

Nº Parcial:  
Nombre:  
CI:

# Primer Parcial. Programación 1

## Instituto de Computación

### Año 2023

#### Observaciones:

- El parcial consta de 14 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 12 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** Sea el siguiente fragmento de código donde x es de tipo integer:

```
x := 0;  
if (x>0) and (10 mod x > 11) then  
  write('IF1')  
else  
  write('ELSE1');
```

- El programa da error de compilación.
- El programa da error de ejecución.
- El programa escribirá IF1.
- El programa escribirá ELSE1.
- El programa no da error pero no despliega nada.

**Ejercicio 2** Suponga que i y j son variables de tipo integer, con valores distintos de cero.

Indique cuál de las siguientes asignaciones es válida:

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

**Ejercicio 3** Dado el siguiente fragmento de programa:

```
var i, sum : integer;  
begin  
  sum := 0; i := 1;  
  while (i <= 20) and (sum <= 222) do  
    begin  
      if i mod 2 = 0 then  
        sum := sum + i div 2;  
      i := i + 1  
    end;  
    write(sum)  
  end.
```

Indique cual 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.

**Ejercicio 4** Dado el siguiente programa, decir qué despliega.

```
program ppp;  
const N = 3;  
var i, j : integer;  
begin  
  i := 2;  
  while i <> N do  
    begin  
      for j := N to N + i do  
        begin  
          write('*');  
          i := i + 1  
        end;  
      write('$');  
      i := i - 2  
    end  
  end.  
end.
```

- \$\$\$
- \*\$\$
- \*\*\*
- \*\*\$\*\$
- \*\$\$

**Ejercicio 5** 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:

- 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-

**Ejercicio 6** Dado el siguiente código, donde N es una constante mayor que 0 y las variables son de tipo integer:

```
cont := 0;  
repeat  
  read(a);  
  if (a = 8) or (cont > N) then  
    cont := 0  
  else cont := cont + 1  
until (cont >= N) and (a = 8);  
write(a)
```

Indique cuál afirmación es correcta, asumiendo que siempre se ingresan enteros válidos:

- Se despliega el último valor leído
- La ejecución del ciclo no termina nunca
- El cuerpo del repeat no se ejecuta nunca
- Se despliega el último valor leído distinto de 8
- El cuerpo del repeat se ejecuta N veces

**Ejercicio 7** Dado el siguiente fragmento de código:

```
for i := 0 to n do  
  write('*');
```

Indique cuál de las siguientes afirmaciones es verdadera:

- Si  $n \geq 0$ , se despliegan n asteriscos
- Si  $n < 0$ , se produce un error en tiempo de ejecución
- Si  $n < 0$ , la ejecución del ciclo no termina nunca
- Si  $n = 0$ , no se despliega nada y no se produce ningún error
- Si  $n < 0$ , no se despliega nada y no se produce ningún error

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

```
procedure externo(a : boolean);
  procedure interno1(b : integer);
    procedure masInterno1(c : char);
      begin
        (* cuerpo masInterno1 *)
      end;
  begin
    (* cuerpo interno1 *)
  end;
  procedure interno2(d : real);
    procedure masInterno2(e : boolean);
      procedure recontraInterno(f : integer);
        begin
          (* cuerpo de recontraInterno *)
        end;
      begin
        (* cuerpo masInterno2 *)
      end;
    end;
  begin
    (* cuerpo interno2 *)
  end;
begin
  (* cuerpo externo *)
end;
```

Indique cuál subprograma **no** puede ser invocado desde el cuerpo de masInterno2

- a) externo
- b) interno1
- c) masinterno1
- d) recontrainterno
- e) interno2

**Ejercicio 9** 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.

**Ejercicio 10** Indique la expresión que toma el valor false

- a) ord('9') + ord('a') = ord('8') + ord('b')
- b) ord('7') - ord('5') <= 3
- c) chr(ord('a')) >= chr(ord('b'))
- d) ord('a') <= ord(chr(ord('b')))
- e) ord('2') + 2 = ord('4')

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

```
...
suma := 0;
i := 2;
for i := 1 to 3 do suma := suma + i;
...
```

Indique los valores de i y suma al terminar la iteración.

- a) 4 y 6
- b) 5 y 6
- c) no está definido y 6
- d) no está definido y 5
- e) 2 y 5

**Ejercicio 12** Dado el siguiente encabezado de un procedimiento:

```
procedure proc (var x : real; u : boolean);
```

Indique cuál de las invocaciones es correcta de acuerdo con la siguiente declaración de variables:

```
var a,b : real; i : integer; p : boolean;
```

- a) proc (sqrt(i), p)
- b) proc (b, i)
- c) proc (i, true)
- d) proc (trunc(a), not p)
- e) proc (a, (i = b) or p)

**Ejercicio 13** Se considera como invierno al período comprendido desde el 21/6 hasta el 20/9, ambos incluidos.

¿Cuál de los siguientes códigos es incorrecto?

```
a) function enInvierno (dia, mes: integer): boolean;
begin
  enInvierno := (6 <= mes) and (mes <= 8) and
    (dia >= 21) or (7 <= mes) and
    (mes <= 9) and (dia < 21)
end;
```

```
b) function enInvierno (dia, mes: integer): boolean;
begin
  if dia >= 21 then
    enInvierno := (7 <= mes) and (mes <= 9)
  else
    enInvierno := (6 <= mes) and (mes <= 8)
end;
```

```
c) function enInvierno (dia, mes: integer): boolean;
begin
  case mes of
    6: enInvierno := dia >= 21;
    7,8: enInvierno := true;
    9: enInvierno := dia < 21
  else enInvierno := false
  end
end;
```

```
d) function enInvierno (dia, mes: integer): boolean;
begin
  case mes of
    6: enInvierno := dia >= 21;
    9: enInvierno := dia < 21
  else enInvierno := (mes = 7) or (mes = 8)
  end
end;
```

**Ejercicio 14** Suponiendo que p, q y r son variables booleanas que fueron inicializadas, la siguiente expresión

```
p or not r and q
```

al ser evaluada, siempre produce como resultado el mismo valor que la expresión:

- a) ((p or (not r)) and q)
- b) (p or (not (r and q)))
- c) (p or ((not r) and q))
- d) (p or ((not r) and (not q)))
- e) (p or (not (q and r)))

## Ejercicio de Resolución

- El puntaje máximo por este ejercicio es **12 puntos** (no resta puntos).
- 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.

a) Escribir un subprograma:

```
function esPotencia(k,N : integer) : boolean;
```

que recibe enteros  $k$  y  $N$  tales que  $k > 1$  y  $N > 0$  y retorna **true** si y sólo si  $N$  es una potencia exacta de  $k$ .

**Ejemplos:**

k	N	esPotencia
3	1	true
2	32	true
3	14	false
25	5	false
3	81	true

No se pueden usar las funciones  $\log$ ,  $\ln$  y  $\exp$  de Pascal.

b) Escribir un subprograma:

```
function cantidad (k, m, n : integer): integer;
```

que recibe tres enteros  $k$ ,  $m$ ,  $n$  tales que  $k > 1$  y  $m > 0$  y retorna la cantidad de enteros mayores o iguales a  $m$  y menores o iguales a  $n$  que son potencia de  $k$ .

Se debe usar el subprograma de la parte a).

**Ejemplos:**

k	m	n	cantidad
3	8	27	2
3	10	26	0
25	5	27	1
2	2	16	4
3	0	4	2
3	10	5	0