

# CADP 2017

## Práctica 3 - Modularización

1. a. Dado el siguiente programa, indicar qué valores imprime en pantalla.

```

program principal;

procedure respuesta(var a, b: INTEGER);
var
    x, y: integer;
begin
    x:= 3;
    a:= a-b;
    y:= x;
    b:= 2*b-y
end; {fin respuesta}

var
    x, y: integer;

begin {programa principal}
    x:=2;
    y:=5;
    respuesta(x,y);
    WRITELN('X:',x,' Y:',y);
    readln;
end. {fin programa principal}
  
```

b. ¿Qué imprimiría el programa presentado en el inciso “a.” si se cambia el encabezado del procedimiento por: “**procedure** respuesta(a:INTEGER; var b:INTEGER);”?

c. ¿Qué imprimiría el programa presentado en el inciso “a.” si se cambia el encabezado del procedimiento por: “**procedure** respuesta(a:INTEGER; b:INTEGER);”?

2. De acuerdo al siguiente programa:

```

program principal;

procedure intercambiar(var x,y : integer);
var
    aux : integer;
begin
    aux := x;
    x := y;
    y := aux;
end; {fin intercambiar}

function doblePar( x: integer) : integer;
begin
    if (x mod 2 = 0) then
        doblePar := 2*x
    else
        doblePar := x;
end; {fin doblePar}
  
```

```

procedure sumar(var acumulado : integer; limite, resto : integer);
var
    i : integer;
begin
    acumulado := 0;
    for i:= 1 to limite do
        acumulado := acumulado + i;
        acumulado := acumulado - resto;
        writeln(acumulado);
        intercambiar(acumulado,limite);
    end; {fin sumar}

var
    cant1, cant2, total, i: integer;

begin {programa principal}
    read(cant1);
    read(cant2);
    for i :=1 to cant1 do
        begin
            sumar(total, doblePar(i), cant2);
        end;
        writeln('El total es: ',total);
        writeln('Cant1 vale: ',cant1, ' y cant2 vale: ',cant2);
    end. {fin programa principal}

```

- a. Indicar qué imprime en pantalla, si se ingresan los valores: **1 4.**
  - b. Indicar qué imprime en pantalla, si se ingresan los valores: **3 2.**
  - c. Indicar qué imprime en pantalla, si se ingresan los valores: **5 0.**
  - d. Indicar qué imprime en pantalla, si se ingresan los valores: **6 6.**
3. Indicar qué imprime en pantalla el programa, si se lee la siguiente secuencia de números:  
**3 2 4 0 -1.**

```

program principal;

procedure intercambiar(var x,y : integer);
var
    aux : integer;
begin
    aux := x;
    x := y;
    y := aux;
end; {fin intercambiar}

function doblePar( x: integer) : integer;
begin
    if (x mod 2 = 0) then
        doblePar := 2*x
    else
        doblePar := x;
    end; {fin doblePar}

procedure sumar(var acumulado : integer; limite, resto : integer);
var
    i : integer;
begin
    acumulado := 0;
    for i:= 1 to limite do

```

```

        acumulado := acumulado + i;
        acumulado := acumulado - resto;
        writeln(acumulado);
        intercambiar(acumulado, limite);
    end; {fin sumar}

var
    cant1, suma: integer;

begin {programa principal}
    read(cant1);
    suma := 0;
    while (cant1 <> -1) do begin
        sumar(suma, doblePar(cant1), cant1);
        read (cant1);
    end;
end. {fin programa principal}

```

4. Realizar un programa que calcule el factorial para cada uno de los números entre 1 y 50. Informar en pantalla el resultado de cada factorial calculado. Modularizar la solución.
5. Realice un programa que lea desde teclado la base y altura de 100 rectángulos. Informar en pantalla el área ( $base * altura$ ) y el perímetro ( $(base * 2) + (altura * 2)$ ) de cada uno de ellos. Modularizar la solución.
6. Realice un programa que lea desde teclado una secuencia de números enteros. La secuencia finaliza cuando llega el número 0. Al finalizar la lectura informar cuantos números poseen más dígitos pares que impares. Modularizar la solución.
7. Realizar un programa que lea secuencia de números enteros. La lectura finaliza cuando llega el número 123456, el cual debe procesarse. Informar en pantalla para cada número la suma de sus dígitos pares y la suma de sus dígitos impares. Modularizar la solución.
8. Realizar un programa que lea una secuencia de caracteres terminada ".". La secuencia está dividida en palabras separadas por uno o más blancos. Al finalizar la lectura informar en pantalla las longitudes de las dos palabras con menos caracteres. Modularizar la solución.
9. Realizar un programa que lea una secuencia de caracteres terminada ".". La secuencia está dividida en palabras separadas por uno o más blancos. Al finalizar la lectura informar en pantalla las longitudes de las dos palabras más largas que comienzan con "a". Modularizar la solución.
10. Realizar un programa que lea una secuencia de caracteres finalizada en "\*". Determinar si la secuencia cumple con el patrón (en caso de no cumplir, informar que parte del patrón no se cumplió). Modularizar la solución.

**Patrón:**        **A%B\***

**Donde:**        % seguro existe.

**A** → Debe ser una secuencia de caracteres en la que no exista el carácter "#".

**B** → Debe ser una secuencia con la misma cantidad de caracteres que aparecieron en **A** y en la que aparezca a lo sumo 5 veces el carácter "!".

- 11.** Realizar un programa que lea una secuencia de caracteres finalizada en “\$”. Determinar si la secuencia cumple con el patrón (en caso de no cumplir, informar que parte del patrón no se cumplió). Modularizar la solución.

*Patrón:*        **A&B&C\$**

*Donde:*        & seguro existe.

**A →** Debe ser una secuencia de caracteres donde la letra “a” aparezca como mínimo el 85% de los caracteres leídos.

**B →** Debe ser una secuencia de caracteres donde aparezcan sólo vocales o al menos tres veces el carácter “\*”.

**C →** Debe ser una secuencia de caracteres cuya cantidad sea menor a la cantidad de caracteres leídos en **A** y mayor a la cantidad de vocales que aparecieron en **B**.