

No. Parcial:
Nombre:
CI:

Primer Parcial. Programación 1

Instituto de Computación
Año 2011

Observaciones:

- Cada ejercicio tiene una única opción correcta.
- Una respuesta correcta suma 2 puntos
- Una respuesta incorrecta resta 0.5 puntos.
- Un ejercicio sin respuesta no afecta el puntaje.
- 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 las siguientes declaraciones de variables:

```
var
    resultado, total: integer;
    promedio: real;
    car1, car2: char;
```

Indique cuál de las siguientes **no** es una expresión real:

- a) total - resultado * (total / promedio)
- b) abs(total) DIV abs(resultado) + promedio
- c) total - ord(car1) * ord(car2)
- d) sqrt(resultado) * trunc(promedio)
- e) sqr(resultado) * exp(total)

Respuesta: c)

Ejercicio 2 Considere el programa

```
program uno;
var
    i, j : integer;
begin
    i:= 5;
    while i> 0 do
    begin
        for j:= i downto 1 do
            write(' *');
        writeln('=');
        i:= i - 2
    end
end.
```

Indique cuál es la salida

a)	b)	c)	d)	e)
*****=	*****=	*****	=	*****=
***=	***=	***	***=	***=
*=	*=	*	*****=	=
=	=			

Respuesta: a)

Ejercicio 3 El producto escalar de dos vectores a y b se define como: $a_1 \times b_1 + a_2 \times b_2 + \dots + a_n \times b_n$

Dada las siguientes declaraciones:

```
CONST N = ...;
TYPE Vector = ARRAY[1..N] OF Real;
VAR a, b: Vector; i: Integer; r: Real;
```

Indicar cuál de los siguientes códigos almacena en la variable r el producto escalar de los vectores a y b .

a)
for i := 1 to N do
 r[i] := a[i] * b[i];

b)
for i := 1 to N do
 r := a[i] * b[i]

c)
r := a * b

d)
for i := 1 to N do
 r[i] := a[i] * b[i];
r:= 0;
for i := 1 to N do
 r := r[i] + r

e)
r:= 0;
for i := 1 to N do
 r := a[i] * b[i] + r

Respuesta: e)

Ejercicio 4 Diga cuál de las siguientes afirmaciones, sobre el programa (incompleto) que se da a continuación, es verdadera:

```
procedure p(x,y: integer);
var z: integer;

function f(a: integer) : integer;
var b: integer;
begin
    ... (* sentencias de f *)
end; {f}

function g(c: real) : integer;
procedure k(var d: real);
begin
    ... (* sentencias de k *)
end; {k}
begin
    ... (* sentencias de g *)
end; {g}

begin
    ... (* sentencias de p *)
end; {p}
```

- a) La función k puede ser llamada desde las sentencias de p .
- b) La función g puede ser llamada desde las sentencias de p y dentro de las propias sentencias de g pero no desde las sentencias de k .
- c) La función g no puede ser llamada desde f .
- d) La función f no puede ser llamada desde g .
- e) El procedimiento k puede ser llamado desde fuera de g .

Respuesta: c)

Ejercicio 5 Sean las siguientes declaraciones de variable:

```
var
    x: integer;
    p: boolean;
    y: Integer;
    c: Char;
```

Indicar cuál de las siguientes instrucciones **no** provoca un error al compilar:

- a) $p := p$ and $(x > y)$
- b) case x of a : $y := y + 1$ else $y := 0$ end
- c) $p := \text{ord}(c) > \text{ord}(c+1)$
- d) read(p) or read(x)
- e) write ($p := \text{not } p$)

Respuesta: a)

Ejercicio 6 Se considera el siguiente programa:

```
program bx(input,output);
var a,m: integer;
begin
  a:= 1;
  for m:= 5 to 1 do
  begin
    a:= m * a;
    writeln(a);
  end;
end.
```

Indique cuál de las siguientes afirmaciones es correcta con respecto al programa anterior:

- a) No compila correctamente.
- b) Produce un error en tiempo de ejecución.
- c) Ejecuta sin errores y despliega 5 números en la salida.
- d) Ejecuta sin errores y no produce ninguna salida.
- e) No termina nunca.

Respuesta: d)

Ejercicio 7 El *reverso* de un arreglo es otro arreglo que tiene los mismos elementos pero en orden inverso.

Por ejemplo si un arreglo contiene los elementos: (a_1, a_2, \dots, a_n) su reverso sería: $(a_n, a_{n-1}, \dots, a_2, a_1)$

Dadas las siguientes declaraciones:

```
CONST N = ...;
TYPE Arreglo = ARRAY[1..n] OF Char;
VAR a, b: Arreglo; i: Integer; c: Char;
```

Indicar cuál de los siguientes códigos transforma el arreglo a en su reverso:

a)

```
FOR i := 1 TO N div 2 DO
BEGIN
  c := a[N-i+1];
  a[N-i+1] := a[i];
  a[i] := c
END
```

b)

```
FOR i := 1 TO N div 2 DO
BEGIN
  a[i] := a[N-i+1];
  a[N-i+1] := a[i]
END
```

c)

```
FOR i := 1 TO N DO
BEGIN
  c := a[N-i+1];
  a[N-i+1] := a[i];
  a[i] := c
END
```

d)

```
FOR i := 1 TO N DO
BEGIN
  a[i] := a[N-i+1];
  a[N-i+1] := a[i]
END
```

e)

```
FOR i:= 1 TO N DO b[N-i+1] := a[i];
FOR i:= 1 TO N DO a[N-i+1] := b[i];
```

Respuesta: a)

Ejercicio 8 ¿Cuál es la salida del siguiente programa?

```
program Prog(input, output);
var a,b,c : Char;
function f(var v : Char) : Char;
begin
  v := succ(v);
  v := succ(v);
  case v of
    'c' : f := 's';
    'd' : f := 'u';
    'e' : f := 'y'
  else
    f := 'z'
  end
end;
procedure Caracteres(var c1 : Char; c2 : Char);
var a : Char;
begin
  a := c2;
  c2 := succ(a);
  b := f(c1);
  c := a;
  write(a,b,c)
end;
begin
  a := 'a';
  b := 'b';
  c := 'c';
  caracteres(a,b);
  write(a,b,c)
end.
```

- a) cscabc
- b) bsbabc
- c) bsbcsb
- d) abcabc
- e) cscsc

Respuesta: c)

Ejercicio 9 Sea j una variable entera a la cual se le asigna el valor (-4). Indique cuál de las siguientes opciones es la correcta:

a) Cuando se ejecuta:

```
if (sqrt(j)<2) or (j<0) then
  write(1)
else write(2)
```

la salida es 1 si la expresión booleana se evalúa por circuito corto y produce un error si la evaluación es por circuito completo.

b) Cuando se ejecuta:

```
if (j<0) or (sqrt(j)<2)
then write(1)
else write(2)
```

la salida es 1 si la expresión booleana se evalúa por circuito corto y produce un error si la evaluación es por circuito completo.

c) Cuando se ejecuta:

```
if (j<0) or (sqrt(j)<2)
then write(1)
else write(2)
```

la salida es 1 independientemente de la forma de evaluación de la expresión booleana(circuito corto o completo).

d) Cuando se ejecuta:

```
if (sqrt(j)<2) or (j>0)
then write(1)
else write(2)
```

produce un error si la expresión booleana se evalúa por circuito corto, mientras que escribe 2 en la salida si la evaluación es por circuito completo

e) Ninguna de las opciones anteriores es correcta.

Respuesta: b)

Ejercicio 10 Se quiere hacer un programa que muestre todas los caracteres desde la 'A' a la 'Z' con su correspondiente número ordinal como se muestra a continuación:

```
A --> 65
B --> 66
C --> 67
D --> 68
...
Z --> 90
```

Indique cuál de los siguientes códigos realiza lo indicado suponiendo que la variable *k* es entera y la variable *c* es de tipo char

```
a)
for k:= 'A' to 'Z' do
write(chr(k), '-->', k)
```

```
b)
c:= 'A';
while c <= ord('Z') do
begin
writel(c, '-->', ord(c));
c:= c + 1;
end
```

```
c)
c:= pred('A');
repeat
c:= succ(c);
writel(c, '-->', ord(c));
until c = 'Z'
```

```
d)
for k:= ord('A') to ord('Z') do
begin
writel(chr(k), '-->', k);
k:= k + 1;
end
```

```
e)
k:= ord('A');
while k < ord('Z') do
begin
writel(chr(k), '-->', k);
k:= k + 1;
end
```

Respuesta: c)

Ejercicio 11 Dadas las siguientes definiciones:

```
CONST N = ...;
TYPE Arreglo = ARRAY[1..N] OF Boolean;
VAR b : Arreglo; i : Integer;
```

Indicar cuál de los siguientes códigos inicializa un arreglo de booleanos de tal forma que las celdas de índice impar contengan el valor true y las otras el valor false. La constante *N* puede ser par o impar.

```
a)
FOR i := 1 TO N DIV 2 DO
BEGIN
b[i*2] := false;
b[i*2+1] := true
END
```

```
b)
FOR i := 1 TO N DO
if i mod 2 = 0 then
b[i] := false
else
b[i] := not b[i-1]
```

```
c)
FOR i := 1 TO N DIV 2 DO
BEGIN
b[i*2] := false;
b[i*2-1] := true
END
```

```
d)
FOR i:= 1 TO N DO
b[i] := i mod 2 = 1
```

```
e)
FOR i:= 1 TO N DIV 2 DO
b[i*2] := false
```

Respuesta: d)

Ejercicio 12 ¿Cuál es la salida del siguiente programa?

```
program p(input, output);

var p1, p2 : Integer;

procedure proc(a: Integer; var b :Integer);
var p1 : Integer;
begin
p1 := a+b;
b := p1+2;
p2 := a+1;
end;

begin
p1 := 1;
p2 := p1+3;
proc(p1,p2);
write(p1, ' ', p2);
end.
```

- a) 1 4
- b) 5 8
- c) 5 4
- d) 5 2
- e) 1 2

Respuesta: e

Ejercicio 13 Considere el siguiente fragmento de programa.

```

read (a);
if (a < '0') or (a > '9') then
  begin
    x := 8;
    writeln(x)
  end
else x := 1;
case (x mod 3) of
  1: writeln(sqr (x+2));
  2: writeln(sqrt (2*x):3:2)
else
  writeln(x)
end

```

Indique cuál de las siguientes opciones es correcta suponiendo que la variable a es de tipo char y la variable x es de tipo integer.

- a) si se ingresa el carácter 1 se produce un error de ejecución
- b) si se ingresa el carácter A se escriben los valores 8 y 4.00
- c) si se ingresa el carácter 0 se escribe 8 y 9
- d) si se ingresa el carácter 3 se escribe 8
- e) si se ingresa el texto "hola" se produce un error de compilación

Respuesta: b

Ejercicio 14 Se considera el siguiente programa:

```

program AB(output);
var k,i, f: integer;
begin
  f:= 1;
  for k:= 1 to 4 do
  begin
    i:= 1;
    repeat
      f:= f * i;
      i:= i + 1
    until i > k;
    writeln(f);
  end;
end.

```

Indique cuál de las siguientes afirmaciones es correcta:

- a) El programa despliega los factoriales de los números 1,2,3 y 4.
- b) El último número que despliega el programa es mayor que el factorial de 4.
- c) El programa no produce ninguna salida.
- d) El programa no compila.
- e) El primer número que despliega el programa es mayor que factorial de 4.

Respuesta: b)

Ejercicio 15 Dadas las siguientes declaraciones:

```

CONST N = ...;
TYPE Arreglo = ARRAY[1..N] OF Integer;
VAR a : Arreglo; i : Integer; resultado : Boolean;

```

Indicar cuál de los siguientes códigos deja en la variable resultado el valor true si el arreglo a está ordenado de menor a mayor y false en caso contrario.

```

a)
i := 0;
WHILE (a[i] <= a[i+1]) AND (i < N) DO
  i := i+1;
resultado := i = N

```

```

b)
i := 0;
REPEAT
  i := i+1;
UNTIL NOT (a[i] < a[i+1]) AND (i < N);
resultado := i = N

```

```

c)
i := 1;
WHILE (i < N) AND (a[i] <= a[i+1]) DO
  i := i+1;
resultado := i = N

```

```

d)
resultado := true;
FOR i := 1 TO N-1 DO
  resultado := a[i] < a[i+1]

```

```

e)
resultado := false;
FOR i := 1 TO N-1 DO
  resultado := (a[i] < a[i+1]) AND resultado

```

Respuesta: c)

Ejercicio 16 Diga cuál es la salida del programa siguiente:

```

program p(input, output);
var x, y : Integer;

  procedure proc(a: Integer; var b :Integer);
  var z : Integer;
  begin
    z := a;
    y := a+5;
    b := (z+20) div b;
    x := sqr(a)
  end;

begin
  x := 5;
  y := x+3;
  proc(x,y);
  write(x, ' ', y);
end.

```

- a) 5 8
- b) 25 2
- c) 25 10
- d) 5 3
- e) 5 10

Respuesta: b

Ejercicio 17 Sean las declaraciones:

```

var palo : 1 .. 4;
  valor: 1 .. 12;

```

Considere que los valores ingresados en la entrada son 3 y 7. Indique cuál fragmento de código despliega los dos mensajes: "es un 7 pero no es de oro" y "es un valor de oro".

```

a)
readln (palo, valor);
if (palo <= 3) and (valor = 7) then
  writeln ('es un 7 pero no es de oro')
else writeln ('es un valor de oro');

```

```

b)
readln (palo, valor);
if (palo > 3) or (valor <> 7) then
  writeln ('es un valor de oro')
else writeln ('es un 7 pero no es de oro');

```

```
c)
if (palo = 3) then
  if (valor <= 7) then
    writeln ('es un 7 pero no es de oro')
  else writeln ('es un valor de oro')
else writeln ('es sota, caballo o rey, de oro');
```

```
d)
case palo of
  1, 2, 3:
    writeln ('es un ', valor, ' pero no es de oro');
  4:
    writeln ('es un ', valor, 'de oro');
end;
```

```
e)
readln (palo, valor);
if (palo <= 3) then
  if (valor = 7) then
    writeln ('es un 7 pero no es de oro');
  writeln ('es un valor de oro');
```

Respuesta: e)

```
c)
primero := a[1];
i := 2;
WHILE (i <= N) AND (a[i] = primero) DO
  i := i+1;
if (i = N + 1) then
  res := -1
else
  res := i;
```

```
d)
i := 1;
primero := a[1];
REPEAT
  i := i+1;
UNTIL NOT (a[i] < a[i+1]) AND (i < N);
if (i = N) then
  res := -1
else
  res = a[i];
```

```
e)
res := -1;
primero := a[1];
FOR i := N TO 2 DO
  if (primero <> a[i]) then
    res := a[i];
```

Respuesta: c)

Ejercicio 18 Sea el programa

```
program UU(output);
var i,a: integer;
begin
  a:= 2;
  for i:= 1 downto -1 do
    a:= a * a;
  writeln(a)
end.
```

El valor que despliega el programa es:

a) 2, b) 4, c) 8, d) 256, e)32

Respuesta: d)

Ejercicio 20 Diga cuál de las siguientes afirmaciones sobre los subprogramas es falsa:

- Los subprogramas pueden ser invocados varias veces desde otras partes del programa.
- Los subprogramas se comunican mediante el pasaje de parámetros.
- Cada subprograma tiene su propio espacio de nombres (identificadores locales).
- Dentro de un mismo programa no pueden existir un procedimiento y una función con el mismo nombre.
- Los identificadores del programa principal pueden ser utilizados dentro de los subprogramas.

Respuesta: d)

Ejercicio 19 Dadas las siguientes definiciones:

```
CONST N = ...;
TYPE Arreglo = ARRAY[1..N] OF Integer;
VAR a : Arreglo; i, res, primero: Integer;
```

Indicar cuál de los siguientes códigos guarda en la variable `res` el índice del primer elemento del arreglo distinto de `a[1]`, o el valor `(-1)` en caso que no exista tal elemento.

```
a)
primero := a[1];
res := 2;
WHILE (res < N) and (a[res] = primero) DO
  res := res+1;
if (res = N) then
  res := -1
```

```
b)
res := 1;
primero := a[1];
REPEAT
  res := res+1;
UNTIL NOT (a[res] < a[res+1]) AND (res < N);
if (res = N) then
  res := -1
```