

CADP 2020 - Turnos Recursantes CADP y ADP

Primera autoevaluación

Fecha: Lunes 11 de mayo de 2020

Hora de inicio: 18.30 hs

Hora de fin: 20:00 hs

Datos del alumno

Apellidos: _____

Nombres: _____

Nro. de Grupo: _____

Pautas generales

La actividad consta de 2 partes: 5 preguntas conceptuales (5 puntos) y un ejercicio práctico (5 puntos). Al tratarse de una autoevaluación, la nota obtenida será sólo para diagnóstico tanto de los alumnos como de la cátedra.

Una vez completada, la actividad debe enviarse a través de la mensajería de IDEAS a los JTP (Eduardo Ibáñez y Gonzalo Villarreal). Pueden enviar un archivo de Word (.doc o .docx) o un archivo de texto plano. Quienes no dispongan de una computadora, pueden escribir las respuestas en papel, y enviar una foto. En este caso, se ruega encarecidamente que el texto sea prolijo y que la foto tenga calidad suficiente para ser leída por otra persona. **No se corregirán actividades cuyas fotos no se puedan leer.**

PRIMERA PARTE - PREGUNTAS CONCEPTUALES

1. Dos alumnos tenían que implementar el módulo “masPatas”, este módulo recibe dos “animales” y devuelve el animal que más patas tiene. Cada alumno hizo la solución, se la mostró a la profesora y ella dijo:
- Ninguna solución es correcta
 - La solución 1 es correcta.
 - La solución 2 es correcta
 - Ambas soluciones son correctas.

Solución 1: Program solucionUNO; Type Animal = record Nombre: string; Raza: string; cantPatas:integer; end ; procedure masPatas (a1:animal; a2:animal; var aMax:animal) //Este módulo devuelve en aMax el animal con más patas Begin If (a1.cantPatas >= a2.cantPatas) then aMax:= a1 else aMax:= a2; end ;	Solución 2: Program solucionDOS; Type Animal = record Nombre: string; Raza: string; cantPatas:integer; end ; function masPatas (a1:animal; a2:animal): animal; //Este módulo devuelve el animal con más patas Var a3: animal; Begin If (a1.cantPatas >= a2.cantPatas) then a3:= a1 else a3:= a2; masPatas:= a3; end ;
---	--

2. Suponga que el programa contiene un módulo de tipo función entonces, considerando sólo la implementación de la función (no el llamado a la misma):
- Siempre va a poder cambiarlo por un módulo que sea procedimiento.
 - Lo va a poder cambiar por un módulo procedimiento sólo cuando la función devuelva un único valor.
 - Lo va a poder cambiar por un módulo procedimiento sólo cuando la función devuelva un integer o un char, o un boolean.
 - Ninguna de las opciones es verdadera.

3. Dado el siguiente módulo, cuál/cuáles de las siguientes invocaciones son correctas:

Program solucionUNO; Type animal = record nombre: string; raza: string; patas:integer;
--

<pre> end; procedure patasPar (x: integer; var par:boolean) //Este módulo devuelve verdadero en el parámetro par si x es par begin par:= (x MOD 2 = 0); end; </pre>	
<pre> OPCION 1 Var a1,a2:animal; par:boolean; Begin a1.nombre:= "Pipo"; a1.raza:= "perro"; a1.patas:= 4; a2.nombre:= "Pepe"; a2.raza:= "pez"; a2.patas:= 0; patasPar (a1.patas+a2.patas,par); End. </pre>	<pre> OPCION 2 Var a1,a2:animal; par:boolean; Begin a1.nombre:= "Pipo"; a1.raza:= "perro"; a1.patas:= 4; a2.nombre:= "Pepe"; a2.raza:= "pez"; a2.patas:= 0; patasPar (a1,par); End. </pre>
<pre> OPCION 3 Var a1,a2:animal; par:boolean; Begin a1.nombre:= "Pipo"; a1.raza:= "perro"; a1.patas:= 4; a2.nombre:= "Pepe"; a2.raza:= "pez"; a2.patas:= 0; patasPar (a1.patas,par); </pre>	<pre> OPCION 4 Var num1,num2:integer; par:boolean; Begin num1:= 8; num2:= 9; patasPar (num1,par); End. </pre>
<pre> OPCION 5 Var num1,num2:integer; par:boolean; Begin num1:= 8; num2:= 9; patasPar (num1+num2,par); End. </pre>	

- Todas las opciones para invocar al módulo funcionan.
- Sólo la OPCION 3 es correcta.
- Sólo la OPCION 4 es correcta.
- Sólo la OPCION 2 es correcta.
- Las opciones OPCION 1, OPCION 3, OPCION 4 y OPCION 5 son correctas.
- Ninguna opción es correcta.

4. De las siguientes afirmaciones marque la más adecuada:

- a. Un procedimiento puede invocar una función.
- b. Un procedimiento puede invocar a otro procedimiento.
- c. Las funciones pueden invocar a un procedimiento.
- d. Todas son falsas.
- e. Todas son verdaderas.

5. Dado el siguiente código, elija la opción más adecuada

```
Program pregunta;  
procedure auxiliar (valor1: integer; var valor2:integer);  
Begin  
    .....  
end;  
Var  
    x,y:integer;  
Begin  
    x:= 18;  
    auxiliar (x,y);  
    write (x);  
End.
```

- a. La instrucción write (x) dará error ya que no se conoce el código del procedimiento.
- b. El programa pregunta dará error ya que la variable y no está inicializada.
- c. El procedimiento auxiliar dará error ya que la variable y no está inicializado.
- d. El write imprimirá basura.
- e. El write imprimirá 18 no importa el código del procedimiento auxiliar.

SEGUNDA PARTE - EJERCICIO PRÁCTICO

Analice el siguiente problema

Realizar un programa que lea datos de los alumnos de primero y segundo año de la Facultad de Informática. De cada alumno se lee el apellido, número de documento, cantidad de cursadas aprobadas, cantidad de finales aprobados y nota promedio. En total son 2400 alumnos.

Una vez finalizada la lectura, el programa debe informar:

- a) la cantidad de alumnos considerados "al día". Un alumno está al día si ha aprobado con examen final al menos el 60% de las cursadas realizadas. Por ejemplo, si un alumno aprobó 8 cursadas y 4 finales, entonces sólo aprobó con final la mitad (50%) de las cursadas, y no se lo considera "al día". Y si otro alumno aprobó 10 cursadas y 7 finales, entonces aprobó con final el 70% de las materias cursadas, con lo cual se lo considera al día.
- b) el promedio general de todos los alumnos
- c) el DNI de los dos mejores promedios

Consigna

Un programador de su equipo ha comenzado a resolver el problema pero no llegó a completarlo. Sin embargo, dejó anotaciones en los 7 lugares donde faltó implementar o revisar. Su trabajo consiste en completar el programa agregando o corrigiendo el código faltante.

Ayudita:

- Los comentarios que indican **COMPLETAR**, se refieren a código que falta completar
- Los comentarios que indican **ARREGLAR**, se refieren a código que podría dar error y que hay que revisar y/o acomodar

```

program alumnos;
const
    cantAlumnos = 2400;
type
    TAlumno = record
        apellido : string;
        dni : integer;
        cursadas : integer;
        finales : integer;
        promedio : real;
    end;

// 1. COMPLETAR: DEJO EL ENCABEZADO, ME FALTA HACER TODO EL MÓDULO
procedure actualizarMaximos(a : TAlumno; var max1, max2 : integer; var prom1,
prom2 : integer);

// 2. ARREGLAR: creo que algo no está bien aquí
procedure leerAlumno(a : alumno);
begin
    read(apellido);
    read(dni);
    read(cursadas);
    read(finales);
    read(promedio);
end;
function promedioGeneral (notas : real) : real;
begin
    promedioGeneral := notas / cantAumnos;
end;

// 3. COMPLETAR: REALIZAR MÓDULO estaAlDia
{ programa principal }
var
    alumno : TAlumno;
    i, alDia: integer;
    sumaNotas, mejorProm,segundoMejorProm : real;

    //4. ARREGLAR ME FALTAN ALGUNAS VARIABLES :-(

begin

    //5. ARREGLAR INICIALIZAR LAS VARIABLES
    for i := 1 to cantAlumnos do
    begin
        leerAlumno(alumno);
        sumaNotas := sumaNotas + alumno.promedio;
        if (estaAlDia(alumno.cursadas, alumno.finales)) then
            alDia := alDia + 1;
        //6. COMPLETAR: AQUÍ HAY QUE UTILIZAR actualizarMaximos
    end; //fin del for
    writeln('El promedio general de todos los alumnos es:
',promedioGeneral(sumaNotas));
    // 7. COMPLETAR: qué más había que informar???
end.

```