

N° Parcial:

Nombre:

CI:

## Primer Parcial. Programación 1

Instituto de Computación  
Curso 2018

### Observaciones:

- El parcial consta de 16 preguntas y **un** ejercicio de programación.
- Cada pregunta tiene una única opción correcta.
- Una respuesta correcta suma 2 puntos. Una respuesta incorrecta resta 0.5 puntos. Una pregunta sin respuesta no afecta el puntaje.
- El ejercicio de programación tiene un total de 8 puntos.
- Debe entregar la planilla con las respuestas junto con la letra y la resolución del ejercicio. Todo debe estar identificado con nombre y cédula.
- En todos los ejercicios se utiliza el lenguaje Pascal tal como fue dado en el curso (ej. circuito corto, case con else, etc.).

### Ejercicio 1 Dada la siguiente declaración de variables:

```
var k, n : integer;  
    x    : real;  
    c    : char;  
    b1, b2 : boolean;
```

Indique cuál de las siguientes asignaciones **no** es válida:

- $k := k \text{ div } 2 + \text{ord}(c)$
- $x := \text{ord}(c) * n + k$
- $b1 := k \geq \text{round}(x) \text{ and } b2$
- $x := k \text{ mod } 2 + 1$
- $c := \text{chr}(n \text{ div } 3)$

**Respuesta al final:** <sup>1</sup>

### Ejercicio 2 Dado el siguiente fragmento de programa:

```
var n, cont1, cont2 : integer;  
begin  
    cont1 := 0;  
    cont2 := 0;  
    repeat  
        read(n);  
        if n > 0 then  
            cont1 := cont1 + n  
        else  
            cont2 := cont2 + 1;  
        until (cont2 >= 2);  
        writeln(cont1, ' ', cont2)  
    end.
```

Si se ingresan los siguientes valores en la entrada estándar:

2 8 0 5 -2 3 0

Indique cuál será la salida del programa:

- Ninguna, el programa nunca termina
- 16 2
- 16 3
- 15 2
- 15 3

**Respuesta al final:** <sup>2</sup>

**Ejercicio 3** Suponga que  $i$  y  $j$  son variables de tipo `integer`, con valores distintos de cero.

Indique cuál de las siguientes asignaciones es válida:

- $i := \text{sqrt}(i) \text{ div } j$
- $j := j \text{ mod } (2 * 0.1)$
- $i := j / 2 * 2$
- $i := j + j \text{ mod } i$
- $j := \text{sqrt}(j) * \text{sqrt}(j)$

**Respuesta al final:** <sup>3</sup>

**Ejercicio 4** Considere  $k$  de tipo `integer`. Indique cuál de las siguientes expresiones booleanas **no** genera error en tiempo de ejecución para ningún valor de  $k$ :

- $(k > 6) \text{ and } (10 \text{ div } (k-6) = 10 \text{ mod } k)$
- $(k > 6) \text{ or } (10 \text{ div } (k-6) = 10 \text{ mod } k)$
- $(k \geq 6) \text{ and } (10 \text{ div } (k-6) = 10 \text{ mod } k)$
- $(k \geq 6) \text{ or } (10 \text{ div } (k-6) = 10 \text{ mod } k)$
- $(k <> 0) \text{ and } (10 \text{ div } (k-6) = 10 \text{ mod } k)$

**Respuesta al final:** <sup>4</sup>

### Ejercicio 5 Dado el siguiente fragmento de programa:

```
var i, sum : integer;  
begin  
    sum := 0;  
    for i := 1 to 20 do  
        if i mod 2 = 0 then  
            sum := sum + i div 2;  
        write(sum)  
    end.
```

Indique cuál de las siguientes afirmaciones es verdadera:

- El programa despliega la suma de los enteros del 1 al 20.
- El programa despliega la suma de los enteros pares del 1 al 20.
- El programa despliega la suma de los enteros del 1 al 10.
- El programa despliega la suma de los enteros pares del 1 al 10.
- El programa despliega la suma de los enteros impares del 1 al 20.

**Respuesta al final:** <sup>5</sup>

### Ejercicio 6 Dado el siguiente fragmento de programa:

```
var i : integer;  
    a : array [1..5] of integer;  
begin  
    a[1] := 1;  
    i := 1;  
    while (a[i] > 0) and (i < 5) do  
        begin  
            a[i+1] := i + a[i];  
            i := i + 1  
        end;  
        writeln(a[5])  
    end.
```

Indique cuál de las siguientes afirmaciones es verdadera:

- El acceso al arreglo se va de rango.
- El programa despliega el resultado de sumar los números del 1 al 5.
- El programa despliega el número 11.
- El cuerpo del `while` nunca se ejecuta.
- El programa despliega el número 5.

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

**Ejercicio 7** Dado el siguiente fragmento de programa (sin indentar):

```

var b, c : Integer;
begin
readln(b,c);
if b > 1 then
if c < 0 then
write('Uno ')
else
write('Dos ');
write('Fin')
end.

```

Indique cuál será la salida si se ingresan como datos de entrada:

- 0 -2
- Fin
  - Dos Fin
  - Uno Fin
  - Uno Dos Fin
  - Uno

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

**Ejercicio 8** Dado el siguiente fragmento de programa incompleto:

```

const N = ...; (* entero mayor o igual a uno *)
var i, j, num, sum : Integer;
begin
sum := 0;
for i := 1 to N do
begin
num := 1;
(*****)
num := num * i;
sum := sum + num;
end;
writeln(sum)
end.

```

Indique cuál código debe colocarse en lugar de (\*\*\*\*\*) para que su salida sea la siguiente sumatoria:

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

- for j := 1 to N do
- while j <= N do
- for j := i downto 1 do
- while j <= i do
- for j := 1 to 2\*i do

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

**Ejercicio 9** Dado el siguiente fragmento de programa:

```

var
T1, T2: array[1..10] of integer;
i, j : integer;
begin
for i:= 1 to 10 do
T1[i] := i;
for i:= 1 to 10 do
T2[i] := T1[i] * 4 div 2;
for j:= 1 to 20 do
write(T2[j])
end.

```

Indique cuál de las siguientes afirmaciones sobre el programa es verdadera:

- Despliega los números pares entre el 1 y el 20 y termina sin errores.
- Tiene un error un tiempo de ejecución porque el acceso a un arreglo se sale de rango.
- Tiene un error en tiempo de ejecución porque utiliza la misma variable de control en dos iteraciones.
- Lee un arreglo de la entrada, despliega el doble de lo ingresado en cada celda y termina sin errores.
- Carga el arreglo T1, luego duplica sus valores en T2 y termina sin errores.

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

**Ejercicio 10** Indique cuál de las siguientes declaraciones de tipos es correcta:

- t = '5' .. 14
- t = '5' .. '0'
- t = 90 .. maxint
- t = 1.0 .. 5.0
- t = '5' .. '14'

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

**Ejercicio 11** Dado el siguiente fragmento de programa:

```
var i, j, k : Integer;
    a, b : Boolean;
begin
    i := 20;
    j := 30;
    k := 0;
    a := i > j;
    b := not a;
    if a and b then
        writeln('falso');
    if not a and (j > i div k) then
        writeln('verdadero');
end.
```

Indique cuál de las siguientes afirmaciones es verdadera:

- a) El programa termina sin errores y la salida es verdadero.
- b) El programa da error en tiempo de ejecución.
- c) El programa termina sin errores y la salida es falso.
- d) El programa termina sin errores y no despliega nada.
- e) El programa no termina porque queda esperando una entrada.

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

**Ejercicio 12** Dada una variable *a* de tipo array [1 .. N] of integer, donde N es una constante previamente definida. Considerando que todas las variables usadas están definidas correctamente, indique cuál de los siguientes fragmentos de código invierte el arreglo *a*. Por ejemplo, si el arreglo es [2, 4, 1, 7] el resultado es [7, 1, 4, 2].

- a) 

```
for i := 1 to N do
begin
    aux := a[i];
    a[i] := a[N-i+1];
    a[N-i+1] := aux
end
```
- b) 

```
for i := 1 to N div 2 do
    a[i] := a[N-i+1]
```
- c) 

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

```
for i := 1 to N do
    for j := N downto 1 do
        a[i] := a[j]
```
- e) 

```
for i := 1 to N div 2 do
begin
    aux := a[i];
    a[i] := a[N-i+1];
    a[N-i+1] := aux
end
```

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

**Ejercicio 13** Dado el siguiente fragmento de programa:

```
var c1,c2,c3 : char;
begin
    read(c1,c2,c3);
    case c1 of
        'a','c': if (c1<=c2) or (c2<c3) then
            c1 := 'W';
        'b' : if (c3>c1) then
            c2 := 'X'
            else
            c3 := 'Y'
        else
            c3 := 'Z'
    end;
    write(c1,c2,c3)
end.
```

Indique cuál de las siguientes afirmaciones es verdadera:

- a) Si se lee abc se imprime WbZ
- b) Si se lee abc se imprime abc
- c) Si se lee bca se imprime bcZ
- d) Si se lee bca se imprime bcX
- e) Si se lee bca se imprime bcY

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

**Ejercicio 14** Dado el siguiente fragmento de programa:

```
var i, x : integer;
begin
    i := 0;
    read(x);
    repeat
        i := i + 1;
        x := x - 1
    until i > x;
    write(x,'$',i)
end.
```

Indique cuál de las siguientes afirmaciones es verdadera:

- a) Si la entrada es 0 la salida es -1\$1
- b) Si la entrada es 0 el repeat itera infinitamente
- c) Si la entrada es 0 la salida es 0\$0
- d) Si la entrada es -1 la salida es -1\$0
- e) Si la entrada es -1 la salida es 0\$3

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

**Ejercicio 15** Dado el siguiente fragmento de programa:

```
const nada = ' ';
      algo = '*';
var i,j: integer;
begin
    for i:= 5 downto 1 do
        begin
            for j:= 1 to i-1 do
                write(nada);
            for j:= i to 5 do
                write(algo);
            writeln
        end
    end.
```

Indique cuál es su salida:

a) 

```
*****
****
***
**
*
```

b) 

```
*
**
***
****
*****
```

c) 

```
***
**
*
**
***
```

d) 

```
*
**
***
****
*****
```

e) 

```
*****
*****
*****
```

**Respuesta al final:** <sup>15</sup>

**Ejercicio 16** Dado el siguiente fragmento de programa:

```
var i, j : integer;
begin
  i := 1;
  for j := 1 to 5 do
    while i <= 2+j do
      begin
        write(j,i,'-');
        i := i + j
      end
    end
end.
```

Su salida es:

- a) 11-12-13-24-46-
- b) 11-12-13-24-35-46-57-
- c) 11-12-13-21-23-31-34-41-45-51-56-
- d) 11-21-31-41-51-
- e) 03-03-456-

**Respuesta al final:** <sup>16</sup>

## Ejercicio de Resolución

- El puntaje máximo por este ejercicio es **8 puntos** (no resta puntos).
- Este ejercicio debe resolverse en esta hoja.
- En este ejercicio se evaluará, además de la lógica correcta, la utilización de un buen estilo de programación de acuerdo a los criterios impartidos en el curso. De esta manera se tendrá en cuenta entre otros conceptos: indentación apropiada, correcta utilización de las estructuras de control, código elegante y legible, eficiencia de los algoritmos, etcétera.

Escriba un programa en Pascal que lea una secuencia de números naturales y devuelva la posición que ocupa el último número cero en la secuencia de entrada. Para indicar el final de la secuencia se ingresa el número  $-1$ . Si no se ingresa ningún número cero, el programa debe devolver  $-1$ .

Ejemplos:

<b>Entrada:</b> 8 0 9 12 0 -1 <b>Salida:</b> 5	<b>Entrada:</b> -1 <b>Salida:</b> -1
<b>Entrada:</b> 2 0 9 0 10 -1 <b>Salida:</b> 4	<b>Entrada:</b> 8 6 1 2 -1 <b>Salida:</b> -1

**Respuesta:**

## Respuestas

1c

2d

3d

4a

5c

6c

7a

8c

9b

10c

11b

12e

13e

14a

15d

16a