Universidad de Valladolid

Departamento de Informática

Programación II. I.T.I. de Gestión

Ejercicios de prueba de programas

Curso 2009/10

Ejercicio 1

Un programa recibe como entrada un número entero y positivo de tres cifras y devuelve el número resultante de invertir sus cifras. Elaborar una batería de pruebas para este programa.

Ejercicio 2

En cierto lenguaje de programación las palabras definidas por el usuario deben construirse conforme a las siguientes reglas:

- Tienen como máximo 30 caracteres.
- El juego de caracteres utilizable es:
  - Letras (mayúsculas o minúsculas).
  - Dígitos (0–9).
  - guión (–).
- En los nombres se diferencian mayúsculas de minúsculas.
- El guión no puede estar ni al principio ni al final de la palabra, pero puede haber varios consecutivos.
- Deben contener al menos un carácter alfabético.
- No puede ser una de las palabras reservadas del lenguaje (if, data, real, . . .).

Realizar una batería de pruebas para validar un programa que recibe como entrada la palabra del usuario y proporciona como salida un mensaje de error, si la palabra no es correcta, o la palabra, si es correcta.

Ejercicio 3

Elaborar una batería de pruebas para un programa que analiza la validez de una palabra clave.
Una clave es válida cuando cumple los siguientes requisitos:

- Está formada por más de 7 y menos de 13 caracteres.
- Los caracteres permitidos son:
  - Las letras a – z, A – Z.
  - Los dígitos 0 – 9
  - El carácter especial % .
- Contiene al menos dos letras.
- Contiene al menos un carácter que no es letra.
- El primer y el último caracteres son letras.
- No aparece en un diccionario de palabras prohibidas (user%10a,user%aa, . . .).

UVa **Universidad** de **Valladolid.** Dpto. de Informática. Edificio de Tecnologías de la Información y las Telecomunicaciones, Campus Miguel Delibes. 47011 Valladolid. Tfno. 983.423000 ext.5617/Fax 983.423671.
www.infor.uva.es/~felix e-mail:felix@infor.uva.es

Ejercicio 4

Un subprograma tiene como entrada tres parámetros enteros x, y, z. Los valores de x e y debe representar un intervalo $[x, y]$ y el subprograma tiene como misión estudiar la pertenencia del valor z al intervalo. Las salidas posibles del programa son:

Extremo: Si z coincide con uno de los extremos del intervalo.

Medio: Si z es el punto medio del intervalo, pero no está en la situación anterior.

Interior: Si z está en el intervalo, pero no en las situaciones anteriores.

Exterior: Si z no está en el intervalo.

Error: si la entrada es errónea.

Elaborar una batería de pruebas para el subprograma mediante técnicas de partición de equivalencia con análisis de valores límite.

Ejercicio 5

Elaborar una batería de pruebas para un programa que lee una hora en formato hh:mm:ss e indica si la hora es correcta.

Ejercicio 6

Elaborar una batería de pruebas para un programa que procesa un archivo con datos de las calificaciones de alumnos, elaborando una lista de aprobados en la asignatura con la calificación obtenida en letra. La calificación de la asignatura es la media de las calificaciones parciales, todas ellas entre 0 y 10.

```
TRegsitro = Record
   Apellidos,Nombre: String[20];
   nota1,nota2,nota3,nota4:Real
End;
```

Ejercicio 7

Se pretende construir un sistema interactivo para grabar los resultados de unas hipotéticas elecciones. Para ello se utilizarán los siguientes ficheros:

- `partidos`, fichero maestro accesible por clave `COD-PARTIDO`.
- `resultados`, fichero secuencial, que se grabará en este proceso.

```
TRegPartido = Record
   CodPartido:String[3];
   NombrePartido:String[40];
   CabezaDeLista:String[30]
End;
```

```
TRegResultado=Record
    CodigoProvincia:Integer;
    CodigoMesa:Integer;
    CodigoPartido:String[3];
    NumeroDeVotos:Integer
End;
```

El programa de grabación consiste fundamentalmente en un formulario que se presenta al usuario y, una vez introducidos los datos, se comprueban las siguientes circunstancias:

- *COD-PROV* numérico, debe estar entre 1 y 50.
- *COD-MESA* numérico, entre 1 y 780.
- *COD-PARTIDO* debe existir en el fichero PARTIDOS.
- *NUM-VOTOS* numérico, entre 0 y 1350.

Si cualquiera de los datos es erróneo se muestra un mensaje en la última línea de la pantalla indicando el tipo de error, solicitando nuevamente los datos. Una vez que la respuesta sea correcta, se graba un registro en el fichero *RESULTADOS*.

Utilizando método(s) de caja negra, proponer una batería de pruebas para el programa, suponiendo que el fichero de *partidos* está validado.

Ejercicio 8

Elaborar una batería de pruebas para las siguientes funciónes, aplicando técnicas de caja blanca y negra.

```
Function Producto(x,y: integer): integer;
  var
    z: integer;
  begin
    z:= 0;
    while (x<>0) do
      begin
        if (odd(x)) then z:= z+y;
        x:= x div 2;
        y:= 2*y
      end;
    Prod:= z
  end;

Function MCD(x,y: integer): integer;
  var
    a,b: integer; {x > 0, y > 0}
  begin
    a:= x; b:= y;
    while (a<>b) do
      if a>b then a:= a-b else b:= b-a;
    MCD:= a {a = b = MCD(x, y)}
  end;
```

Ejercicio 9

La orden de 15 de septiembre de 2000 que modifica el reglamento general vehículos aprobado por real decreto 2822/1998 de 23 de diciembre dice:

> *En las placas de matrícula se inscribirán dos grupos de caracteres constituidos por un número de cuatro cifras, que irá desde el 0000 al 9999, y de tres letras, empezando por las letras BBB y terminando por las letras ZZZ, suprimiéndose las cinco vocales, y las letras Ñ, Q, CH y LL.*

Elaborar una batería de pruebas de caja negra con análisis de valores límite para un módulo que recibe una cadena de caracteres y determina si la cadena es una matrícula correcta o no en función de esta normativa.

Nota: Ejercicio de examen del curso 2008-2009