

Nº Parcial:

Nombre:

CI:

Primer Parcial. Programación 1

Instituto de Computación

Septiembre 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 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: ¹

Ejercicio 2 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.
```

Su salida es:

- 11-12-13-24-46-
- 11-12-13-24-35-46-57-
- 11-12-13-21-23-31-34-41-45-51-56-
- 11-21-31-41-51-
- 03-03-456-

Respuesta al final: ²

Ejercicio 3 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: ³

Ejercicio 4 Suponga que i y j son variables de tipo integer, con valores distintos de cero.

Indique cuál de las siguientes asignaciones es válida:

- i := sqrt(i) div j
- j := j mod (2 * 0.1)
- i := j / 2 * 2
- i := j + j mod i
- j := sqrt(j) * sqrt(j)

Respuesta al final: ⁴

Ejercicio 5 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:

- Si la entrada es 0 la salida es -1\$1
- Si la entrada es 0 el repeat itera infinitamente
- Si la entrada es 0 la salida es 0\$0
- Si la entrada es -1 la salida es -1\$0
- Si la entrada es -1 la salida es 0\$3

Respuesta al final: ⁵

Ejercicio 6 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: 6

Ejercicio 7 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:

- a) `k := k div 2 + ord(c)`
 b) `x := ord(c) * n + k`
 c) `b1 := k >= round(x) and b2`
 d) `x := k mod 2 + 1`
 e) `c := chr (n div 3)`

Respuesta al final: 7

Ejercicio 8 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:

- a) Ninguna, el programa nunca termina
 b) 16 2
 c) 16 3
 d) 15 2
 e) 15 3

Respuesta al final: 8

Ejercicio 9 Indique cuál de las siguientes declaraciones de tipos es correcta:

- a) `t = '5' .. 14`
 b) `t = '5' .. '0'`
 c) `t = 90 .. maxint`
 d) `t = 1.0 .. 5.0`
 e) `t = '5' .. '14'`

Respuesta al final: 9

Ejercicio 10 Considere `k` de tipo `integer`. Indique cuál de las siguientes expresiones booleanas **no** genera error en tiempo de ejecución para ningún valor de `k`:

- a) `(k > 6) and (10 div (k-6) = 10 mod k)`
 b) `(k > 6) or (10 div (k-6) = 10 mod k)`
 c) `(k >= 6) and (10 div (k-6) = 10 mod k)`
 d) `(k >= 6) or (10 div (k-6) = 10 mod k)`
 e) `(k <> 0) and (10 div (k-6) = 10 mod k)`

Respuesta al final: 10

Ejercicio 11 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 cual de las siguientes afirmaciones es verdadera:

- a) El programa despliega la suma de los enteros del 1 al 20.
 b) El programa despliega la suma de los enteros pares del 1 al 20.
 c) El programa despliega la suma de los enteros del 1 al 10.
 d) El programa despliega la suma de los enteros pares del 1 al 10.
 e) El programa despliega la suma de los enteros impares del 1 al 20.

Respuesta al final: 11

Ejercicio 12 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:

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

- a) `for j := 1 to N do`
 b) `while j <= N do`
 c) `for j := i downto 1 do`
 d) `while j <= i do`
 e) `for j := 1 to 2*i do`

Respuesta al final: 12

Ejercicio 13 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: ¹³

Ejercicio 14 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: ¹⁴

Ejercicio 15 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:

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

Respuesta al final: ¹⁵

Ejercicio 16 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:

- Si se lee abc se imprime WbZ
- Si se lee abc se imprime abc
- Si se lee bca se imprime bcZ
- Si se lee bca se imprime bcX
- Si se lee bca se imprime bcY

Respuesta al final: ¹⁶

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:

Entrada: 8 0 9 12 0 -1 Salida: 5	Entrada: -1 Salida: -1
Entrada: 2 0 9 0 10 -1 Salida: 4	Entrada: 8 6 1 2 -1 Salida: -1

Respuesta:

Respuestas

1c

2a

3b

4d

5a

6e

7c

8d

9c

10a

11c

12c

13a

14d

15b

16e