

No. Parcial:
Nombre:
CI:

Primer Parcial. Programación 1

Instituto de Computación
año 2010

Observaciones:

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

Ejercicio 1 Considere el siguiente programa:

```
program uno;  
var x,y:Integer;  
begin  
  x:=3;  
  y:=0;  
  while (y>0)or(y<=0)do  
  begin  
    x:=x-1;  
    y:=y-1;  
  end;  
  writeln('(', x, ', ', y, ')');  
end.
```

¿Qué se exhibe en la salida cuando se ejecuta este programa?

- a) (2, -1), b) (1, -2), c) (3, 0), d) (0, 3)
e) no despliega nada y no termina.

Respuesta: e)

Ejercicio 2 Sea el siguiente programa:

```
var y : integer;  
procedure procl( VAR x : integer; y : Integer);  
  
  function f(x, y : integer) : integer;  
  begin  
    f := 2 * x + y;  
  
  end;  
begin  
  x := f(3, x);  
  x := x + f(x, 3);  
  x := x - y * 3;  
end;  
  
begin  
  y := 5;  
  procl(y, 3);  
  writeln(y);  
end.
```

El programa despliega en pantalla:

- a) 33, b) 23, c) 27, d) 5, e) 3

Respuesta: c)

Ejercicio 3 Considere el siguiente programa

```
program dos;  
var x:Integer;  
begin  
  x := -5;  
  while (abs(x) > 0) or (trunc(x)/x = 1)do  
    x := x + 1;  
    write (trunc(x))  
  end.
```

¿Cuál de las siguientes afirmaciones es correcta?

- a) el programa nunca termina.
b) el programa imprime 5 en la salida estándar.
c) el programa imprime 0 en la salida estándar.
d) el programa produce error en ejecución.
e) ninguna de las anteriores.

Respuesta: d)

Ejercicio 4 Considere las declaraciones:

```
procedure proc (a, b, c: integer);  
var z: integer;  
  
  function func (x: real) : integer;  
  procedure otras (var y: real);  
  begin  
    ... (* sentencias de otras *)  
  end; {otras}  
begin  
  ... (* sentencias de func *)  
end; {func}  
  
begin  
  ... (* sentencias de proc *)  
end; {proc}  
  
function otras (x, y: integer) : integer;  
var z: integer;  
begin  
  ... (* sentencias de otras *)  
end; {otras}
```

¿Cuál de las siguientes afirmaciones es verdadera?

- a) La función `func` puede ser invocada desde la función `otras`
b) El procedimiento `proc` no puede invocar a la función `func`
c) El procedimiento `proc` no puede invocar a la función `otras`
d) El procedimiento `otras` no puede invocarse a sí mismo.
e) El procedimiento `otras` puede ser invocado desde `proc`.

Respuesta: c)

Ejercicio 5 Sea el siguiente programa:

```
program tres;  
var i,num: integer;  
    b : boolean;  
begin  
  b:= true;  
  readln(num);  
  for i:= 30 downto 20 do  
    b:= b and (num*num <> i);  
  if not b then  
    write('no ');  
  writeln('se cumple')  
end.
```

La ejecución del programa anterior despliega el mensaje "no se cumple" si en la entrada se ingresa el valor:

- a) 2, b) 3, c) 5, d) 6, e) 25

Respuesta: c)

Ejercicio 6 Sea el fragmento de código:

```
const N = 5;
type arreglo = array [1 .. N] of integer;
var i : 1 .. N;
    A, secuencia : arreglo;
procedure ejemplo (incremento : integer; B : arreglo);
begin
    for i := 1 to N do
        B[i] := B[i] + incremento;
    A := B;
end;

begin
for i := 1 to N do
    secuencia[i] := i;
ejemplo(2, secuencia);
...
```

¿Cuáles son los contenidos de los arreglos A y secuencia luego de la invocación al procedimiento ejemplo?

- a) A = [1 2 3 4 5], secuencia = [1 2 3 4 5]
- b) A = [3 4 5 6 7], secuencia = [1 2 3 4 5]
- c) A = [1 2 3 4 5], secuencia = [3 4 5 6 7]
- d) A = [3 4 5 6 7], secuencia = [3 4 5 6 7]
- e) otros valores diferentes de los anteriores

Respuesta: b)

Ejercicio 7 Sea el siguiente programa

```
program cuatro;
const
    espacio = ' ';
    estrella = '*';
    N = 6;
var
    i, j : integer;
begin
    i := 1;
    repeat
        for j := (N-i) downto 1 do
            write(espacio);
        j := 1;
        while (j <= i) do
            begin
                write(estrella);
                j := j+1;
            end;
        writeln;
        i := i+1;
    until i > N
end.
```

Indique cuál de las siguientes es la salida que despliega:

- | a) | b) | c) | d) | e) |
|-------|-------|-------|-------|-----|
| ***** | ***** | * | * | * |
| ***** | ***** | ** | ** | ** |
| ***** | ***** | *** | *** | *** |
| *** | *** | **** | **** | *** |
| ** | ** | ***** | ***** | ** |
| * | * | ***** | ***** | * |

Respuesta: d)

Ejercicio 8 Sean las declaraciones

```
var entrada, result: real;
    contador: integer;
function ejemplo (num: real; coef: integer): real;
...
begin
...
end
```

Suponiendo que todas las variables fueron inicializadas ¿cuál de las siguientes es una instrucción correcta?

- a) if contador = 0 then ejemplo(2.0, 2)
- b) result := ejemplo (entrada / 2, contador)
- c) result := ejemplo (entrada DIV 2, contador)
- d) contador := ejemplo (2.0, 2)
- e) if ejemplo = 0 then write ('para esos valores, es cero')

Respuesta: b)

Ejercicio 9 De los siguientes fragmentos de programa hay uno que no despliega la misma salida que los otros. Indique cuál es.

- a) i := 1; while i <> 22 do begin writeln(i, '->', i*2); i := i+1; end;
- b) i := 2; while i <= 44 do begin writeln(i div 2, '->', i); i := i+2; end;
- c) for i := 1 to 21 do writeln(i, '->', i*2);
- d) i := 0; repeat i := succ(i); writeln(i, '->', i*2) until i = 21;
- e) i := 0; repeat i := succ(succ(i)); writeln(i div 2, '->', i) until i > 40;

Respuesta: b)

Ejercicio 10 Dadas las siguientes declaraciones:

```
const N = ...; (* algún valor mayor que 0 *)
type arr = array[1..N] of Char;
var
  a, b : arr;
  aux : Char;
  i, j : Integer;
```

Indicar cuál de los siguientes códigos intercambia los valores de las celdas de los arreglos. Por ejemplo, si originalmente $a[i] = 'x'$ y $b[i] = 'y'$, luego de ejecutar quedaría $a[i] = 'y'$ y $b[i] = 'x'$, para cualquier valor de i .

- a)
- ```
a := b;
b := a;
```
- b)
- ```
for i := 1 to N do
begin
  aux := a[i];
  for j := 0 to N - 1 do
    b[N - j] := aux;
  a[i] := b[i];
end;
a[i] := b[i];
for j := N downto 1 do
  b[j] := a[j];
```
- c)
- ```
for i := 1 to N do
 a[i] := b[i];
for j := N downto 1 do
 b[j] := a[j];
```
- d)
- ```
for i := 1 to N do
begin
  aux := a[i];
  a[i] := b[i];
  b[i] := aux;
end;
```
- e)
- ```
for i := 1 to N do
begin
 a[i] := b[i];
 b[i] := a[i];
end;
```

**Respuesta:** d)**Ejercicio 11** Dadas la declaración de variables siguiente:

```
VAR i, j : Integer;
 x, y : Real;
```

Diga cuál asignación produce un error de compilación.

- a)  $x := i + j$   
 b)  $i := trunc(j + j)$   
 c)  $i := i \bmod j + i$   
 d)  $j := i + (x \bmod i)$   
 e)  $y := i \bmod j$

**Respuesta:** d)**Ejercicio 12** Dada las siguientes declaraciones (donde N, M y P son constantes previamente definidas todas mayores que 0)

```
type MatrizNxM = array[1..N,1..M] of Integer;
type MatrizMxP = array[1..M,1..P] of Integer;
type MatrizNxP = array[1..N,1..P] of Integer;
var
 a : MatrizNxM;
 b : MatrizMxP;
 c : MatrizNxP;
 suma, i, j, k : Integer;
```

Indicar cual de los siguientes códigos almacena en c el resulta de la multiplicación de las matrices a y b

- a)
- ```
for i := 1 to N do
  for j := 1 to P do
    c[i, j] := a[i, j] * b[i, j];
```
- b)
- ```
for i := 1 to N do
 for j := 1 to P do
 begin
 suma := 0;
 for k := 1 to M do
 suma := suma + a[i, k] * b[k, j];
 c[i, j] := suma;
 end;
```
- c)
- ```
for i := 1 to N do
  for j := 1 to M do
  begin
    suma := 0;
    for k := 1 to P do
      suma := suma + a[i, k] * b[k, j];
    c[i, j] := suma;
  end;
```
- d)
- ```
for i := 1 to N do
 for j := 1 to M do
 begin
 c[i, k] := 0;
 for k := 1 to P do
 c[i, k] := c[i, k] + a[i, k] * b[k, j];
 end;
```
- e)
- ```
for i := 1 to N do
  for j := 1 to P do
  begin
    suma := 0;
    for k := 1 to M do
      suma := suma + b[i, k] * a[k, j];
    c[i, j] := suma;
  end;
```

Se recuerda que si A y B son matrices tales que

$$A = \begin{bmatrix} a_{11} & \dots & a_{1n} \\ \dots & \dots & \dots \\ a_{m1} & \dots & a_{mn} \end{bmatrix}$$

y

$$B = \begin{bmatrix} b_{11} & \dots & b_{1p} \\ \dots & \dots & \dots \\ b_{n1} & \dots & b_{np} \end{bmatrix}$$

entonces

$$A \times B = \begin{bmatrix} c_{11} & \dots & c_{1p} \\ \dots & \dots & \dots \\ c_{m1} & \dots & c_{mp} \end{bmatrix}$$

donde

$$c_{ij} = a_{i1}b_{1j} + a_{i2}b_{2j} + \dots + a_{in}b_{nj}$$

Respuesta: b)

Ejercicio 13 Dada la siguiente declaración de variables:

```
VAR n : Integer;
    a,b : Char;
```

Diga cuál asignación NO produce un error de compilación.

- a) `n := ord(1.5)`
- b) `n := chr('1')`
- c) `n := ord(a) div ord(b)`
- d) `'a' := b`
- e) `a := b + '1'`

Respuesta: c)

Ejercicio 14 Dada las siguientes declaraciones:

```
const MAX = ... ; (* algún valor mayor que 1 *)
type Arreglo = array[1..MAX] of Integer;
var a:Arreglo; i:Integer;
```

¿Cuál de los siguientes códigos imprime el contenido de todas las celdas de un arreglo una sola vez?

- a)

```
for i:= 1 to MAX div 2 do
  writeln(a[2*i])
```
- b)

```
i:=1;
repeat
  writeln(a[i]);
  i:=i+1
until (i=MAX)
```
- c)

```
i:=1;
while (i<MAX) do
begin
  writeln(a[i]);
  i:=i+1
end
```
- d)

```
for i:=MAX downto 1 do
  writeln(a[MAX-i+1])
```
- e)

```
for i:= 1 to (MAX div 2) do
  writeln(a[i]);
for i:= MAX downto (MAX div 2) do
  writeln(a[i]);
```

Respuesta: d)

Ejercicio 15 Diga cuál es la salida del siguiente programa:

```
program seleccion(input, output);
var b1,b2 : Boolean;
    n : Integer;
begin
  b1 := TRUE;
  b2 := FALSE;
  n := 3;
  if b1 then
    if b2 then
      case n of
        1: writeln('caso 1');
        2: writeln('caso 2');
        else writeln('otro caso')
      end
    else
      begin
        n := 1;
        writeln('n es 1')
      end
    else
      writeln('caso final')
  end.
end.
```

- a) El programa despliega el mensaje: otro caso
- b) El programa despliega el mensaje: caso 3
- c) El programa despliega el mensaje: n es 1
- d) El programa despliega el mensaje: caso final
- e) El programa no despliega nada.

Respuesta: c)

Ejercicio 16 Dada las siguientes declaraciones:

```
type Arr = array[1..20] of Integer;
var a:Arr; i:Integer;
```

¿Cuál de los siguientes códigos cambia al doble el valor guardado en cada celda del arreglo a?

- a)

```
for i:= 1 to 10 do
  a[2*i] := 2*a[2*i]
```
- b)

```
for i:= 1 to 20 do
  a[2*i] := a[i]
```
- c)

```
for i:= 1 to 20 do
  a[i] := a[i]+a[i]
```
- d)

```
for i:= 1 to 20 do
  a[i] := a[i+i]
```
- e)

```
for i:= 1 to 20 do
  if i mod 2 = 0 then
    a[i] := 2*a[i]
  else
    a[i+1] := 2*a[i+1]
```

Respuesta: c)

Ejercicio 17 Sea el siguiente programa:

```
program nueve;
var
    b, res : integer;
    a : real;
    cond1 : boolean;
begin
    a := 5.8;
    b := trunc(a);
    res := 0;
    cond1 := (ord('$') mod 2) = 0;

    case cond1 of
        true, false : res := res + 1;
    end;

    if ( (res mod 2) = 0) and cond1) then
        res := res * b
    else
        res := res + b;
    case res of
        10: if (cond1) then
            res := res + 1;
        6: if ( (res <> b) or cond1 ) then
            res := res - 1;
        else
            res := res * 2;
    end;
    writeln(res);
end.
```

Indique la afirmación correcta:

- a) Se necesita conocer el valor de `ord('$')` para determinar la salida del programa
- b) El programa despliega el valor 11
- c) El programa despliega el valor 0
- d) El programa despliega el valor 5
- e) El programa despliega el valor 10

Respuesta: d)

Ejercicio 18 Indique cuál de las siguientes declaraciones define un arreglo que permite almacenar una cadena de diez caracteres alfanuméricos cualesquiera.

- a) `type Arreglo = array ['1'..'10'] of Char;`
- b) `const rangoIni = 1; rangoFin = 10; type Arreglo = array [rangoIni .. rangoFin] of 'A'..'Z';`
- c) `type Arreglo = array [1,2,3,4,5,6,7,8,9,10] of Char;`
- d) `type Arreglo = array [1.0 .. 10.0] of Char;`
- e) `const ini =1; fin = 10; type Rango = ini .. fin; Arreglo = array [Rango] of Char;`

Respuesta: e)

Ejercicio 19 Sea la siguiente declaración de variables y las dos primeras sentencias del cuerpo principal:

```
var
    a : boolean;
    cond1, cond2 : boolean;
begin
    cond2 := false;
    cond1 := not cond2;
```

Indique cuál instrucción, ejecutada luego de las anteriores, deja en la variable `a` el valor `false`.

- a) `a := not cond1 and not cond2;`
- b) `if cond2 then begin if cond1 then a := false else a := true; end;`
- c) `if cond2 then if cond1 then a := true else a := false;`
- d) `case cond1 of true: if (not cond1) then a := true; false : if (cond1) then a := false else a := true; end;`
- e) `a := not cond2 or cond1;`

Respuesta: a)

Ejercicio 20 Sea el siguiente programa:

```
program diecinueve;
CONST
    Max = 6;
var
    i, j: integer;
    a : array [1..Max] of integer;

    procedure d( VAR j : integer);
    begin
        j := i + 1;
    end;
    procedure e (j : integer);
    begin
        j := j - 2; i := j;
    end;
    procedure F( n : integer);
    begin
        e(n); d(n);
        a[n+1] := a[i] + a[n];
    end;
begin
    for j := 1 to Max do
        a[j] := (j-1) mod 2;
    for j := 3 to Max do
        F(j);
    end.
```

Indique el contenido del arreglo `a` al finalizar el programa:

- a) El contenido del arreglo `a` es: 0 1 1 2 3 5
- b) El contenido del arreglo `a` es: 0 1 1 2 2 1
- c) El contenido del arreglo `a` es: 0 1 1 3 5 8
- d) El contenido del arreglo `a` es: 0 1 0 1 0 1
- e) El contenido del arreglo `a` es: 0 1 1 3 2 5

Respuesta: a)