

Trabajo Práctico Nº 6

Tema: CONTROL DE SECUENCIA – CONTROL DE DATOS

1. Dar un algoritmo para pasar de infijo a postfijo para expresiones simples (+,-,*,/). Proponer un mecanismo para manejar paréntesis.
2. Dar un algoritmo para evaluar expresiones aritméticas simples en notación:
a) Prefija b) Postfija
3. Escribir las reglas de prioridad de evaluación de expresiones aritméticas. Indicar operaciones que se evalúen de izquierda a derecha (asociatividad de izquierda a derecha) y operaciones que se evalúen de derecha a izquierda (asociatividad de derecha a izquierda).

4. Dada la función FUN:

```
Function FUN (var K:integer):integer;  
Begin  
  K:=K+4;  
  FUN:=3*K-1;  
end
```

Suponiendo que FUN es usada en el siguiente programa:

```
.....  
I:=10;  
SUM1:=(I/2)+FUN(I);  
J:=10;  
SUM2:= FUN(J)+(J/2);  
.....
```

Calcular los valores que asumen SUM1 y SUM2 sí:

- a) los operandos en las expresiones se evalúan de izquierda a derecha
 - b) los operandos en las expresiones se evalúan de derecha a izquierda
5. Dado el siguiente programa en C:

```
int fun(int *i) {  
  *i +=5;  
  return 4;  
}  
void main () {  
  int x = 3;  
  x = x + fun(&x);  
}
```

Calcular el valores que asume x sí:

- a) los operandos se evalúan de izquierda a derecha
 - b) los operandos se evalúan de derecha a izquierda
6. Dada la siguiente definición de función en C para fun:

```
int fun(int *k) {  
  *k +=4;  
  return 3 * (*k) - 1;  
}
```

Suponiendo que fun es usada en el siguiente programa:

```
void main () {  
  int x = 10, j=10, sum1, sum2;  
  sum1=( i / 2)+ fun(&i);  
  sum2= fun(&j) + (j/2);  
}
```

Determinar los valores de sum1 y sum2, ejecutando el programa en computadora. Explique los resultados.

Trabajo Práctico Nº 6

Tema: CONTROL DE SECUENCIA – CONTROL DE DATOS

7. Suponiendo que $F(X)$ evalúa a 3 e incrementa colateralmente a A en 1, entónces discutir los efectos de evaluar la expresión $A + F(X) + A$ considerando la asociatividad y/o la optimización. Discutir como afecta a la legibilidad del programa.

8. Describa el orden de evaluación de las siguientes asignaciones y expresiones aritméticas en C:

- a) $a++$; b) $b = ++a$; c) $a = a++$;
d) $b+ = a$; e) $b+ = a--$; f) $b- = --a$;

Esta forma de escribir operaciones, ¿afecta la legibilidad del programa? ¿cómo escribirían las mismas en Pascal?

9. En C las expresiones booleanas con “&&” y “ ” se evalúan de izquierda a derecha hasta encontrar el primer 0 (falso) o el 1 (verdadero), respectivamente. Discutir como afecta a la legibilidad del programa.

Para lenguajes de programación que realizan la evaluación de expresiones lógicas, aplicando la regla de evaluación ansiosa, como puede evitarse el error que se produciría en el siguiente condicional: `if (a=0) or (b/a > 10)` si a tiene el valor 0. Discutir como afecta a la legibilidad del programa.

10. En el lenguaje C puede escribirse la siguiente asignación:

$A = B = C ;$

Y en el lenguaje Pascal esto debe hacerse:

$B=C;$

$A=B;$

Explique porque C lo permite y Pascal no.

11. ¿Qué enunciados (control explícito) se utilizan en C para realizar saltos y bifurcaciones (tipo goto) Explicar y ejemplificar su funcionamiento.

12. Comparar la sentencia For en Pascal y en C.

13. Dado el siguiente programa en un lenguaje de programación parecido a Pascal:

```
I001  Programa A
I002  Procedimiento B
I003  ----
I004  Funcion C
I005  ComienzaC
      ----
I015  TerminaC
I016  ComienzaB
I017  ---
I025  C
I026  TerminaB
I027  Procedimiento D
I028  ComienzaD
      ----
I035  B
I036  ----
I037  TerminaD
I038  ComienzaA
      ---
I045  D
I046  ---
I057  B
I058  ---
I059  TerminaA
```

Trabajo Práctico Nº 6

Tema: CONTROL DE SECUENCIA – CONTROL DE DATOS

- a) ¿Cuántos registros de activación se crean y destruyen durante la ejecución del programa A? Distinguirlos e indicar el orden de activación.
- b) ¿Dónde retorna el control luego que se ejecuta el subprograma C?
- c) ¿Dónde retorna el control luego que se ejecuta el subprograma D?
- d) ¿Dónde retorna el control luego que se ejecuta el subprograma B?

14. Dado el siguiente programa en un lenguaje de programación parecido a Pascal:

```
Programa Factorial
Variables
  N: entero
Funcion Fact(N:entero):entero
  Comienza
    Si N = 0 Entonces Fact = 1
    Sino Fact = Fact(N-1)*N
  Termina
Comienza
  Leer N
  Escribir(Fact(N))
Termina.
```

Realizar los diagramas de las activaciones si se leen los valores 4 y 0 en N.

15. Dado el siguiente programa en un lenguaje de programación parecido a Pascal:

<pre>Programa Rec Variables N: entero Procedimiento Mensaje Comienza Escribir "Ejemplo de recursion" Termina Funcion Recursiva(N:entero):entero Comienza Si N > 0 Entonces Comienza Recursiva=recursiva(N-1) Mensaje Termina Comienza Escribir(Recursiva(4)) Termina.</pre>	<pre>Programa Rec Variables N: entero Procedimiento Mensaje Comienza Escribir "Ejemplo de recursion" Termina Funcion Recursiva(N:entero):entero Comienza Si N > 0 Entonces Comienza Mensaje Recursiva=recursiva(N-1) Termina Comienza Escribir(Recursiva(4)) Termina.</pre>
--	--

Dadas las dos versiones del programa Rec, analizar desde el punto de vista de eficiencia del almacenamiento y del tiempo cuales son las diferencias de los mismos.

16. En interrupciones ocurridas por situaciones excepcionales en la ejecución de una operación, no siempre es deseable que el control se reasuma inmediatamente después de ocurrida la excepción. Discutir puntos de retorno adecuado:

- a) ocurrió overflow durante el producto
- b) ocurrió EOF durante la lectura de un archivo
- c) ocurrió un subíndice fuera de rango en el acceso a una matriz

17. Dos corrutinas A y B pueden ser al mismo tiempo mutuamente recursivas. Discutir los mecanismos de control necesarios para obtener la implementación simultánea de RESUME y CALL-RETURN.

Trabajo Práctico Nº 6

Tema: CONTROL DE SECUENCIA – CONTROL DE DATOS

18. Dado un programa en Pascal, determine:

- a) Para cada subprograma su ambiente de referencia b) Para cada identificador su visibilidad

19. Considere el programa siguiente:

```
program Principal;  
var Y: integer;  
procedure P(X:integer);  
begin  
  X := X+1;  
  write(X,Y);  
end  
begin  
  Y:=1;  
  P(Y);  
  write(Y);  
end.
```

Proporcione los tres números que se imprimen en caso que Y se transmita a P:

- a) por valor; c) por valor-resultado
b) por referencia; d) por nombre

20. Considere el siguiente programa, parecido a Pascal:

```
program principal  
var  
  i,j,k,m: integer;  
procedure q(var i: integer; m: integer);  
begin  
  i:=i+k;  
  m:=j+1;  
  writeln(i,j,k,m);  
end;  
procedure p(var i:integer, j:integer);  
var  
  k:integer;  
begin  
  i:=i+k;  
  j:=j+k;  
  Q(i,j);  
end;  
begin  
  i:=1;  
  j:=2;  
  k:=3;  
  p(i,k);  
  writeln(i,j,k);  
end.
```

Completar la siguiente tabla para cada enunciado writeln suponiendo los mecanismos de paso de parámetros dados:

Modo de parámetros	i	j	k	m
Pascal, como esta escrito				
Todos los parametros se pasan por referencia				
Todos los parametros se pasan por valor				

Trabajo Práctico Nº 6

Tema: CONTROL DE SECUENCIA – CONTROL DE DATOS

Todos los parametros se pasan por valor-resultado				
---	--	--	--	--

21. En un lenguaje de programación, la implementación del ambiente local de referencia de un subprograma puede asumir dos enfoques: retención o eliminación. Explicar utilizando un ejemplo ambos enfoques y citar las ventajas y desventajas de cada uno.

22. Algunos lenguajes de programación utilizan la cadena dinámica (alcance dinámico) para establecer el ambiente no local de referencia de un subprograma. Proponga un ejemplo y explique el mecanismo. Grafique las formas de implementación y cite ventajas y desventajas de las mismas.

23. Algunos lenguajes de programación utilizan la cadena estática (alcance estático) para establecer el ambiente no local de referencia de un subprograma. Proponga un ejemplo y explique el mecanismo. Grafique las formas de implementación y cite ventajas y desventajas de las mismas.

24. Dados los siguientes esqueletos de programas, mostrar el estado de la pila central de activaciones (incluir cadena estática y cadena dinámica) cuando la ejecución del programa se encuentra en los puntos señalados:

a)

```
procedure BIGSUB;
  procedure A;
  procedure C;
  begin {B}
    ...
  end; {B}
  procedure C;
  begin {C}
    ...
    B;
    ...
  end; {C}
  begin {A}
    ...
    C;
    ...
  end;
begin {BIGSUB}
  ...
  A;
  ...
end; {BIGSUB}
```

b)

```
procedure BIGSUB;
  procedure A(flag:boolean);
  procedure B;
  ...
  A(false);
  ...
end; {B}
begin {A}
  if flag
  then B
  else C
  ...
end;{A}
  procedure C;
  procedure D;
  ...
end;{D}
.....
D;
end; {C}
A(true);
....
end; (BIGSUB)
```

La secuencia de llamada para este programa en ejecución que conduce a D es:

- BIGSUB calls A
- A calls B
- B calls A
- A calls C
- C calls D