

Nº Parcial:  
Nombre:  
CI:

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

## Primer Parcial. Programación 1

Instituto de Computación  
Curso 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 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-

### Ejercicio 2 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;  
begin  
  (* cuerpo interno2 *)  
end;  
begin  
  (* cuerpo externo *)  
end;
```

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

### Ejercicio 3 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 4 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;  
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 5** Suponga que i y j son variables de tipo integer, con valores distintos de cero.

Indique cuál de las siguientes asignaciones es válida:

- a) `i := sqrt(i) div j`
- b) `j := j mod (2 * 0.1)`
- c) `i := j / 2 * 2`
- d) `i := j + j mod i`
- e) `j := sqrt(j) * sqrt(j)`

**Ejercicio 6** 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:

- 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.

**Ejercicio 7** 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.
```

- a) `***$`
- b) `*$$$`
- c) `***$`
- d) `**$*$`
- e) `$*$$$`

**Ejercicio 8** 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');
```

- a) El programa da error de compilación.
- b) El programa da error de ejecución.
- c) El programa escribirá IF1.
- d) El programa escribirá ELSE1.
- e) El programa no da error pero no despliega nada.

**Ejercicio 9** 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

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

**Ejercicio 10** 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:

- a) Se despliega el último valor leído
- b) La ejecución del ciclo no termina nunca
- c) El cuerpo del repeat no se ejecuta nunca
- d) Se despliega el último valor leído distinto de 8
- e) El cuerpo del repeat se ejecuta  $N$  veces

**Ejercicio 11** 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)  $((p \text{ or } (\text{not } r)) \text{ and } q)$
- b)  $(p \text{ or } (\text{not } (r \text{ and } q)))$
- c)  $(p \text{ or } ((\text{not } r) \text{ and } q))$
- d)  $(p \text{ or } ((\text{not } r) \text{ and } (\text{not } q)))$
- e)  $(p \text{ or } (\text{not } (q \text{ and } r)))$

**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** Dado el siguiente fragmento de código:

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

Indique cuál de las siguientes afirmaciones es verdadera:

## 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.

**Solución con multiplicaciones:**

```
function esPotencia(k, N : integer) : boolean;
var potencia : integer;
begin
  potencia:= 1;
  while potencia < N do
    potencia:= potencia * k;
  esPotencia:= potencia = N
end;
```

**Solución con divisiones:**

```
function esPotencia(k, N : integer) : boolean;
var cociente : integer;
begin
  cociente:= N;
  while cociente mod k = 0 do
    cociente:= cociente div k;
  esPotencia:= cociente = 1
end;
```

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	1	4	2
3	10	5	0

**Solución:**

```
function cantidad (k, m, n : integer) : integer;
var
  i      : integer;
  contador : integer;
begin
  contador := 0;
  for i := m to n do
    if esPotencia(k,i) then contador := contador + 1;
  cantidad := contador
end;
```

## Respuestas

1a

2c

3b

4b

5d

6c

7c

8d

9c

10b)

11c

12e

13e)

14c