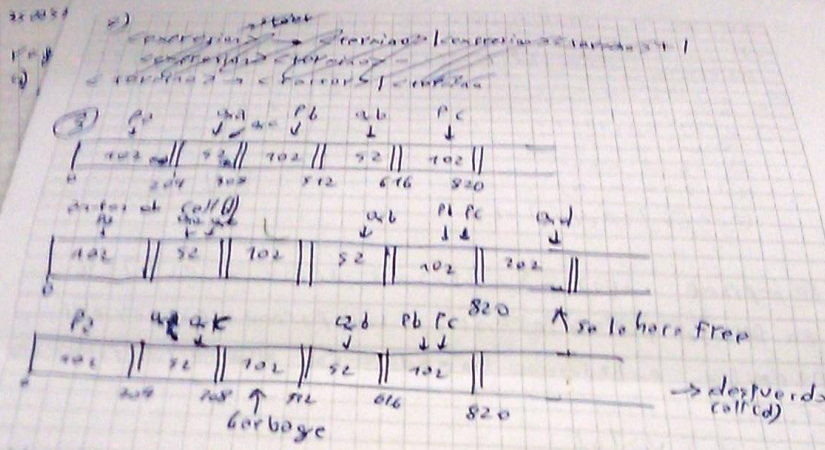
b) por copia valor

se agrega al código de test

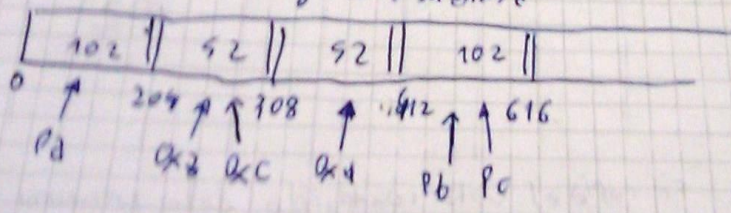
$x$	$w$	$x$	$y$	$z$	$a[1]$	$a[2]$	$a[3]$	$a[4]$
2	2	2	5	1	2	3	4	5
3	3	3	8	2			3	3
							2	

En copia valor se copia el valor contenido de los parámetros  
 reales a los parámetros formales, pero el contenido de estos  
 últimos se pierde al terminar la función test

Se imprime:  $a[1] = 2, a[2] = 3, a[3] = 2, a[4] = 5$  ✓



b) al aplicar Garbage collector en un lenguaje tipo Algol devuelvo las posiciones de memoria que quedara inaccesibles a la lista de libros y aplico compactación. El heap queda de la siguiente manera



c) aparecería una referencia desactivada tras hacer el free(Qd) por que Qd quedaría apuntando a una posición de memoria que no se está utilizando, sin embargo Qd desaparecerá inmediatamente después una vez que termina D y se borra su



Registro de activación.

4) Ada posee verificadores de tipo por nombre, por lo tanto a pesar de tratarse todos de Integer, hay que realizar conversiones explicitas, en Ada todas las conversiones <sup>de tipo</sup> son explicitas

$$X := \text{INTEGER}(A * B) / 2$$

$$\sup(A) \left( \frac{b \times \ln}{2} \right)$$

b)  $X := \text{INTEGER}(Z) * \text{INTEGER}(S);$

5)

$\langle \text{expresion} \rangle \xrightarrow{\text{start}} \langle \text{termino} \rangle * \langle \text{termino} \rangle + \dots$

$\langle \text{termino} \rangle \rightarrow \langle \text{cte} \rangle \langle \text{cte} \rangle / \langle \text{cte} \rangle \langle \text{expresion} \rangle$

$\langle \text{expresion} \rangle \langle \text{cte} \rangle / \langle \text{expresion} \rangle \langle \text{expresion} \rangle$

