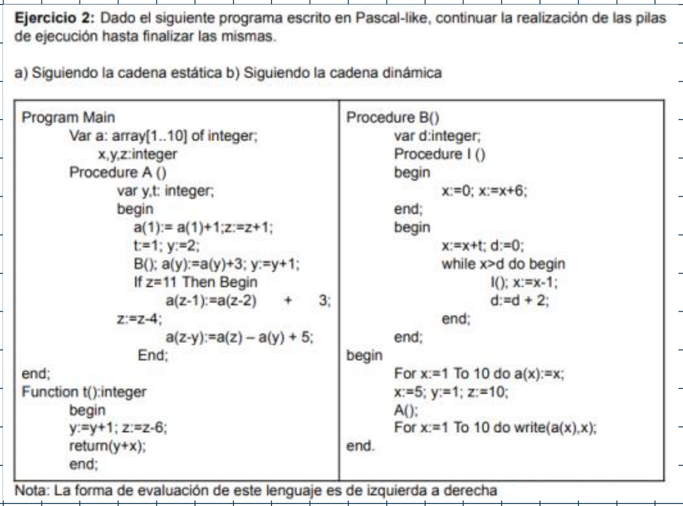
# Práctica 5

martes, 16 de abril de 2024 22:11



**Ejercicio 2:** Dado el siguiente programa escrito en Pascal-like, continuar la realización de las pilas de ejecución hasta finalizar las mismas.

a) Siguiendo la cadena estática b) Siguiendo la cadena dinámica

```
Program Main
        Var a: array[1..10] of integer;
             x,y,z:integer
        Procedure A ()
                var y,t: integer;
                begin
                   a(1):= a(1)+1;z:=z+1;
                   t:=1; y:=2;
                   B(); a(y):=a(y)+3; y:=y+1;
                   If z=11 Then Begin
                         a(z-1):=a(z-2)     +     3;
                z:=z-4;
                         a(z-y):=a(z) – a(y) + 5;
                    End;
end;
Function t():integer
        begin
        y:=y+1; z:=z-6;
        return(y+x);
        end;
```

```
Procedure B()
        var d:integer;
        Procedure I ()
        begin
                x:=0; x:=x+6;
        end;
        begin
                x:=x+t; d:=0;
                while x>d do begin
                        I(); x:=x-1;
                        d:=d + 2;
                end;
        end;
begin
        For x:=1 To 10 do a(x):=x;
        x:=5; y:=1; z:=10;
        A();
        For x:=1 To 10 do write(a(x),x);
end.
```

Nota: La forma de evaluación de este lenguaje es de izquierda a derecha

| |
|---|
| *1 Reg. Activación Main |
| Punto de retorno |
| Ld:- |
| Le:- |
| a(1):~~1~~,2 |
| a(2):~~2~~,5 |
| a(3):3 |
| a(4):4,9 |
| a(5):5 |
| a(6):6 |
| a(7):7 |
| a(8):8 |
| a(9):9 |
| a(10):10,12 |
| x:1..10,~~5,0,6,5,0,6,5,0,6~~,5 |
| y:1 |
| z:~~10~~,11,7 |
| |
| |
| |

Print:
2,1
5,2
3,4
4,9
5,5
6,6
7,7
8,8
9,9
12,10

| |
|---|
| *2 Reg activación A |
| Punto de retorno |
| Ld:*1 |
| Le:*1 |
| y:2,3 |
| t:1 |
| |
| |
| retorno |

a(1)=a(1)+1 z=z+1 a(y)=a(y)+3
a(1)=1+1 z=10+1 a(2)=2+3
a(1)=2 z=11 a(2)=5

Despues de terminar b la cague y fui a a en lugar de a main

| |
|---|
| *3 reg act B |
| Punto de retorno |
| Ld:*2 |
| Le:*1 |
| d:0,2,4,6 |
| |
| |
| retorno |

x=x+t
x=5+(T NO ESTÁ PORQUE POR ESTÁTICO
VA A MAIN?)

x=x-1 d=d+2 x=x-1 d=d+2 x=x-1 d=d+2
x=6-1 d=0+2 x=6-1 d=2+2 x=6-1 d=4+2

| |
|---|
| *4 REG act i |
| Punto de retorno |
| Ld:*3 |
| Le:*3 |
| |
| retorno |

x=0
x=x+6
x=0+6

| |
|---|
| *5 REG act i |
| Punto de retorno |
| Ld:*3 |
| Le:*3 |
| |
| retorno |

x=0
x=x+6
x=0+6

| |
|---|
| *6 REG act i |
| Punto de retorno |
| Ld:*3 |
| Le:*3 |
| |
| retorno |

x=0
x=x+6
x=0+6

Estática de nuevo

| |
|---|
| *1 Reg. Activación Main |
| Punto de retorno |
| Ld:- |
| Le:- |
| a(1):1,2 |
| a(2):2,5 |
| a(3):3 |
| a(4):4 |
| a(5):5 |
| a(6):6 |
| a(7):7 |
| a(8):8 |
| a(9):9 |
| a(10):10 |
| x:1..10,5,6,14,0,6,5,0,6,5,0,6,5, 1..10 |
| y:1,3 |
| z:10,11,5 |
| Procedure a |
| Function t |
| Procedure b |

Print:
2,1
5,2
3,3
4,4
5,5
6,6
7,7
8,8
9,9
10,10

| |
|---|
| *2 Reg activación A |
| Punto de retorno |
| Ld:*1 |
| Le:*1 |
| y:2,3 |
| t:1 |
| |
| |
| |
| retorno |

```
a(1)=a(1)+1         z=z+1
a(1)=1+1=2          z=10+1
                                        a(y)=a(y)+3           y=y+1
                                        a(2)=a(2)+3           y=2+1=3
                                        a(2)=2+3
                                        a(2)=5
```

| |
|---|
| *3 Reg activación B |
| Punto de retorno |
| Ld:*2 |
| Le:*1 |
| d:0,2,4,6 |
| Procedure i |
| |
| Retorno: 9 |

```
x=x+t     x=x-1    d=d+2    x=x-1    d=d+2     x=x-1    d=d+2
x=5+9     x=6-1    d=0+2    x=6-1    d=2+2     x=6-1    d=4+2
x=14      x=5      d=2      x=5      d=4       x=5      d=6
```

| |
|---|
| *4 Reg activación t |
| Punto de retorno |
| Ld:*3 |
| Le:*1 |
| |
| |
| |
| retorno |

```
y=y+1       z=z-6
y=2+1=3     z=11-6=5
```

Return x+y=6+3

| |
|---|
| *5 Reg activación i |
| Punto de retorno |
| Ld:*3 |
| Le:*3 |
| |

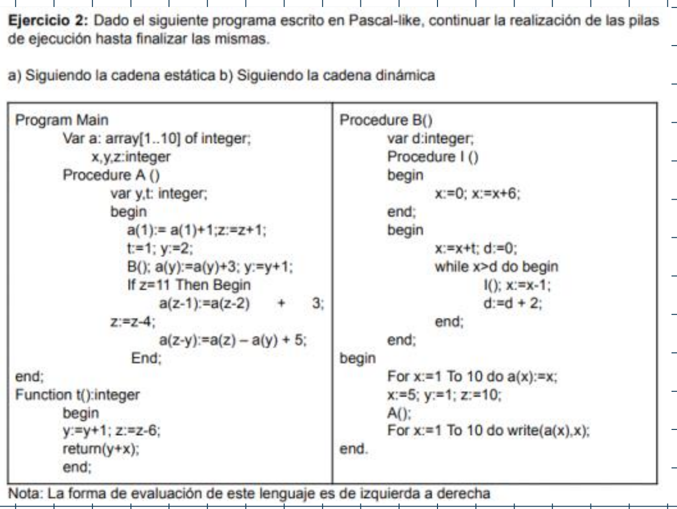
x=0
x=x+6
X=0+6=6

| |
|---|
| *6 Reg activación i |
| Punto de retorno |
| Ld:*3 |
| Le:*3 |
| |

x=0
x=x+6
X=0+6=6

| |
|---|
| *7 Reg activación i |
| Punto de retorno |
| Ld:*3 |
| Le:*3 |
| |

x=0
x=x+6
X=0+6=6



**Ejercicio 2:** Dado el siguiente programa escrito en Pascal-like, continuar la realización de las pilas de ejecución hasta finalizar las mismas.

a) Siguiendo la cadena estática b) Siguiendo la cadena dinámica

```
Program Main
        Var a: array[1..10] of integer;
             x,y,z:integer
        Procedure A ()
                var y,t: integer;
                begin
                   a(1):= a(1)+1;z:=z+1;
                   t:=1; y:=2;
                   B(); a(y):=a(y)+3; y:=y+1;
                   If z=11 Then Begin
                         a(z-1):=a(z-2)     +     3;
                z:=z-4;
                         a(z-y):=a(z) – a(y) + 5;
                    End;
end;
Function t():integer
        begin
        y:=y+1; z:=z-6;
        return(y+x);
        end;
```

```
Procedure B()
        var d:integer;
        Procedure I ()
        begin
                x:=0; x:=x+6;
        end;
        begin
                x:=x+t; d:=0;
                while x>d do begin
                          I(); x:=x-1;
                          d:=d + 2;
                end;
        end;
begin
        For x:=1 To 10 do a(x):=x;
        x:=5; y:=1; z:=10;
        A();
        For x:=1 To 10 do write(a(x),x);
end.
```

Nota: La forma de evaluación de este lenguaje es de izquierda a derecha

Dinámica

| |
|---|
| *1 Reg. Activación Main |
| Punto de retorno |
| Ld:- |
| Le:- |
| a(1):1,2 |
| a(2):2,5 |
| a(3):3 |
| a(4):4,9 |
| a(5):5 |
| a(6):6 |
| a(7):7 |
| a(8):8 |
| a(9):9 |
| a(10):10,12 |
| x:1..10,<br>5,6,0,6,5,0,6,5,0,6,5,<br>1..10 |
| y:1 |
| z:10,11,7 |
| Procedure a |
| Function t |
| Procedure b |

Print
2,1
5,2
3,3
9,4
5,5
6,6
7,7
8,8
9,9
12,10

| |
|---|
| *2 Reg activación A |
| Punto de retorno |
| Ld:*1 |
| Le:*1 |
| y:2,3 |
| t:1 |
| |
| |
| |
| retorno |

a(1)=a(1)+1
a(1)=1+1=2

z=z+1
z=10+1=11

a(y)=a(y)+3
a(2)=a(2)+3
a(2)=2+3=5

y=y+1
y=2+1=3

a(z-1)=a(z-2)+3
a(10)=a(9)+3
a(10)=9+3=12

z=z-4
z=11-4=7

a(z-y)=a(z)-a(y)+5
a(7-3)=a(7)-a(3)+5
a(4)=7-3+5
a(4)=9

| |
|---|
| *3 Reg activación B |
| Punto de retorno |
| Ld:*2 |
| Le:*1 |
| d:2,4,6 |
| Procedure i |
| |
| |
| |
| retorno |

x=x+t
x=5+1=6

x=x+-1
x=6-1=5

d=d+2
d=0+2=2

x=x+-1
x=6-1=5

d=d+2
d=2+2=4

x=x+-1
x=6-1=5

d=d+2
d=4+2=6

| |
|---|
| ~~*4 Reg activación i~~ |
| ~~Punto de retorno~~ |
| ~~Ld:*3~~ |
| ~~Le:*3~~ |
| |
| |
| ~~retorno~~ |

x=0

x=x+6
x=0+6

| |
|---|
| *5 Reg activación i |
| Punto de retorno |
| Ld:*3 |
| Le:*3 |
| |
| |
| retorno |

x=0

x=x+6
x=0+6

| |
|---|
| *6 Reg activación i |
| Punto de retorno |
| Ld:*3 |
| Le:*3 |
| |
| |
| retorno |

x=0

x=x+6
x=0+6

P1

Pp1

Procedure x

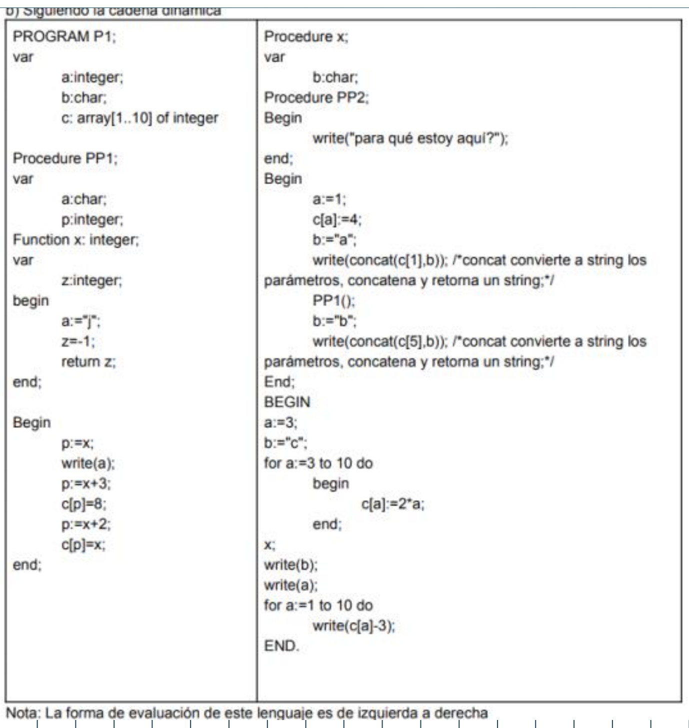
Function x

Procedure pp2

| *1 Reg. Activación Main |
|---|
| Punto de retorno |
| Ld:- |
| Le:- |
| c(1):4,-1 |
| c(2):8 |
| c(3):6 |
| c(4):8 |
| c(5):10 |
| c(6):12 |
| c(7):14 |
| c(8):16 |
| c(9):18 |
| c(10):20 |
| a:3..10, 1, 1..10 |
| b:"c" |
| Procedure pp1 |
| Procedure x |

c[a]=2*a

```
Write(b): c
Write(a): 1
Write(c[a]-3):
          -4
          5
          3
          5
          7
          9
          11
          13
          15
          17
```



b) Siguiendo la cadena dinámica

```
PROGRAM P1;
var
        a:integer;
        b:char;
        c: array[1..10] of integer

Procedure PP1;
var
        a:char;
        p:integer;
Function x: integer;
var
        z:integer;
begin
        a:="j";
        z=-1;
        return z;
end;

Begin
        p:=x;
        write(a);
        p:=x+3;
        c[p]=8;
        p:=x+2;
        c[p]=x;
end;
```

```
Procedure x;
var
        b:char;
Procedure PP2;
Begin
        write("para qué estoy aquí?");
end;
Begin
        a:=1;
        c[a]:=4;
        b:="a";
        write(concat(c[1],b)); /*concat convierte a string los parámetros, concatena y retorna un string;*/
        PP1();
        b:="b";
        write(concat(c[5],b)); /*concat convierte a string los parámetros, concatena y retorna un string;*/
End;
BEGIN
a:=3;
b:="c";
for a:=3 to 10 do
        begin
                c[a]:=2*a;
        end;
x;
write(b);
write(a);
for a:=1 to 10 do
        write(c[a]-3);
END.
```

Nota: La forma de evaluación de este lenguaje es de izquierda a derecha

| *2 Reg activación procedure x |
|---|
| Punto de retorno |
| Ld:*1 |
| Le:*1 |
| b: "a", "b" |
| |
| |
| |
| |
| retorno |

```
a=1
c[a]=4
c[1]=4
b="a"
```

```
Write(c[5],b)
        10, a
```

b="b"

```
Write(c[5],b)
        10, b
```

| *3 Reg activación pp1 |
|---|
| Punto de retorno |
| Ld:*2 |
| Le:*1 |
| a:"j", "j", |
| p:2,1 |
| |
| |
| |
| Retorno -1, -1, -1,-1 |

```
p=x
p=-1
```

Write(a)=j

```
p=x+3
p=-1+3=2
```

```
c[p]=8
c[2]=8
```

```
p=x+2
p=-1+2=1
```

```
c[p]=x
c[1]=-1
```

| *4 Reg activación x |
|---|
| Punto de retorno |
| Ld:*3 |
| Le:*3 |
| z:-1 |
| |
| retorno |

```
a="j"
z=-1
```

| *5 Reg activación x |
|---|
| Punto de retorno |
| Ld:*3 |
| Le:*3 |
| z:-1 |
| |
| retorno |

```
a="j"
z=-1
```

| *6 Reg activación x |
|---|
| Punto de retorno |
| Ld:*3 |
| Le:*3 |
| z:-1 |
| |
| retorno |

```
a="j"
z=-1
```

| *7 Reg activación x |
|---|
| Punto de retorno |
| Ld:*3 |

| |
|---|
| Ld:*3 |
| Le:*3 |
| z:-1 |
| |
| retorno |

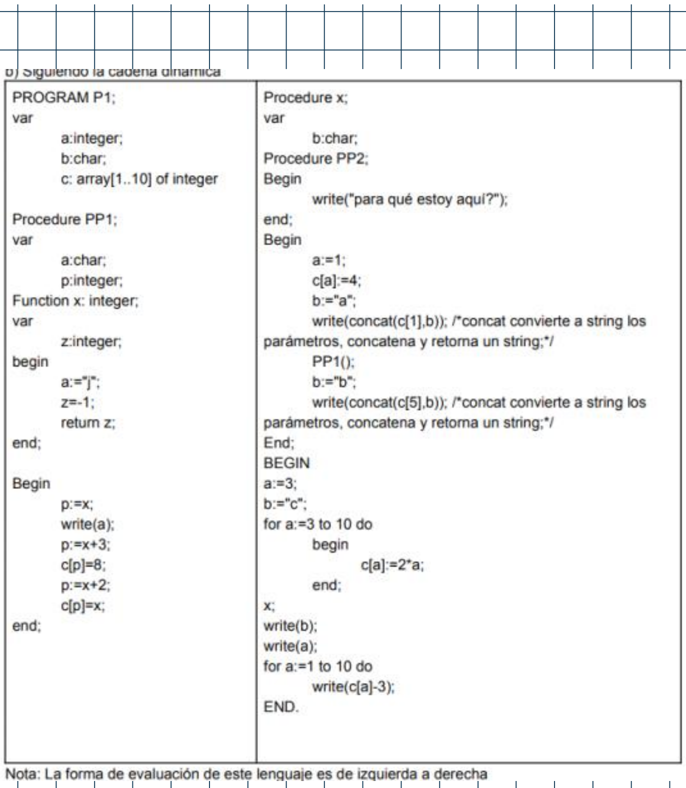
a="j"
z=-1

dinámica

| *1 Reg. Activación Main |
|---|
| Punto de retorno |
| Ld:- |
| Le:- |
| c(1):-1 |
| c(2):8 |
| c(3):6 |
| c(4):8 |
| c(5):10 |
| c(6):12 |
| c(7):14 |
| c(8):16 |
| c(9):18 |
| c(10):20 |
| a:3..10,1, 1..10 |
| b:"c" |
| Procedure pp1 |
| Procedure x |

c[a]=2*a

```
Write(b):"c"
Write(a): 1
Write(c[a]-3):
     -4
     5
     3
     5
     7
     9
     11
     13
     15
     17
```



b) Siguiendo la cadena dinámica

```
PROGRAM P1;
var
        a:integer;
        b:char;
        c: array[1..10] of integer

Procedure PP1;
var
        a:char;
        p:integer;
Function x: integer;
var
        z:integer;
begin
        a:="j";
        z=-1;
        return z;
end;

Begin
        p:=x;
        write(a);
        p:=x+3;
        c[p]=8;
        p:=x+2;
        c[p]=x;
end;
```

```
Procedure x;
var
        b:char;
Procedure PP2;
Begin
        write("para qué estoy aquí?");
end;
Begin
        a:=1;
        c[a]:=4;
        b:="a";
        write(concat(c[1],b)); /*concat convierte a string los
parámetros, concatena y retorna un string;*/
        PP1();
        b:="b";
        write(concat(c[5],b)); /*concat convierte a string los
parámetros, concatena y retorna un string;*/
End;
BEGIN
a:=3;
b:="c";
for a:=3 to 10 do
        begin
                c[a]:=2*a;
        end;
x;
write(b);
write(a);
for a:=1 to 10 do
        write(c[a]-3);
END.
```

Nota: La forma de evaluación de este lenguaje es de izquierda a derecha

| *2 Reg activación procedure x |
|---|
| Punto de retorno |
| Ld:*1 |
| Le:*1 |
| b:"a", "b" |
| |
| |
| |
| retorno |

a=1
c[a]=4
c[1]=4
b="a"

Write(c[1],b):
4,"a"

b:="b"

Write(c[5], b)
10, "b"

| *3 Reg activación pp1 |
|---|
| Punto de retorno |
| Ld:*2 |
| Le:*1 |
| a:"j", "j", "j" |
| p:-1,2,1 |
| Function x |
| |
| Retorno -1,-1,-1 |

p:=x
p:= -1

Write(a)
"j"

p:=x+3
p:=-1+3=2

c[p]=8
c[2]=8

p=x+2
p=-1+2=1

c[p]=x
c[1]=-1

| *4 Reg activación function x |
|---|
| Punto de retorno |
| Ld:*3 |
| Le:*3 |
| z:-1 |
| |
| retorno |

a:="j"
z=-1

| *5 Reg activación function x |
|---|
| Punto de retorno |
| Ld:*3 |
| Le:*3 |
| z:-1 |
| |
| retorno |

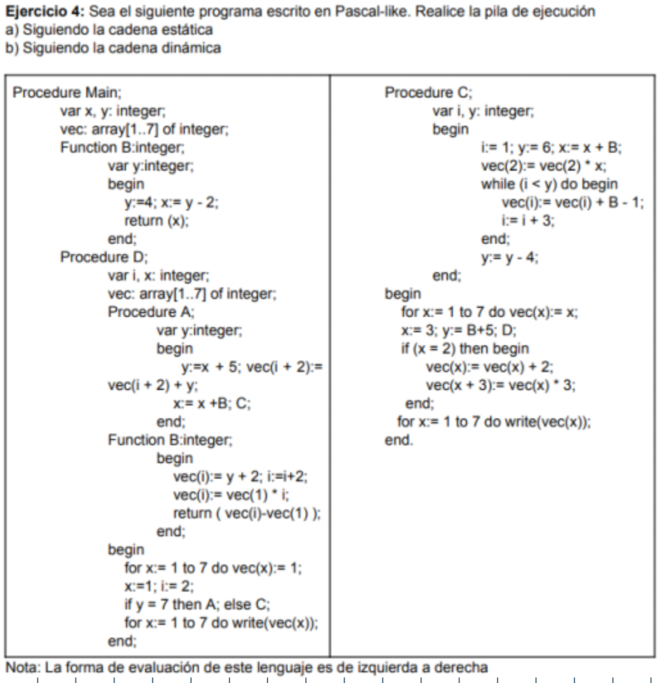
a:="j"
z=-1

| ~~*6 Reg activación function x~~ |
|---|
| ~~Punto de retorno~~ |
| ~~Ld:*3~~ |
| ~~Le:*3~~ |
| ~~z:-1~~ |
| |
| ~~retorno~~ |

a:="j"
z=-1

| ~~*7 Reg activación function x~~ |
|---|
| ~~Punto de retorno~~ |
| ~~Ld:*3~~ |
| ~~Le:*3~~ |
| ~~z:-1~~ |
| |
| ~~retorno~~ |

a:="j"
z=-1



**Ejercicio 4:** Sea el siguiente programa escrito en Pascal-like. Realice la pila de ejecución
a) Siguiendo la cadena estática
b) Siguiendo la cadena dinámica

```
Procedure Main;
      var x, y: integer;
      vec: array[1..7] of integer;
      Function B:integer;
            var y:integer;
            begin
              y:=4; x:= y - 2;
              return (x);
            end;
      Procedure D;
            var i, x: integer;
            vec: array[1..7] of integer;
            Procedure A;
                  var y:integer;
                  begin
                     y:=x + 5; vec(i + 2):=
            vec(i + 2) + y;
                    x:= x +B; C;
                  end;
            Function B:integer;
                  begin
                    vec(i):= y + 2; i:=i+2;
                    vec(i):= vec(1) * i;
                    return ( vec(i)-vec(1) );
                  end;
            begin
              for x:= 1 to 7 do vec(x):= 1;
              x:=1; i:= 2;
              if y = 7 then A; else C;
              for x:= 1 to 7 do write(vec(x));
            end;

Procedure C;
      var i, y: integer;
      begin
            i:= 1; y:= 6; x:= x + B;
            vec(2):= vec(2) * x;
            while (i < y) do begin
               vec(i):= vec(i) + B - 1;
               i:= i + 3;
            end;
            y:= y - 4;
      end;
begin
  for x:= 1 to 7 do vec(x):= x;
  x:= 3; y:= B+5; D;
  if (x = 2) then begin
     vec(x):= vec(x) + 2;
     vec(x + 3):= vec(x) * 3;
  end;
 for x:= 1 to 7 do write(vec(x));
end.
```

Nota: La forma de evaluación de este lenguaje es de izquierda a derecha

| *1 Reg. Activación Main |
|---|
| Punto de retorno |
| Ld:- |
| Le:- |
| v(1):1 |
| v(2):2 |
| v(3):3 |
| v(4):4 |
| v(5):5 |
| v(6):6 |
| v(7):7 |
| x:1..7,3,2,2 |
| y:7,4 |
| Function B |
| Procedure D |
| Procedure C |
| Retorno 2 |

x=3

y=b+5
y=2+5=7

| ~~*2 Reg activación function b~~ |
|---|
| ~~Punto de retorno~~ |
| ~~Ld:*1~~ |
| ~~Le:*1~~ |
| ~~y:4~~ |
| |
| ~~retorno~~ |

y=4
x=y-2
x=4-2=2

| *3 Reg activación Procedure d |
|---|

| |
|---|
| Punto de retorno |
| Ld:1 |
| Le:1 |
| v(1):1 |
| v(2):1,9 |
| v(3):1 |
| v(4):1,7,4 |
| v(5):1 |
| v(6):1 |
| v(7):1 |
| i:1..7,2,4 |
| x:1 |
| Function B |
| Procedure A |
| |
| Retorno |

| |
|---|
| *4 Reg activación function a |
| Punto de retorno |
| Ld:*3 |
| Le:*3 |
| y:6 |
| |
| Retorno 3 |

```
  y=x+5
  y=1+5=6

v(i+2)=v(i+2)+y
v(2+2)=v(2+2)+6
v(4)=v(4)+6=7

  x=x+b
  x=1+b
```

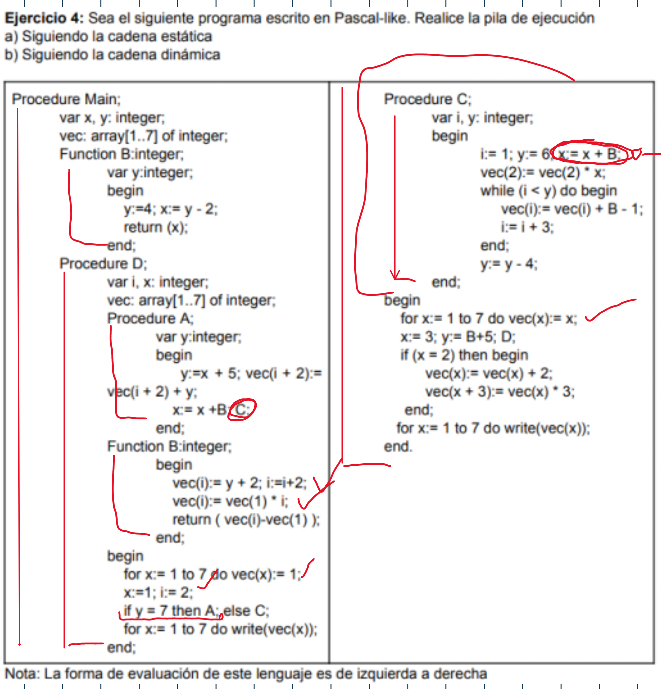
| |
|---|
| ~~*5 Reg activación function b~~ |
| ~~Punto de retorno~~ |
| ~~Ld:*4~~ |
| ~~Le:*3~~ |
| |
| |
| ~~retorno~~ |

```
v(i)=y+2
v(2)=7+2=9

i=i+2
i=2+2=4
```

```
v(i)=v(1)*i
v(4)=1*4=4

Return v(i)-v(1)= 4-1=3
```

| |
|---|
| *6 Reg activación procedure c |
| Punto de retorno |
| Ld:*4 |
| Le:*1 |
| y:6 |
| i:1 |
| Retorno |

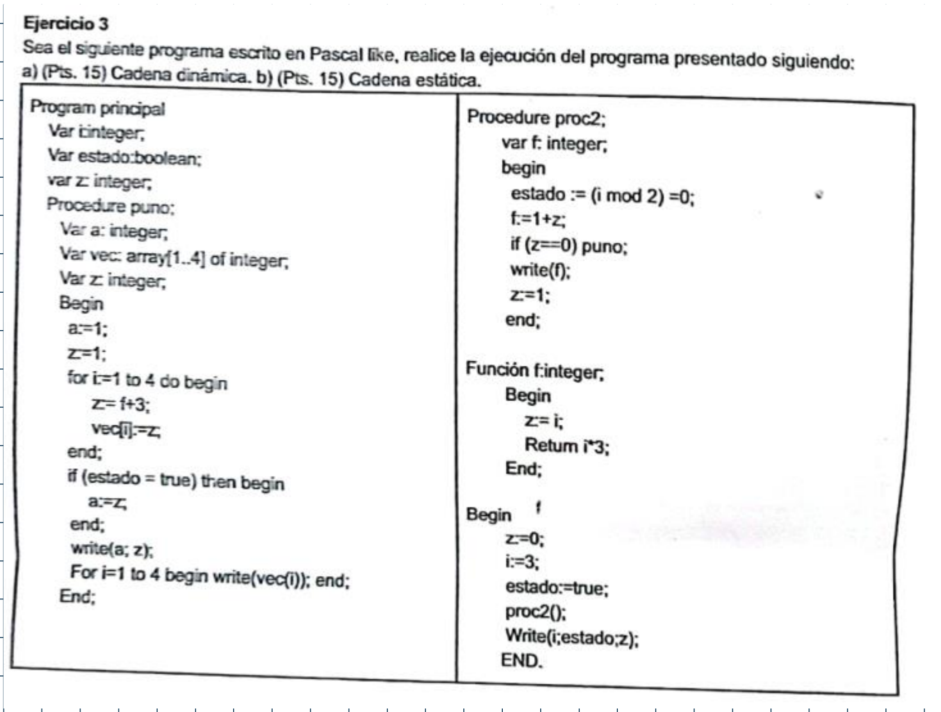
```
x=x+b
x=2+b
x=2+
```



**Ejercicio 4:** Sea el siguiente programa escrito en Pascal-like. Realice la pila de ejecución
a) Siguiendo la cadena estática
b) Siguiendo la cadena dinámica

```
Procedure Main;
        var x, y: integer;
        vec: array[1..7] of integer;
        Function B:integer;
                var y:integer;
                begin
                   y:=4; x:= y - 2;
                   return (x);
                end;
        Procedure D;
                var i, x: integer;
                vec: array[1..7] of integer;
                Procedure A;
                        var y:integer;
                        begin
                            y:=x + 5; vec(i + 2):=
                vec(i + 2) + y;
                           x:= x +B; C;
                        end;
                Function B:integer;
                        begin
                           vec(i):= y + 2; i:=i+2;
                           vec(i):= vec(1) * i;
                           return ( vec(i)-vec(1) );
                        end;
                begin
                   for x:= 1 to 7 do vec(x):= 1;
                   x:=1; i:= 2;
                   if y = 7 then A; else C;
                   for x:= 1 to 7 do write(vec(x));
                end;
        Procedure C;
                var i, y: integer;
                begin
                        i:= 1; y:= 6; x:= x + B;
                        vec(2):= vec(2) * x;
                        while (i < y) do begin
                           vec(i):= vec(i) + B - 1;
                           i:= i + 3;
                        end;
                        y:= y - 4;
                end;
begin
  for x:= 1 to 7 do vec(x):= x;
  x:= 3; y:= B+5; D;
  if (x = 2) then begin
     vec(x):= vec(x) + 2;
     vec(x + 3):= vec(x) * 3;
   end;
  for x:= 1 to 7 do write(vec(x));
end.
```

Nota: La forma de evaluación de este lenguaje es de izquierda a derecha



**Ejercicio 3**

Sea el siguiente programa escrito en Pascal like, realice la ejecución del programa presentado siguiendo: a) (Pts. 15) Cadena dinámica. b) (Pts. 15) Cadena estática.

```
Program principal
  Var i:integer;
  Var estado:boolean;
  var z: integer;
  Procedure puno;
    Var a: integer;
    Var vec: array[1..4] of integer;
    Var z: integer;
    Begin
     a:=1;
     z:=1;
     for i:=1 to 4 do begin
         z:= f+3;
         vec[i]:=z;
     end;
     if (estado = true) then begin
        a:=z;
     end;
     write(a; z);
     For i=1 to 4 begin write(vec(i)); end;
    End;
```

```
Procedure proc2;
      var f: integer;
      begin
       estado := (i mod 2) =0;
       f:=1+z;
       if (z==0) puno;
       write(f);
       z:=1;
      end;

Función f:integer;
      Begin
         z:= i;
         Return i*3;
      End;

Begin
      z:=0;
      i:=3;
      estado:=true;
      proc2();
      Write(i;estado;z);
     END.
```