PROGRAMACIÓN BÁSICA – Examen
Evaluación Continua – Global
19 de enero de 2022

SOLUCIÓN
PARTE 1

PREGUNTA 1. (1 punto) Especificaciones y casos de prueba (30 min.)

1.1 Especifica el procedimiento *Sugerencias_de_Propina*.

```
procedure Sugerencias_Propinas(A_Pagar: Float; S1, S2, S3: out Float);
-- pre: A_Pagar representa la cantidad a pagar por un cliente.
--      A_Pagar > 0.0
-- post: S1 representa sugerencia de un 10% de A_Pagar de propina a
--       pagar, S2 de un 15% y S3 de un 20%. Es decir:
--       S1=A_Pagar*0.1 ; S2=A_Pagar*0.15; S1=A_Pagar*0.2
```

1.2 Especifica la función *Valor_imc*.

```
function Valor_imc (Altura, Peso: Float) return Character;
-- pre: Altura representa la altura de una persona en cm, Peso
--      representa su peso en Kg. Altura, Peso > 0.0
-- post: devuelve una letra correspondiente al índice de masa corporal:
--      A extremadamente delgado (imc<18,5), B normal (18.5<=imc<=
--      24.9),C para sobrepeso (25<= imc<= 29.9) y D para obesidad
--      (imc>=30.0). Cálculo de imc: imc=peso/altura_m², donde
--      altura_m es el valor de la altura en metros.
```

1.3 Describe 4 casos de pruebas para el subprograma *Valor_imc*.

Nº Caso	Descripción	Datos de entrada	Resultados esperados
1	**A** – Extrema Delgadez (< 18,5)	40kg., 170cm.	**A**
2	**B** – Normal (18.5-24.9)	57kg., 165cm.	**B**
3	**C** – Sobrepeso (25.0-29.9)	75kg., 150cm.	**C**
4	**D** – Obesidad (30.0 o más)	100kg., 170cm.	**D**

PREGUNTA 2. Tipos Básicos (45 min.)

2.1 Implementa en Ada la función *Mas_2_al_reves* **(1.25 punto)**

```
FUNCTION Mas_2_Alreves (N: IN Natural) RETURN Natural IS
   Digitos, Dig, Mas2, Resultado : Natural;
BEGIN
   Mas2 := N+2;
   Digitos := contar_digitos(Mas2);
   Resultado := 0;
   FOR I IN REVERSE 1..Digitos LOOP
       Digito_I(Mas2, I, Dig);
       Resultado := Resultado*10 + Dig;
   END LOOP;
   RETURN Resultado;
END Mas_2_Alreves;
```

2.2 (1.75 puntos)

```
PROCEDURE Secuencia_Mas_2_Alreves IS
   N, Anterior, Mas2alreves, Cont : Integer;
BEGIN
   Put_Line("Escribe números enteros, terminada en un negativo");
   LOOP
      Get(N);
      EXIT WHEN N < 0;
      Cont := 0;
      Anterior := N;
      LOOP
         Mas2alreves := Mas_2_Alreves(Anterior);
         EXIT WHEN Mas2alreves < Anterior;
         Cont := Cont+1;
         Anterior := Mas2alreves;
      END LOOP;
      Put(Cont);
   END LOOP;
END Secuencia_M As_2_Alreves;
```

PARTE 2

PREGUNTA 3. Vectores (t= 45 min.)

3.1. (1.75 puntos)

```
FUNCTION Encaja_Este (SL: IN T_SOPA_LETRAS; P: IN STRING;
                                    F, C: IN Positive) RETURN Boolean IS
   i, c_aux : Natural;
   Encaja : Boolean := True;
BEGIN
   c_aux:= C;
   I := P'first;
   WHILE I <= P'Last AND c_aux<= SL(F)'Last AND Encaja  LOOP
      IF SL(F)(c_aux) /= ' ' AND SL(F)(c_aux) /= P(I) THEN
         Encaja := False;
      ELSE
         I := I+1;
         c_aux:= c_aux+1;
      END IF;
   END LOOP;
   IF I <= P'Last THEN
      Encaja := False;
   END IF;
   RETURN Encaja;
END Encaja_Este;
```

3.2. (0.75 puntos)

```
PROCEDURE Colocar_Sur (SL: IN OUT T_SOPA_LETRAS; P: IN String;
                                                F, C: IN Positive) IS
BEGIN
   FOR I IN P'range LOOP
      SL(F+(I-P'first))(C) := P(I);
   END LOOP;
END Colocar_Sur;
```

PROGRAMACIÓN BÁSICA – Examen
Evaluación Continua – Global
19 de enero de 2022

PREGUNTA 4. (1.5 puntos) Listas estáticas (t= 30 min.)

```
PROCEDURE Insertar_Palabra(VP : IN OUT T_Todas_Palabras; P: IN String) IS
   I    : Natural;
   Num  : Natural          := P'Length;
   L    : T_Lista_Est_Pal  := VP (Num);
   Long : Natural          := L.Cont;
BEGIN
   I := 1;
   WHILE I <= Long AND THEN L.Palabra(I)(P'range)< P LOOP
      I := I+1;
   END LOOP;

   IF (I<= Long AND THEN L.Palabra(I)(P'range)> P) OR I> Long THEN
      VP(Num).Cont := Long + 1;
      VP(Num).Palabra(I+1..VP(Num).Cont) := VP(Num).Palabra(I..Long);
      VP(Num).Palabra(I)(P'range) := P;
   END IF;

END Insertar_Palabra;
```

PARTE 3

PREGUNTA 5. (2 puntos) Listas dinámicas (t=45min)

```
PROCEDURE Modificar_Palabra(VP: IN OUT T_Todas_Palabras; P: String) IS
   Num : Natural := P'Length;
   Ant, Act, Nuevo : T_Lista_Din_Pal := VP (Num);
BEGIN
   WHILE Act /= NULL AND THEN Act.Pal(P'range)< P LOOP
      Ant := Act;
      Act := Act.Sig;
   END LOOP;

   IF Act/=NULL AND THEN Act.Pal(P'range)=P THEN
      IF Ant=Act THEN
         VP(Num):= Act.Sig;
      ELSE
         Ant.Sig:= Act.Sig;
      END IF;
   ELSE --No está la palabra
      Nuevo:= := new t_Nodo_Pal;
      Nuevo.Pal(P'range):= P;
      Nuevo.Sig:= Act;
      IF Ant=Act THEN
         VP(Num):= Nuevo;
      ELSE
         Ant.Sig:= Nuevo;
      END IF;
   END IF;
END Modificar_Palabra;
```