

Nº de Prueba:

Nombre:

CI:

Casilla de Control: 5

## Programación 1 . Primer Parcial 2024

### Instituto de Computación

**Ejercicio 1** Sean  $a$  y  $b$  variables de tipo `integer`, y `fin` de tipo `boolean`. Considere el siguiente fragmento de código:

```
read(a);
read(b);
fin := false;
while not fin and (a div b > 0) do
begin
    read(b);
    if b = 0 then
        fin := true
    end;
write(b)
```

Asuma que siempre se ingresan valores enteros. Indique cuál de las siguientes afirmaciones es **verdadera**.

- A) El código despliega el último valor leído de la entrada en todos los casos.
- B) El código da error en tiempo de ejecución siempre.
- C) El código puede dar error en tiempo de ejecución en algún caso.
- D) El código despliega el penúltimo valor leído de la entrada en todos los casos.
- E) El código queda en loop (la iteración no termina nunca) en todos los casos.

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

```
cont:=0;
i := 4;
for i := 1 to 6 do
    if i mod 2 = 1 then
        cont := cont + 1;

write (cont)
```

Indique cuál de las siguientes afirmaciones es **verdadera**:

- A) Se imprime 1
- B) Se imprime 2
- C) Se imprime 3
- D) Se imprime 4
- E) Se produce un error en tiempo de ejecución.

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

```
var n, i, j: integer;
begin
    read(n);
    for i := 1 to n do
        for j:= 1 to n do
            if (i+j = n+1) then
                write('(',i,',',j,')')
        end.
end.
```

Si la entrada es 3, indique qué se imprime en la salida:

- A) (1, 2) (1, 3) (2, 3)
- B) (2, 1) (3, 1) (3, 2)
- C) (1, 1) (2, 2) (3, 3)
- D) (1, 3) (2, 2) (3, 1)
- E) (3, 1) (2, 2) (1, 3)

**Ejercicio 4** Considere  $k$  de tipo `integer`. Indique cuál de las siguientes expresiones booleanas nunca genera error en tiempo de ejecución:

- A)  $(k = 0)$  or  $(10 \text{ div } k = 10 \text{ div } (10-k))$
- B)  $(k = 0)$  and  $(10 \text{ div } k = 10 \text{ div } (10-k))$
- C)  $(10 \text{ div } k = 10 \text{ div } (10-k))$
- D)  $(k \text{ mod } 10 = 0)$  or  $(10 \text{ div } k = 10 \text{ div } (10-k))$
- E)  $(k \text{ mod } 10 = 0)$  and  $(10 \text{ div } k = 10 \text{ div } (10-k))$

**Ejercicio 5** Dado el siguiente fragmento de código donde  $a$ ,  $b$  y  $c$  son de tipo `integer`:

```
read(a,b,c);
if a > b then
    if a > c then
        write(a)
    else
        write(c)
else
    if b > c then
        write(b)
    else
        write(c)
```

Indique cuál de las siguientes afirmaciones es **verdadera**:

- A) Se imprime  $a$  cuando  $a > b$  y  $a < c$ .
- B) Se imprime  $c$  cuando  $a < b$  y  $b > c$ .
- C) Se imprime  $b$  cuando  $a > b$  y  $b < c$ .
- D) Se imprime el máximo entre  $a$ ,  $b$  y  $c$ .
- E) Se imprime el mínimo entre  $a$ ,  $b$  y  $c$ .

**Ejercicio 6** Indique cuál de los siguientes fragmentos de código **no** es equivalente a los demás si num es de tipo integer:

- A) 

```
if num = 1 then
  writeln('A')
else if num = 2 then
  writeln('B')
else if num = 3 then
  writeln('C')
```
- B) 

```
case num of
  1 : writeln('A');
  2 : writeln('B');
  3 : writeln('C')
end
```
- C) 

```
if num = 1 then writeln('A');
if num = 2 then writeln('B');
if num = 3 then writeln('C')
```
- D) 

```
if num = 1 then
  writeln('A')
else if num = 2 then
  writeln('B');
if num = 3 then writeln('C')
```
- E) 

```
if num = 1 then
  writeln('A')
else if num = 2 then
  writeln('B')
else writeln('C')
```

**Ejercicio 7** Dado el siguiente encabezado de un procedimiento:

```
procedure miproc (var x : real; y : integer);
```

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

```
var a,b : real; c : integer;
```

- A) miproc (2.7, c)  
 B) a := miproc (b, 8)  
 C) miproc (a + b, 6)  
 D) c := miproc (a, c)  
 E) miproc (a, 9)

**Ejercicio 8** Dadas las siguientes declaraciones de variables:

```
var a : integer; x : real; c : char;
```

Indique cuál de las siguientes asignaciones da error de compilación:

- A) a := a + ord (c)  
 B) x := a + sqr (a)  
 C) c := chr(x \* a)  
 D) c := chr(a + round(x))  
 E) x := ord(c) + ord('c')

**Ejercicio 9** Considere el siguiente programa:

```
program iter;
var a, b: integer;
begin
  readln (a,b);
  repeat
    while not (a < 0) do
      begin
        a := a + b;
        write ('Hi', ' ')
      end;
    b := b - 2 * a;
    write ('Low', ' ')
  until b >= 0;
  write (a * b)
end.
```

Si la entrada es 23 -6, indique qué imprime el programa:

- A) Hi Hi Hi Hi Low Low Low 0  
 B) Hi Hi Hi Low Low Low 16  
 C) Hi Hi Hi Hi Low Low 2  
 D) Hi Hi Hi Hi Hi Low Low -12  
 E) Hi Low Low Low 24

**Ejercicio 10** Sea el siguiente fragmento de código donde a y b son de tipo integer:

```
read (a,b);
if (a < b) and (a > b div 2) then
  write ('A')
else
  if a > (b + 2) then
    write ('B');
  if a mod 2 = b mod 2 then
    write ('C')
```

Indique con cuál de las siguientes entradas se imprime AC en salida:

- A) 4 7  
 B) 6 4  
 C) 4 6  
 D) 5 5  
 E) 4 8

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

```
while (j <> 0) do
begin
  for i := 1 to j mod 10 do write('o');
  write('-');
  j := j div 10
end;
```

Si el valor de j es 135 antes de ejecutar el fragmento, indique cuál de las siguientes afirmaciones es **verdadera**:

- A) Se imprime ooooo-ooo-o-  
 B) Se imprime o-ooo-ooooo-  
 C) Se imprime ooooo-ooo-  
 D) Se imprime o-ooo-  
 E) Se imprime ooooo-

**Ejercicio 12** Dado el siguiente esquema de programa:

```
<inicio>
repeat
  <cuerp01>
until <cond1>;
while <cond2> do
  <cuerp02>
```

Indique cuál de las siguientes afirmaciones es **falsa**, asumiendo que nunca ocurre un error en tiempo de ejecución:

- A) <cuerp01> se va a ejecutar por lo menos una vez.
- B) Si después de una ejecución de <cuerp01> no se modificó alguna de las variables de <cond1> el **repeat** vuelve a iterar.
- C) La iteración del **repeat** termina cuando <cond1> evalúa a true.
- D) Si después de una ejecución de <cuerp02> no se modificó alguna de las variables de <cond2> el **while** vuelve a iterar.
- E) Puede suceder que nunca se ejecute <cuerp02>.

**Ejercicio 13** Considere el siguiente programa:

```
program Ejercicio;
var a, b : integer;

procedure procl (x, y: integer);
  function func2 (x, y: integer): integer;
  begin
    ...
  end;
begin
  ...
end;

function func1(x, y: integer): integer;
  procedure proc2 (a : integer);
  begin
    ...
  end;
  procedure proc3 (b: integer);
  begin
    ...
  end;
begin
  ...
end;

begin
end.
```

Indique cuál de las siguientes afirmaciones es **verdadera**:

- A) La variable global a es visible en todo el código.
- B) La variable global a es visible en todo el código menos en func1.
- C) Es posible invocar func2 desde func1.
- D) Es posible invocar proc2 desde procl.
- E) La variable global b es visible en todo el código menos en proc3.

**Ejercicio 14** Considere el siguiente programa:

```
program prog;
var a, b : Integer;
function f(a, b : Integer): Integer;
begin
  f := a + b
end;
procedure p(var a : Integer);
var b : Integer;
begin
  b := a;
  a := f(a,b);
  write(a, ' ', b, ' ')
end;
begin
  a := 3;
  b := a * 2;
  p(b);
  write(a, ' ', b)
end.
```

Indique qué se imprime en pantalla:

- A) 12 6 3 12
- B) 12 6 12 6
- C) 3 6 3 6
- D) 12 6 12 12
- E) 6 3 6 3

## 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, pasaje correcto de parámetros, etcétera.

Recuerde del laboratorio el procedimiento `leerPalabraLargo`, con el siguiente encabezado:

```
procedure leerPalabraLargo ( var largo : integer; var fin : boolean);
```

`leerPalabraLargo` lee de la entrada estándar una palabra, es decir, una cadena de caracteres terminada en SEPARADOR o FINALIZADOR. `leerPalabraLargo` retorna en el parámetro de salida `largo` el largo de la palabra (sin contar al SEPARADOR o FINALIZADOR) y en el parámetro de salida `fin` un booleano que indica si el último carácter es FINALIZADOR o no. Asumimos que el largo de la palabra es mayor o igual que uno.

- a) Programe un procedimiento `cantidadDeLetras`, que lee de la entrada estándar una oración y retorna en el parámetro de salida `cnt` la cantidad de letras que esta contiene. La oración está compuesta por palabras separadas por un único carácter SEPARADOR, y finaliza con el carácter FINALIZADOR. Los SEPARADORES y el FINALIZADOR no deben contarse. Asumir que la oración tiene al menos una palabra. Para implementar este procedimiento debe invocar al procedimiento `leerPalabraLargo` (que no se debe implementar), y no debe usar los procedimientos `read` ni `readln`.

```
procedure cantidadDeLetras (var cnt: integer);
```

- b) Programe un procedimiento `oracionMasLarga`, que lee de la entrada estándar `n` oraciones, con  $n > 0$ , y devuelve el largo de la oración más larga (la que tiene más letras). Todas las oraciones se ingresan en la misma línea y cada oración tiene el formato de la parte a), en particular, no hay separadores entre cada finalizador y la primera palabra de la siguiente oración. Para implementar este procedimiento debe invocar al procedimiento de la parte a) y no debe usar los procedimientos `read` ni `readln`.

```
procedure oracionMasLarga (n : Integer; var largo: integer);
```

Ejemplos:

Para  $n = 1$ , con la entrada:

Hoy llueve.

se devuelve en `largo` el valor 9.

Para  $n = 3$ , con la entrada:

La casa es roja.El auto esta roto.Hoy llueve.

se devuelve en `largo` el valor 14.

Para  $n = 5$ , con la entrada

Caramba.Llueve.Hace calor.Acaso hubo buhos aca.Hoy es jueves.

se devuelve en `largo` el valor 17

**Solución:**

```
a) procedure CantidadDeLetras (var cnt: integer);
var largo : integer;
    fin : boolean;
begin
    cnt := 0;
    repeat
        leerPalabraLargo (largo, fin);
        cnt := cnt + largo
    until fin
end
```

```
b) procedure oracionMasLarga (n: Integer; var largo: Integer);
var i, cant: integer;
begin
    cantidadDeLetras(largo);
    for i := 2 to n do
    begin
        cantidadDeLetras(cant);
        if cant > largo then
            largo := cant
    end
end;
```

## Respuestas MO

1c

2c

3d

4d

5d

6e

7e

8c

9a

10c

11a

12b

13e

14a