

Fechas de realización: Del 28 de septiembre al 5 de octubre, según horario de cada grupo. Práctica voluntaria.

Objetivo: Diseño e implementación de un módulo que implementa el TAD genérico “*Números en base n*” (numerosbasen) que será implementado sobre un vector de dígitos naturales, y utilización del mismo para la realización de un programa que permita realizar operaciones aritméticas en distintas bases.

Descripción detallada:

1. Se desea desarrollar un TAD genérico que permita realizar operaciones aritméticas con números en bases 2,3,...,9 . Concretamente se desea que el módulo permita “definir” números en bases menores a 10 y realizar con éstos las operaciones de suma, resta, multiplicación y división entera. Para simplificar el procedimiento se puede implementar la representación de los números utilizando un vector que almacene cada dígito. Los números operados no tendrán nunca más de 80 dígitos y serán siempre positivos.

El TAD debe implementar las siguientes funciones:

```
generic
  type digito is (<>);
  --- el tipo enumerado de los dígitos válidos en base n
  with procedure sumadig(x,y:in digito; s,acarreo:out digito);
  --- procedimiento de suma de dígitos en base n. Devuelve el último dígito
  --- de la suma en s y en acarreo devuelve 1 si es que la suma ha excedido
  --- el dígito máximo
  with procedure restadig (x,y:in digito; r,acarreo:out digito);
  --- procedimiento de resta de dígitos en base n. Devuelve el último dígito
  --- de la resta en r y en acarreo devuelve 1 si es que la resta ha necesitado
  --- de un “préstamo”. Bajo estas condiciones restadig SIEMPRE es posible.
package numerosbasen is
  type nbasen is array(1..80) of digito;
  --- almacena los dígitos del número en base n
  procedure cero(v:out nbasen);
  --- asigna a un número en base n, el elemento inicial
  function suma(x,y:nbasen) return nbasen;
  --- suma un par de números en base n
  function resta(x,y:nbasen) return nbasen;
  --- resta un par de números en base n: x - Y
  function multip(x,y:nbasen) return nbasen;
  --- multiplica un par de números en base n
  function divide(x,y:nbasen) return nbasen;
  --- divide un par de números en base n: x / y
  function errorOp return boolean
  --- informa si la última función aritmética realizada ha producido un error.
private
  ...
end numerosbasen;
```

El programa debe permitir implementar las operaciones de suma, resta, multiplicación y división y además informar si las operaciones se han realizado correctamente o han producido un error (Ej. `resta(2,3)` produciría un error en tanto que `resta(3,2)` se ejecutaría correctamente).

Nótese que se pasan como parámetros genéricos el tipo enumerado con los dígitos válidos en base n así como dos procedimientos que permiten respectivamente sumar y restar estos dígitos. Además informan si la suma ha producido un acarreo o si la resta ha necesitado de un “préstamo” para ser ejecutada.

2. Utilizar el módulo anterior para desarrollar un programa que:
 - a) Permita definir dos tipos de números de bases b1 y b2, donde b1 es el último dígito mayor a 1 del NIP del primer alumno de la pareja y b2 es el último dígito mayor a 1 del NIP del segundo alumno de la pareja.
 - b) El programa debe presentar un menú de opciones para
 - Seleccionar la base de numeración (b1 o b2).
 - Leer números en la base seleccionada (iniciar el programa con la selección de la base b1 por defecto).
 - Realizar operaciones aritméticas en la base seleccionada (y enseñar los resultados, o un mensaje de error si la operación aritmética no ha sido realizada correctamente).
 - Salir del programa.
 - c) El programa debe indicar siempre cuál es la base en la que se está trabajando. El programa debe indicar si las operaciones se han efectuado correctamente, en caso contrario debe proporcionar un mensaje que explique la causa del fallo (Ej: Una multiplicación cuyo resultado excede de los 80 dígitos).
 - d) El módulo `numerosbasen` no debe implementar ningún procedimiento o función que muestre datos o mensajes por pantalla. Toda la interacción y mensajes con el usuario debe ser realizada por el programa principal.
 - e) El programa no debe interrumpir su ejecución en ningún caso.