

Nº Parcial:

Nombre:

CI:

Respuesta: <sup>3</sup>

## Primer Parcial. Programación 1

Instituto de Computación  
Año 2015

### 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.
- En todos los ejercicios se utiliza el lenguaje Pascal tal como fue dado en el curso.

**Ejercicio 1** Dadas las siguientes instrucciones, en donde final y valor son variables enteras:

```
final := 5; valor := 108;
```

- 1) while (final > 0) and (valor DIV final < 300) do  
    final := final - 1
- 2) while (valor DIV final < 300) and (final > 0) do  
    final := final - 1

¿Cuál de las siguientes afirmaciones sobre las instrucciones 1) y 2) es correcta?

- a) Ninguna de las instrucciones produce error en tiempo de ejecución.
- b) Ambas instrucciones producen error en tiempo de ejecución.
- c) Solo la instrucción 1 produce error en tiempo de ejecución.
- d) Solo la instrucción 2 produce error en tiempo de ejecución.
- e) Si final se inicializa en 0, en vez de 5, ambas producen error de ejecución.

Respuesta: <sup>1</sup>

**Ejercicio 2** Dado el siguiente programa:

```
program ejercicio;  
var x, y: Integer;  
begin  
  read(x);  
  y := x;  
  if x mod 2 = 0  
  then y := x + 2;  
  if (x + y) mod 2 = 0  
  then write('a')  
  else write('b')  
end.
```

Indicar la opción correcta al ejecutarlo.

- a) Despliega 'a' si y solo si x es par.
- b) Despliega 'a' si y solo si x es impar.
- c) Siempre despliega 'b'.
- d) Despliega 'b' si y solo si x es impar.
- e) Siempre despliega 'a'.

Respuesta: <sup>2</sup>

**Ejercicio 3** ¿Cuál de las siguientes expresiones produce un error de compilación?

- a)  $4 < 5$  and  $6 > 3$
- b) true
- c) 'A' < 'B'
- d) false and true
- e)  $2 + 1 < 3 + 4$

**Ejercicio 4** Dadas las siguientes declaraciones de variables:

```
VAR  
  num1, num2 : Integer;  
  r1, r2 : Real;  
  c : Char;
```

Indique cuál asignación es válida:

- a)  $r2 := c$
- b)  $num1 := r1 + r2$
- c)  $c := 'a' + 'b'$
- d)  $r1 := num1 + num2$
- e)  $c := 'r1'$

Respuesta: <sup>4</sup>

**Ejercicio 5** Dadas las siguientes instrucciones, indicar la opción correcta, asumiendo que siempre se ingresan 3 valores distintos.

```
var a, b, c : Integer;  
begin  
  readln(a, b, c);  
  if a < b then  
    if b < c then  
      writeln(a)  
    else if c < a then  
      writeln(c)  
    else writeln(a)  
  else if c < b then  
    writeln(c)  
    else writeln(b)  
end.
```

- a) Imprime el mínimo de los 3 valores ingresados si y solo si  $a < b$  y  $c > a$ .
- b) Imprime el mínimo de los 3 valores ingresados si y solo si  $a < b$  y  $b > c$ .
- c) Imprime el mínimo de los 3 valores ingresados si y solo si  $a < b$  y  $b < c$ .
- d) En todos los casos imprime el mínimo de los tres valores.
- e) Imprime el mínimo de los 3 valores ingresados si y solo si  $a > b$  y  $c < b$ .

Respuesta: <sup>5</sup>

**Ejercicio 6** Dado el siguiente fragmento de código:

```
var  
  i, j: integer;  
  b : boolean;  
begin  
  b := false;  
  j := 0;  
  for i := 3 to 5 do  
  begin  
    while (j < i) and (not b) do  
    begin  
      write('S');  
      j := j + 1  
    end;  
    while (j > 0) and b do  
    begin  
      write('B');  
      j := j - 1  
    end;  
    b := not b  
  end  
end.
```

Indique cuál es la salida generada por su ejecución:

- a) SSSSS
- b) SSSBBBBSSSS
- c) SSSBBBBSSSSBBBBSSSSSSBBBB
- d) SBSSBBBBSSSS
- e) SSSBBBB

**Respuesta:** <sup>6</sup>

**Ejercicio 7** Considere el siguiente programa:

```

program prog;
type
  rango = 1 .. 3;
  arreglo = array [rango] of Integer;
var
  a : arreglo;
  i : rango;
begin
  for i := 1 to 3 do a[i] := i;
  i := 1;
  i := i+1;
  a[i] := a[i+1];
  a[i+1] := a[i] + 1;
  for i := 1 to 3 do write(a[i], ' ');
  i := 1;
  a[i+1] := a[i];
  i := i+1;
  a[i-1] := a[i] - 1;
  for i := 1 to 3 do write(a[i], ' ');
end.

```

Indicar cuál de las opciones se imprime:

- a) 2 3 3 -1 3 3
- b) 1 3 3 2 3 3
- c) 1 3 4 0 1 4
- d) 2 2 4 2 3 4
- e) 1 3 4 0 1 3

**Respuesta:** <sup>7</sup>

**Ejercicio 8** Dado el siguiente programa:

```

program seleccion;
var n : integer;
begin
  readln(n);
  case n of
    0: writeln('cero');
    1: writeln('uno');
    2: writeln('dos');
    > 2: writeln('mayor');
  end;
  writeln('final')
end.

```

Diga cuál afirmación es verdadera.

- a) Si  $n > 2$  el programa da error en tiempo de ejecución.
- b) Si  $n < 0$  el programa da error en tiempo de ejecución.
- c) El programa no compila.
- d) El programa siempre da error en tiempo de ejecución.
- e) Si  $n < 0$  el programa despliega el mensaje 'final'.

**Respuesta:** <sup>8</sup>

**Ejercicio 9** Sean las instrucciones siguientes:

```

I1) while cond1 do
      cuerpo1

I2) repeat
      cuerpo2
until cond2

```

Diga cuál afirmación es falsa.

- a) cuerpo1 puede no ejecutarse nunca.
- b) I1 termina cuando cond1 es falsa.
- c) I2 termina cuando cond2 es verdadera.
- d) cuerpo2 se ejecuta al menos una vez.
- e) I1 es equivalente a repeat cuerpo1 until not cond1.

**Respuesta:** <sup>9</sup>

**Ejercicio 10** ¿Cuál expresión **no** es de tipo Integer?

- a)  $19 \bmod 5 / 2$
- b)  $10 + \text{trunc}(2.5)$
- c)  $2 * 3 + 4$
- d)  $\text{ord}('a') + \text{ord}('5')$
- e)  $\text{trunc}(\text{sqrt}(16))$

**Respuesta:** <sup>10</sup>

**Ejercicio 11** Dados los siguientes fragmentos de código:

```
var c: Char;
begin
  readln(c);
  while ('0' <= c) and (c <= '9') do
  begin
    repeat
      c:= succ(c);
    until c = 'Z';
    writeln(c);
  end
end.
```

```
var c: Char;
begin
  repeat
    readln(c);
    while (c <> 'Z') do
      c:= succ(c);
    writeln(c);
  until (c < '0') or (c > '9');
end.
```

Teniendo en cuenta que los elementos de tipo Char están ordenados del siguiente modo (los tres puntos indican las letras del alfabeto que faltan):

```
! "# $ % & ' ( ) * + , - . / 0 1 2 3 4 5 6
7 8 9 : ; <= > ? @ A ... Z [ \ ] ^ _ ` a ...
z { | } ~
```

Marque la opción correcta:

- a) Si la entrada es Z, uno de los códigos da error de ejecución.
- b) Si la entrada es a, ninguno de los códigos da error de ejecución.
- c) Si la entrada es @, ambos códigos imprimen lo mismo.
- d) Si la entrada es un dígito, ambos códigos imprimen lo mismo.
- e) Si la entrada es Z, ambos códigos dan error de ejecución.

**Respuesta:** <sup>11</sup>

**Ejercicio 12** Dado el siguiente fragmento de código:

```
var
  x, y: integer;
begin
  x := 240;
  y := 0;
  if (x <> 0) and (y mod x = 0) then
    write('si')
  else
    write('no');
  if (y <> 0) and (x mod y = 0) then
    write('si')
  else
    write('no')
end.
```

Indique cuál es el resultado de su ejecución:

- a) Imprime sino
- b) Imprime nono
- c) Imprime nosi
- d) Imprime sisi
- e) Se genera un error en tiempo de ejecución.

**Respuesta:** <sup>12</sup>

**Ejercicio 13** Se considera el programa:

```
program pp;
var i, n : integer;
begin
  readln(n);
  while (n > 0) do
  begin
    for i:= n to 0 do
      n:= n * n;
      n := n - 1
    end;
    writeln(n)
  end.
```

Indicar cuál será la salida cuando se ingresa el valor 4:

- a) 4
- b) 134
- c) 0
- d) Se produce un error en tiempo de ejecución.
- e) El programa no termina nunca y no despliega nada.

**Respuesta:** <sup>13</sup>

**Ejercicio 14** Para que el siguiente código:

```
program p;
const MAX = 10;
      MITAD = 5;
var i, j : integer;
    c1, c2, c3 : char;
begin
  readln(c1, c2, c3);
  for i:= 1 to MAX do begin
    for j := i to MAX do
      if (j <= MITAD) then
        write(c2)
      else
        write(c3);
    writeln;
    for j := 1 to i do
      write(c1)
    end
  end.
```

Produzca la siguiente salida en pantalla:

```
&&&&&@#####
#&&&&&@#####
##&&&&@#####
###&&@#####
####&@#####
#####@#####
#####@#####
#####@#####
#####@#####
#####@#####
#####@#####
#####@#####
#####@#####
#####@#####
```

La entrada ingresada por el usuario debe ser:

- a) #&@
- b) &#@
- c) @#&
- d) @&#
- e) @#@

**Respuesta:** <sup>14</sup>

**Ejercicio 15** Dada la siguiente declaración de constantes y variables:

```
const
  N=...; (*entero mayor o igual a 1*)
  P=...; (*entero mayor o igual a 1*)
var
  num, sum, i, j : Integer;
```

Indique cuál de los siguientes fragmentos de código deja en la va-

riable sum la suma de las P-potencias de los primeros N números naturales con  $P > 0$  :

$$\sum_{k=1}^N k^P = 1^P + 2^P + 3^P + \dots + N^P$$

```
a)
sum := 1;
for i:= N downto 2 do begin
  num := i;
  for j:= P downto 2 do
    num := num*num;
  sum := sum + num
end;
```

```
b)
sum := 1;
for i:= 1 to N do begin
  num := i;
  for j:= 2 to P do
    num := num*i;
  sum := sum + num
end;
```

```
c)
sum := 0;
for i:= N downto 2 do begin
  num := i;
  for j:= P downto 1 do
    num := num*i;
  sum := sum + num
end;
```

```
d)
sum := 1;
for i:= N downto 2 do begin
  num := i;
  for j:= P downto 2 do
    num := num*i;
  sum := sum + num
end;
```

```
e)
sum := 1;
for i:= 1 to N do begin
  num := i;
  for j:= 1 to P do
    num := num*i;
  sum := sum + num
end;
```

**Respuesta:** 15

**Ejercicio 16** Se considera el programa:

```
program qq(input,output);
var i, n : integer;
begin
  readln(n);
  i := n - 1;
  while n mod i <> 0 do
    i := i - 1;
  writeln(n div i)
end.
```

Determinar cuál afirmación es verdadera acerca de la salida de este programa, suponiendo que se ingresa un valor mayor que 1:

- a) Es el mayor divisor primo de la entrada n.
- b) Es el menor divisor no primo de la entrada n.
- c) Para algunas entradas será primo y para otras no.
- d) Es el mayor divisor no primo de la entrada n.
- e) Es el menor divisor primo de la entrada n.

**Respuesta:** 16

**Ejercicio 17** Dadas las siguientes declaraciones

```
CONST MAX = ...;
TYPE
  arreglo = array [1..MAX] of integer;
VAR
  a : arreglo;
  i, num : integer;
```

Suponiendo que tanto el arreglo a como la variable num han sido correctamente inicializados, indique cuál de los siguientes fragmentos de código podría producir un error de ejecución:

```
a)
i:= 1;
while (a[i] <> num) and (i <= MAX) do
  i:= i+1;
```

```
b)
i:= 1;
while (a[i] <> num) and (i < MAX) do
  i:= i+1;
```

```
c)
i:= 1;
while (i <= MAX) and (a[i] <> num) do
  i:= i+1;
```

```
d)
i:= 1;
while (i < MAX) and (a[i] <> num) do
  i:= i+1;
```

```
e)
i:= 1;
while (a[i] <> num) and (i <= MAX-1) do
  i:= i+1;
```

**Respuesta:** 17

**Ejercicio 18** Sea el siguiente fragmento de programa:

```
CONST MAX = ...;
TYPE
  arreglo = array [1 .. MAX] of Char;
VAR
  a : arreglo;
  c : char;
  i, acum : integer;
BEGIN
  ...
  read(c);
  acum := 0;
  (* for ... *)
  ...
END.
```

Suponiendo que el arreglo a ha sido correctamente inicializado, indique qué fragmento hay que insertar luego del comentario para que el programa cuente cuántas veces aparece en el arreglo a el carácter ingresado por el usuario.

```
a)
for i := a[1..MAX] do
  if a[i] = c then
    acum := acum + 1;
```

```
b)
for i:= 1..MAX do
  if a = c[i] then
    acum := acum + 1;
```

```
c)
for i := 1 to MAX do
  if a[c] then
    acum := acum + 1;
```

```
d)
for i := MAX downto 1 do
  if a[i] = c then
    acum := acum + 1;
```

```
e)
for i := MAX downto 1 do
  if a[i] = 'c' then
    acum := acum + 1;
```

**Respuesta:** <sup>18</sup>

**Ejercicio 19** Dado el siguiente programa:

```
program p;
type digitoOctal = 0..7;
var digOct : digitoOctal;
begin
  read(digOct);
  if (digOct < 0) or (digOct > 7) then
    writeln('dato fuera de rango')
  else
    writeln('ok')
end.
```

Indique cuál opción es válida:

- a) El código compila y se ejecuta sin dar errores.
- b) El código compila correctamente pero en la ejecución, al momento de evaluarse la condición del `if`, da error de rango.
- c) El código compila correctamente pero en la ejecución, al momento de la lectura del dato, siempre da error de rango.
- d) El código no compila.
- e) El código compila correctamente pero en la ejecución, al momento de la lectura del dato, puede dar un error de rango.

**Respuesta:** <sup>19</sup>

**Ejercicio 20** Indique cuál de las siguientes definiciones de arreglo es correcta:

```
a) array [0..9.8] of integer;
b) array [char] of boolean;
c) array [real] of real;
d) array ['0'..'19'] of 0..19;
e) array ['0'..'9] of 0..9;
```

**Respuesta:** <sup>20</sup>

## Respuestas

<sup>1</sup>d)

<sup>2</sup>e)

<sup>3</sup>a)

<sup>4</sup>d)

<sup>5</sup>d)

<sup>6</sup>b)

<sup>7</sup>c)

<sup>8</sup>c)

<sup>9</sup>e)

<sup>10</sup>a)

<sup>11</sup>d)

<sup>12</sup>a)

<sup>13</sup>c)

<sup>14</sup>a)

<sup>15</sup>d)

<sup>16</sup>e)

<sup>17</sup>a)

<sup>18</sup>d)

<sup>19</sup>e)

<sup>20</sup>b)