

**Primer Parcial. Programación 1**  
**Instituto de Computación**  
**Octubre 2007**

<b>Nombre</b>	
<b>CI</b>	

<b>Número parcial</b>	
-----------------------	--

Observaciones generales:

- Cada ejercicio tiene una única opción correcta.
- Una respuesta correcta vale 2 puntos.
- Una respuesta incorrecta implica restar 0.5 puntos.
- Un ejercicio sin respuesta lleva 0 punto.
- Debe entregar la planilla con las respuestas junto con la letra.
- Debe poner nombre completo, cédula y número de parcial en la planilla y en la letra.

Observaciones sobre el lenguaje de programación:

Los ejercicios se basan en el lenguaje **Pascal** tal como fue dado en el curso. Éste es el Pascal estándar con los siguientes agregados:

- Utilización de **ELSE** en la instrucción **CASE**. Si no se especifica **ELSE**, se ejecutará la sentencia siguiente al **CASE**, en caso de no coincidir con ninguna de las etiquetas.
- Evaluación por **circuito corto** de las operaciones booleanas (**AND** y **OR** condicional), salvo que se especifique otra forma de evaluación.

**Ejercicio 1** ¿Dado el siguiente fragmento de código Pascal, qué salida produce al ser ejecutado?

```
x:=3;
y:=0;
while (x>0)and(y<=0)do
begin
  x:=x-1;
  y:=y-1;
  writeln(' ', x, ' ', y, ' ');
end;
```

a	Nada
b	(2, -1) (1, -2)
c	(2, -1) (1, -2) (0, -3)
d	(3, 0) (2, -1) (1, -2)
e	(3, 0)

**Ejercicio 2** Decir cuál invocación al procedimiento **p** es válida, para **m** y **n** variables enteras y **p** declarado como sigue:

```
PROCEDURE p(a,b:Integer; VAR v:Integer);
```

a	p(2,5,7)
b	n := p(2,5,m)
c	p(m,n,m)
d	p(m,n,0)
e	p(2,5,m) := n

**Ejercicio 3** Dadas las siguientes declaraciones:

```
Type Arreglo = array[-50 .. 50] of Integer;
Var lista :Arreglo;
```

¿Cuántas celdas tiene el arreglo lista?

a	49
b	50
c	51
d	100
e	101

**Ejercicio 4** Considere los siguientes fragmentos de programa, donde a, b y d son variables enteras

```
(1)
if (a > b) then
  if (d = 4) then d:= d+1
else
  Write('Suerte!');
```

```
(2)
if (a > b) then
begin
  if (d = 4) then
    d:= d+1
  else
    Write('Suerte!')
end;
```

```
(3)
if (a > b) then
begin
  if (d = 4) then
    d:= d+1
  end
else
  Write('Suerte!');
```

```
(4)
if (a > b) and (d = 4) then
  d:= d+1
else if not (a > b) then
  Write('Suerte!');
```

Indicar cuál de las siguientes afirmaciones es correcta:

a	Todas las opciones producen el mismo resultado.
b	(1) y (4) producen el mismo resultado.
c	Solo (3) y (4) producen el mismo resultado.
d	(1) y (2) producen el mismo resultado.
e	(1), (2) y (3) producen el mismo resultado.

**Ejercicio 5** Decir qué se despliega en la salida al ejecutarse el siguiente programa:

```
PROGRAM Ppal(OUTPUT);
  VAR v : Integer;
FUNCTION f(x:Integer; VAR y:Integer) : Integer;
  BEGIN
    y := y+2;
    x := v+3;
    v := v+1;
    f := v+x
  END;
PROCEDURE p(VAR x:Integer);
  VAR a : Integer;
  BEGIN
    a := f(x,x);
    v := 5 + a - x
  END;
BEGIN
  v := 2;
  p(v);
  WriteLn(v)
END.
```

a	13
b	12
c	2
d	8
e	16

**Ejercicio 6** ¿Cuál es la salida de cada una de las siguientes iteraciones?

```
(1)
i:=1;
repeat
  i:=3*i;
until (i>=3);
write(i);
```

```
(2)
i:=1;
while(i<=3)do
  i:=i*3;
write(i);
```

a	Para (1) es 3 y para (2) es 9
b	Para (1) es 9 y para (2) es 3
c	Para (1) y para (2) es 9
d	Para (1) y para (2) la iteración no termina nunca.
e	Para (1) es 3 y para (2) es 1

**Ejercicio 7** ¿Cuál de las siguientes declaraciones de tipos es válida?

a	Type Rango = '1'..'10';
b	Type Rango = A-Z;
c	Type Rango = 0,2,4..8;
d	Type Rango = 1..'10';
e	Type Rango = 'p'..'r';

**Ejercicio 8** Suponga que c1 y c2 son caracteres y num es real. Sean las expresiones E1 y E2:

E1: (c1 < c2) and (sqrt(num) < 10)

E2: (sqrt(num) < 10) and (c1 < c2)

Indique cuál de las siguientes afirmaciones es correcta:

a	Para cualquier valor de c1, c2 y num, E1 y E2 toman el mismo valor o ambas producen un error de ejecución.
b	Si c1 es 'a' y c2 es 'b', E1 es TRUE para cualquier valor de num.
c	Si c1 es 'b' y c2 es 'a' y num es -4.0, la evaluación de E1 produce un error de ejecución.
d	Si num es -4.0, la evaluación de E2 produce un error de ejecución para cualesquiera c1 y c2.
e	Si c1 es 'b' y c2 es 'a', entonces <b>no</b> se produce error de ejecución y E1 y E2 son ambas FALSE, para cualquier valor de num.

**Ejercicio 9** Decir qué se despliega en pantalla al ejecutarse el siguiente programa:

```
PROGRAM Prog(OUTPUT);
  VAR a,x : Integer;

  PROCEDURE p(a:Integer; VAR b:Integer);
  BEGIN
    a := a+2;
    b := a
  END;

  FUNCTION f(x,y:Integer) : Integer;
  BEGIN
    x := x+y;
    f := x+4
  END;

BEGIN
  p(10,a);
  x := f(a,7);
  WriteLn(a, ' ',x)
END.
```

a	19 23
b	12 23
c	12 12
d	19 7
e	12 7

**Ejercicio 10** Dadas las siguientes declaraciones:

Type A = array ['a'..'k'] of 0..999;

B = array [0..9] of Char;

Var x,y :A;

z: B;

¿Cuál de las siguientes asignaciones **no** es válida?

a	x['a'] := y['b'] / ord(z[3])
b	x := y
c	z[ x['e'] mod 10 ] := 't'
d	z[4] := z[ (x['f']+4) mod 9 ]
e	x['d'] := y['e'] div (x['f']+1)

**Ejercicio 11** Dados los siguientes fragmentos de código, indique qué afirmación es verdadera:

(1)

```
if (a > b) AND (c DIV d = 1) then
    Write ('La condición es verdadera')
else
    Write ('La condición es falsa');
```

(2)

```
if (c DIV d = 1) AND (a > b) then
    Write ('La condición es verdadera')
else
    Write ('La condición es falsa');
```

a	Ambos fragmentos se comportan de igual forma si las expresiones booleanas se evalúan por circuito corto, para cualquier valor de las variables $a, b, c, d$ .
b	Ambos fragmentos se comportan de igual forma si las expresiones booleanas se evalúan por circuito completo, para cualquier valor de las variables $a, b, c, d$ .
c	La evaluación por circuito corto de las expresiones booleanas garantiza que (1) nunca produzca un error al ejecutar.
d	Evalutando por circuito corto a las expresiones booleanas, (1) produce un error de ejecución si $a < b$ y $d = 0$ .
e	Evalutando por circuito corto a las expresiones booleanas, (2) no produce un error de ejecución si $a < b$ y $d = 0$ .

**Ejercicio 12**Cuál es la salida del siguiente programa en Pascal?

```
program ejercicio;
var i, j:Integer;
begin
    i:=0;
    while (i<=100) do
        begin
            j:=0;
            repeat
                i:= i +1;
                j:= j + 1;
            until j=10;
        end;
        writeln(i, ', ', j);
    end.
```

a	110,10
b	100,10
c	10,10
d	0,0
e	10,100

**Ejercicio 13** Considere la siguiente sentencia FOR:

for v := e1 to e2 do instrucción;

Determine qué afirmación es correcta:

a	En instrucción es correcto asignarle un valor a v.
b	Si $e1 > e2$ se produce un error.
c	Luego de la ejecución completa del ciclo, se puede afirmar que $v=e2$ .
d	Las expresiones $e1$ y $e2$ pueden tomar cualquier valor numérico.
e	La sentencia FOR siempre puede escribirse utilizando WHILE.

**Ejercicio 14** Dado el fragmento de código, donde a es de tipo integer,

```
FOR i:= 1 to 3 DO
    FOR j:= i to 3 DO
        Read(a);
        Write (a);
```

y la secuencia de entrada

```
1 2 3 2 3
3 4 5 6
```

Determinar qué afirmación es verdadera:

a	La instrucción <i>Read</i> se ejecuta tantas veces como la instrucción <i>Write</i> .
b	La salida en pantalla para este segmento de código es 1 2 3 2 3 3
c	La salida en pantalla para este segmento de código es 3
d	La salida en pantalla para este segmento de código es 1 2 3 2 3 3 4 5 6
e	El fragmento anterior tiene el mismo comportamiento que el siguiente: FOR i:= 1 to 3 DO BEGIN j:= i; REPEAT Read(a); j:=j+1; UNTIL j=3 END; Write(a);

**Ejercicio 15** Se considera la siguiente definición de tipos:

```
type
    AA=array [1..10] of
        record
            campo1:array [1..20] of integer;
            campo2: record
                c11: char;
                c22: boolean;
            end;
        end;
```

y la siguiente declaración de variables:

```
var
    i: integer;
    j: 1..10;
    c: AA;
```

Indique cuál de las siguientes expresiones es correcta

a	$c[i.campo1[13]]$
b	$c.campo2[j].c22$
c	$c[50]$
d	$c[i \bmod 10 + 1].campo2$
e	$c.campo1[i,2]$

**Ejercicio 16** ¿Cuál encabezado es adecuado para un subprograma que calcule la suma y el producto de dos valores enteros m y n?

a	FUNCTION sum_prod(m,n:Integer; VAR s,p:Integer);
b	FUNCTION sum_prod(m,n:Integer):Integer,Integer;
c	PROCEDURE sum_prod(VAR m,n:Integer; s,p:Integer);
d	PROCEDURE sum_prod(VAR m,n,s,p:Integer);
e	PROCEDURE sum_prod(m,n:Integer; VAR s,p:Integer);

**Ejercicio 17** Se considera el siguiente programa:

```

program puntos(input,output);
const
  espacio = ' ';
type
  punto = record
    x,y: real;
  end;
var
  a,b: punto;
  x: real;
begin
  x:= 0;
  with a do
  begin
    x := 8;
    y:= 9 / 2;
    b.x:= 16 / x;
    b.y := x * y;
  end;
  write(a.x:4:2,espacio,a.y:4:2,espacio,
    b.x:4:2,espacio,b.y:4:2,espacio,x:4:2);
end.

```

Indique cuál afirmación es correcta respecto al programa anterior:

a	Despliega 8.00 4.50 2.00 36.00 0.00
b	Al ejecutarse aborta por un error de división por cero.
c	Despliega 0.00 4.50 0.00 0.00 0.00
d	Despliega 8.00 4.50 2.00 36.00 8.00
e	Despliega 8.00 4.50 2.00 0.00 8.00

**Ejercicio 18** Dadas las siguientes declaraciones y asignaciones, indique cuál de ellas compila correctamente:

a	<pre> VAR   valor: real;   incremento: integer; BEGIN   valor:=4.0;   incremento:= valor DIV 2; </pre>
b	<pre> CONST   num1 = 100; VAR   resultado : real; BEGIN   resultado:=num1 + (num1/2); </pre>
c	<pre> CONST   letra = 'A'; TYPE arreglo=array [1..10] of char; VAR   palabra : arreglo; BEGIN   arreglo[1]:=letra; </pre>
d	<pre> CONST   fin = '\$'; VAR   letra1; letra2 : char; BEGIN   letra1:=fin;   letra2:=ord(letra1); </pre>
e	<pre> VAR   es_menor:boolean;   mayor:integer; BEGIN   mayor:=10;   es_menor:=mayor - 1; </pre>

**Ejercicio 19** Dado el siguiente programa en Pascal:

```

program ejercicio;
var a:integer;
begin
  a:=1;
  repeat
    a := a + 1;
    a := a mod 100;
  until (a = 100);
  writeln(a);
end.

```

Indique cuál afirmación es correcta.

a	Al ejecutarse, el programa escribe 1.
b	Al ejecutarse, el programa escribe 0.
c	Al ejecutarse, el programa no termina nunca.
d	Al ejecutarse, el programa escribe 2.
e	Al ejecutarse, el programa escribe 100.

**Ejercicio 20** El siguiente programa resuelve el problema de calcular la suma de los componentes de un arreglo de enteros, pero está incompleto.

```

PROGRAM sumar(INPUT,OUTPUT);
CONST N = 100;
TYPE Arreglo = ARRAY [1..N] OF Integer;
VAR a :Arreglo;
    i,suma :Integer;
BEGIN
  FOR i:=1 TO N DO
    read(a[i]);

    <para completar>

    writeln(suma)
END.

```

Indicar cuáles de las siguientes instrucciones deben ser escritas en lugar de <para completar> para que el programa resuelva correctamente el problema.

a	<pre> i:=1; suma:=0; repeat   suma := suma + a[i]; until i &gt; N; </pre>
b	<pre> for i:=1 to N   suma := suma + a[i]; </pre>
c	<pre> suma := a[N]; for i:= N-1 downto 1 do   suma := a[i] + suma; </pre>
d	<pre> suma := 0; for i:=1 to N do   suma := a[i]; </pre>
e	<pre> suma := a[1]; for i:= 1 to N do   suma := suma + a[i+1]; </pre>