

PROGRAMACIÓN BÁSICA – Examen (Grupo B)
Evaluación Ordinaria – CONTINUA
18 de octubre de 2021

1. EJERCICIOS SENCILLOS: especificación, implementación y casos de prueba.

Programas y subprogramas

- 1.1. (2.5 puntos) Especifica, Implementa y Diseña los casos de prueba** del programa **Digito_Posicion_Mayor**, que dado un número positivo N por la entrada estándar, indica por salida estándar cuál es el valor del dígito mayor y su posición (contada desde la derecha). El número N sólo tiene un **dígito mayor** que es único y distinto a los demás.

Por ejemplo, el dígito mayor y la posición del 6291340 es el 9 (que es único) y posición 5, y del 14625 es el 6 (único) y posición 3.

```
WITH Ada.Integer_Text_IO; USE Ada.Integer_Text_IO;

PROCEDURE Digito_Posicion_Mayor IS
  -- Entrada: Un entero N (EE)
  -- pre: N>0, y N sólo tiene un dígito mayor que los demás dígito
  --      (es único y distinto de los demás)
  -- Salida: Dos Naturales, Dig_Max y Pos_Max (SE)
  -- post: Dig_Max es el valor del dígito mayor de N y Pos_Max la posición en la que está
  --        (contando desde la derecha).

  N, Aux: Integer;
  Cont, Pos_Max: Natural:=0;
  Dig_Max, Dig: Natural:=0;

BEGIN
  Get(N);
  Aux:= N;
  WHILE Aux/=0 and Aux/=9 LOOP
    Dig:= Aux mod 10;
    Cont:= Cont+1;
    IF Dig> Dig_Max THEN
      Dig_Max:= Dig;
      Pos_Max:= Cont;
    END IF;
    Aux:= Aux/10;
  END LOOP;
  Put(Pos_Max); Put(Dig_Max);
END Digito_Posicion_Mayor;
```

Nº Caso	Descripción	Datos de entrada	Resultados esperados
1	Un numero de un sólo digito	9	9 1
2	Un numero de muchos dígitos. El mayor en primera posición.	13425649	9 1
3	Un numero de muchos dígitos. El mayor en posición intermedia.	13492564	9 5
4	Un numero de muchos dígitos. El mayor en última posición intermedia.	91342564	9 8

- 1.2. (1.5 puntos) Especifica e Implementa** el programa **Suma_Digitos**, que dados por teclado un número positivo N y dos valores enteros I y J que representan posiciones, obtiene la suma de los dígitos desde la posición I a la J. $1 \leq I \leq$ cantidad de dígitos de N+1, y $0 \leq J \leq$ cantidad de dígitos de N. Si $I > J$ la suma es 0.

PROGRAMACIÓN BÁSICA – Examen (Grupo B)
Evaluación Ordinaria – CONTINUA
18 de octubre de 2021

Por ejemplo, la suma de dígitos del número 6291340 desde la posición 1 hasta la 4 es $8=0+4+3+1$. Para ese mismo número desde la posición 1 a la 0, será 0. Y desde la posición 6 a la 7 será $8=2+6$.

```
WITH Ada.Integer_Text_IO; USE Ada.Integer_Text_IO;

PROCEDURE Suma_Digitos IS
  -- Entrada: Tres enteros, N, I, J (EE)
  -- pre:  $1 \leq I \leq$  cantidad de dígitos de N,  $0 \leq J \leq$  cantidad de dígitos de N.
  -- Salida: 1 entero Suma (SE)
  -- post: Suma es la suma de los dígitos desde la posición I a la J si  $I \leq J$ 
  --       (contando desde la derecha). Si  $I > J$  entonces Suma=0.

  N, Aux: Integer;
  I, J, Cont: Natural;
  Dig: Natural;
  Suma: Integer:=0;

BEGIN
  Get(N); Get(I); Get(J);
  Aux:= N / 10**(I-1);
  Cont:= I;
  WHILE Cont<= J LOOP
    Dig:= Aux mod 10;
    Suma:= Suma+ Dig;
    Aux:= Aux/10;
    Cont:= Cont+1;
  END LOOP;
  Put(Suma);
END Suma_Digitos;
```

- 1.3. (1 punto) Diseña la cabecera de los subprogramas** (no es necesaria la especificación completa) Dígito_Posicion_Mayor y Suma_Digitos anteriores en los que los datos y resultados se pasen y devuelvan por parámetro y no por dispositivos estándares. Es decir, transformaremos la cabecera de los programas anteriores en subprogramas.

```
PROCEDURE Dígito_Posicion_Mayor(N: Integer; Max, Pos: out Natural);

FUNCTION Suma_Digitos(N : Integer; I, J: Natural) RETURN Integer ;
```

- 1.4. (1 punto) Implementa** el programa Es_Especial, que dado un número por entrada estándar escriba por pantalla si es o no un número especial. Un **número es especial** si la suma de los dígitos de la parte derecha del dígito mayor de N, es igual a la suma de los dígitos de la parte izquierda del dígito mayor.

Por ejemplo, **6291340** es Número Especial ya que $6+2=1+3+4+0$. El número **14625** no es Número Especial ya que $1+4 \neq 2+5$. Pero sí lo es el **400**, $0=0+0$, o el **5**, $0=0$.

NOTA: Para implementar el programa Es_Especial es necesario utilizar los subprogramas cuyas cabeceras se han definido en el apartado 3.

```
WITH Ada.Integer_Text_IO, Ada.Text_IO; USE Ada.Integer_Text_IO, Ada.Text_IO;

WITH Dígito_Posicion_Mayor, Suma_Digitos;

PROCEDURE Es_Especial IS
```

PROGRAMACIÓN BÁSICA – Examen (Grupo B)
Evaluación Ordinaria – CONTINUA
18 de octubre de 2021

```
N, Aux: Integer;  
I, J, Cont: Natural;  
Max, Pos_Max, Dig: Natural;  
Suma_dch: Integer:=0;  
  
BEGIN  
  Get(N);  
  Digito_Posicion_Mayor(N, Max, Pos_Max);  
  Suma_Dch:= Suma_Digitos(N, 1, Pos_Max-1);  
  
  Aux:= N / 10**(Pos_Max);  
  WHILE Aux/=0 and Suma_Dch>0 LOOP  
    Dig:= Aux mod 10;  
    Suma_dch:= Suma_dch- Dig;  
    Aux:= Aux/10;  
  END LOOP;  
  Put(Boolean'Image(Suma_Dch=0 and Aux=0));  
END Es_Especial;
```

2. EJERCICIOS SECUENCIAS: implementación y uso de subprogramas

- 2.1. (1 punto) Implementa en Ada el programa** Secuencia_Dig_Mayor **que** dada por entrada estándar una secuencia de números positivos (sin dígito mayor repetido), escriba por salida estándar la posición y el dígito mayor de cada número de la entrada. Se debe hacer uso en la implementación del programa Secuencia_Dig_Mayor el subprograma Digito_Posicion_Mayor.

Por ejemplo, si la secuencia de entrada es <891340 14625 348124 4531>, se mostraría por pantalla <5 9 3 6 4 8 3 5>.

```
WITH Ada.Integer_Text_IO, Ada.Text_IO; USE Ada.Integer_Text_IO, ADa.Text_IO;  
with Digito_Posicion_Mayor;  
  
PROCEDURE Secuencia_Dig_Mayor IS  
  
  N, Pos, Max: Integer;  
  
BEGIN  
  while not end_of_file loop  
    Get(N);  
    Digito_Posicion_Mayor(N, Max, Pos);  
    Put(Pos); Put(Max);  
  END LOOP;  
END Secuencia_Dig_Mayor;
```

- 2.2. (1 punto) Implementa en Ada el programa** Son_Especiales **que** dada por entrada estándar una secuencia de números positivos terminada en 0, indique si todos los elementos de la secuencia son números especiales. Los números de la entrada no tienen dígito mayor repetido.

*Por ejemplo, si la secuencia de entrada es <891340 14625 348124 4531 0>, se indicará por pantalla **FALSE** ya que el número 14625 no es especial. Si la secuencia fuera <891340 348124 4531 0>, se indicará por pantalla **TRUE**.*

```
WITH Ada.Integer_Text_IO, Ada.Text_IO; USE Ada.Integer_Text_IO, ADa.Text_IO;
```

PROGRAMACIÓN BÁSICA – Examen (Grupo B)
Evaluación Ordinaria – CONTINUA
18 de octubre de 2021

```
WITH Es_Especial;  
  
PROCEDURE Son_Especiales IS  
  
    N: Integer;  
  
BEGIN  
    loop  
        Get(N);  
        EXIT WHEN N=0 OR ELSE not es_especial(N);  
    END LOOP;  
    Put(Boolean'Image(N=0));  
  
END Son_Especiales;
```

- 2.3. (2 puntos) Implementa en Ada el programa** *Secuencia_3_Especiales* que dada por entrada estándar una secuencia de números positivos terminada en 0, escriba por pantalla los tres primeros *números especiales*. Si en la secuencia no hubiera 3 *números especiales*, se escribirán por pantalla tantos -1 como *números especiales* falten.

Por ejemplo, si la secuencia de entrada es <891340 14625 348124 4531 358215 35210>, se mostraría por pantalla 891340, 348124, 4531 (el segundo no es un número especial). Otro ejemplo, si la secuencia fuera <891340 14625 435621 0>, se mostrará por pantalla 891340 -1 -1 (sólo el primero es un número especial).

```
WITH Ada.Integer_Text_IO; USE Ada.Integer_Text_IO;  
WITH Es_Especial;  
  
PROCEDURE Secuencia_3_Especiales IS  
    N : Integer;  
    Cont : Natural := 0;  
  
BEGIN  
    LOOP  
        Get(N);  
        EXIT WHEN N=0 OR Cont=3;  
        IF Es_Especial(N) THEN  
            Put(N);  
            Cont:= Cont+1;  
        END IF;  
    END LOOP;  
  
    FOR I IN Cont+1..3 LOOP  
        Put(-1);  
    END LOOP;  
  
END Secuencia_3_Especiales;
```