

Nº de Prueba:

Nombre:

CI:

Casilla de Control: 4

Programación 1 . Primer Parcial 2024

Instituto de Computación

Ejercicio 1 Dado el siguiente fragmento de programa donde todas las variables son de tipo `integer`:

```
y := 0;
z := 1;
x := -1;
repeat
  read(x);
  write(x,y,z,'|');
  z := y - x;
  y := y + x + 1
until (x = -1) or (y > 10);
write(x,y,z)
```

Indique qué se despliega en pantalla si se ingresan como datos de entrada 2 5 1 -1:

- A) 2 0 1 | 2 3 -2
- B) -1 0 1
- C) 2 0 1 | 5 3 -2 | 5 9 -2
- D) 2 0 1 | 5 3 -2 | 1 9 -2 | 1 11 8
- E) 2 0 1 | 5 3 -2 | 1 9 -2 | -1 11 8 | -1 11 12

Ejercicio 2 Sea el siguiente fragmento de código, donde `c1` y `c2` son variables de tipo `char`:

```
for c1 := 'A' to 'C' do
begin
  write(c1);
  for c2 := c1 to 'B' do
    write(c1,c2)
end
```

Indique cuál es la salida del programa:

- A) AAAABBBBC
- B) ABC
- C) AAABB
- D) AAABBBCCC
- E) ABCAB

Ejercicio 3 Dado el siguiente fragmento de programa, donde `i` es una variable de tipo `integer`:

```
i := 5;
while i <> 1 do
  case i of
    1: i := -i;
    2: i := sqr(i);
    3: i := i - 2;
    4: i := round(sqrt(i)) - 1;
    5: i := i DIV 2;
  end;
```

Determine cuántas veces se ejecuta la instrucción `case`:

- A) 1
- B) 2
- C) 3
- D) 4
- E) 5

Ejercicio 4 Dado el siguiente fragmento de código, donde `a` y `b` son variables de tipo `integer`:

```
if subp1(a, b > 0) then
  subp2(b, a-b)
else
  subp2(a, b-1)
```

Indique cuál de los siguientes pares de encabezados es **correcto**:

- A) `function subp1(var a, b : integer): integer;`
`procedure subp2(var a, b : integer);`
- B) `procedure subp1(a : integer; b : boolean);`
`function subp2(var a, b : integer): integer;`
- C) `function subp1(a : integer; b : boolean): boolean;`
`procedure subp2(var a : integer; b : integer);`
- D) `function subp1(a : integer; b : boolean): boolean;`
`procedure subp2(var a, b : integer);`
- E) `function subp1(a : integer; b : integer): boolean;`
`procedure subp2(a, b : integer);`

Ejercicio 5 Dada la siguiente declaración de variables:

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

Indique cuál de las siguientes instrucciones de asignación es **válida**:

- A) `b := a / 3`
- B) `a := b div c`
- C) `c := a + b + c`
- D) `a := a div b`
- E) `b := a div 3`

Ejercicio 6 Dadas las siguientes declaraciones de variables:

```
var x, z : integer; w, y : real;
```

y el siguiente encabezado de subprograma:

```
procedure p (x: integer; y : real; var z : real);
```

Indique cuál es una invocación **válida**:

- A) p(x, y, z)
- B) p(x, w, round(w) + 0.5)
- C) p(trunc(y), trunc(w), y)
- D) p(1, 2.0, 3.0)
- E) p(z, x, x)

Ejercicio 7 Sea el siguiente fragmento de código:

```
var p: integer;
begin
  p := 3;
  repeat
    p := p + 1
  until p = 3;

  writeln('El valor de p es: ',p)
end.
```

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

- A) El programa despliega El valor de p es: 3
- B) El programa despliega El valor de p es: 4
- C) El programa no termina
- D) El programa da error en tiempo de compilación
- E) El programa despliega El valor de p es: 6

Ejercicio 8 Dado el siguiente fragmento de programa:

```
cont := 0;
c := '9';
while ord(c) - ord('0') > 6 do
begin
  cont := succ(cont);
  c := pred(pred(c))
end
```

Determine el valor de **cont** al finalizar la iteración:

- A) 1
- B) 2
- C) 3
- D) 5
- E) 7

Ejercicio 9 Determine cuál de las siguientes declaraciones de constantes y variables es **válida**:

- A) const c1 = 'A'; c2 = 'B'; var c1, c2 : char;
- B) CONST num1 = 9; VAR num2, num3 : real, integer;
- C) const pi; VAR r : real;
- D) CONST a = 'B'; var b : integer; c : integer;
- E) const MAX = 10; var min = MAX;

Ejercicio 10 Dado el siguiente fragmento de programa:

```
if chr(ord('E') - 1) = 'F' then
  write('Si - ', ord('5')-ord('0'))
else
  write('No - ', chr(ord('R')))
```

Determine su salida:

- A) Si - 5
- B) Si - 0
- C) Si - 5-0
- D) No - R
- E) No - 'R'

Ejercicio 11 Sean e1 y e2 dos expresiones booleanas, instr1 e instr2 dos instrucciones, y el siguiente fragmento de código:

```
if e1 then
if e2 then
  instr1
else
  instr2
```

Determinar cuál afirmación es **verdadera**:

- A) Si e1=TRUE y e2=TRUE, se ejecuta instr2
- B) Si e1=TRUE y e2=FALSE, se ejecuta instr2
- C) Si e1=FALSE y e2=FALSE, se ejecuta instr2
- D) Si e1=FALSE y e2=TRUE, se ejecuta instr1
- E) Si e1=FALSE y e2=TRUE, se ejecuta instr2

Ejercicio 12 Dado el siguiente programa:

```
program prog;
var p: real;
    a: integer;
    c: char;
begin
  read(a);
  p := a * (ord('D') - ord('A')) / a;
  c := chr(ord('3'));
  if p = c then
    writeln('son iguales')
  else
    writeln('no son iguales')
end.
```

Determine cuál de las afirmaciones es **correcta**:

- A) El programa da error en tiempo de compilación debido a la instrucción `c := chr(ord('3'))`
- B) El valor de la variable `c` es 51
- C) El programa da error en tiempo de compilación debido a la condición del `if: p = c`
- D) Para cualquier valor de `a <> 0` el programa despliega son iguales
- E) para cualquier valor de `a <> 0` el programa despliega no son iguales

Ejercicio 13 Dado el siguiente fragmento de programa:

```
y := 3;
repeat
  x := 0;
  while x <= y do
  begin
    write('o');
    x := x + 1;
    y := y - 1
  end;
  write('$')
until y < 0
```

Indique qué se despliega en la salida:

- A) oo\$
- B) oo\$\$
- C) oo\$o\$o\$
- D) ooo\$\$
- E) oo\$oo\$

Ejercicio 14 Dado el siguiente fragmento de código:

```
var a: boolean;
    num: integer;
begin
  read(num);
  a := num > 1;
  if a then
  begin
    if a or (num < 1) then
      write('a');
    a := not a;
    write('b')
  end;
  if not a then
    write('c')
end.
```

Indique cuál será la salida si se ingresa como dato de entrada 2:

- A) abc
- B) bc
- C) ab
- D) b
- E) c

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 el procedimiento `procesarMensaje`, realizado en el laboratorio, con el siguiente encabezado:

```
procedure procesarMensaje (clave: integer; accion : char);
```

Este procedimiento recibe dos parámetros, una `clave` que será usada en el algoritmo de cifrado César, y una `accion`, que puede ser 'C' para cifrar o 'D' para descifrar. El procedimiento lee una oración desde la entrada estándar, la cual termina con el carácter FINALIZADOR, procesa cada carácter mediante la función `sustituirLetra` y despliega en la salida estándar el mensaje procesado. El procedimiento finaliza con un `readln` luego de leer el FINALIZADOR.

Escriba un procedimiento llamado `procesarTexto`, con el siguiente encabezado:

```
procedure procesarTexto (n, clave: integer; accion : char);
```

Asumiendo que `MAX` es una constante definida, el procedimiento debe leer de la entrada estándar `MAX` oraciones y desplegarlas teniendo en cuenta que a las primeras `n` oraciones deberá aplicarle, antes de desplegarlas, la `accion` utilizando la `clave`. Al resto de las oraciones deberá desplegarlas sin modificaciones.

Tener en cuenta que si `n` es mayor que `MAX`, el procedimiento finalizará después de procesar las primeras `MAX` oraciones.

Ejemplos de ejecución del procedimiento con `MAX = 3` y `FINALIZADOR = '.'`:

- Ejemplo 1:

Para `n = 2, clave = 1, accion = 'C'`, con la entrada:

```
Hola.  
Hola nuevamente.  
Hola otra vez.
```

El procedimiento desplegará:

```
El mensaje cifrado es: Ipmb.  
El mensaje cifrado es: Ipmb ovfwbnfouf.  
El mensaje cifrado es: Hola otra vez.
```

- Ejemplo 2:

Para `n = 5, clave = 1, accion = 'C'`, con la entrada:

```
Hola.  
Hola nuevamente.  
Hola otra vez.
```

El procedimiento desplegará:

```
El mensaje cifrado es: Ipmb.  
El mensaje cifrado es: Ipmb ovfwbnfouf.  
El mensaje cifrado es: Ipmb push wfa.
```

Solución:

```
procedure procesarTexto (n, clave: integer; accion : char);  
var i, j, min : integer;  
begin  
  if n < MAX then  
    min := n  
  else  
    min := MAX;  
  
  for i := 1 to min do  
    procesarMensaje(clave, accion);  
  
  for j := min + 1 to MAX do  
    procesarMensaje(0, accion)  
end;
```

Respuestas MO

¹d

²a

³c

⁴c

⁵b

⁶c

⁷c

⁸b

⁹d

¹⁰d

¹¹b

¹²c

¹³c

¹⁴a