Relación de ejercicios de AWK

Ejercicio 1

El fichero "asignaturas4.txt" (disponible dentro del fichero asignaturas.zip que se puede descargar del campus virtual) contiene información de asignaturas de distintos cursos de la titulación. La asignaturas están desordenadas y el código de cada una de ellas es un indicativo del curso al que pertenecen. El contenido del citado fichero es:

```
101 Cálculo para la Computación 6
305 Sistemas Operativos 6
403 Calidad del Software 7.5
104 Matemáticas Discretas 12
205 Métodos Numéricos 6
106 Probabilidad y Estadística 7.5
207 Análisis y Diseño de Algoritmos 9
108 Organización Empresarial 6
401 Administración de Bases de Datos 7.5
302 Programación Concurrente 6
201 Tipos Abstractos de Datos 4.5
402 Gestión de Proyectos 4.5
303 Ingeniería de Requisitos 6
406 Arquitectura de Computadores 9
407 Procesadores de Lenguajes 7.5
206 Programación Orientada a Objetos 6
```

Se pide realizar un programa en AWK que realice las siguientes tareas:

- Separar en 4 ficheros distintos según el curso
- Mostrar en pantalla un resumen con los créditos por curso

  El resultado debe ser como:

- curso1.txt
  101 Cálculo para la Computación 6
  104 Matemáticas Discretas 12
  106 Probabilidad y Estadística 7.5
  108 Organización Empresarial 6
- curso2.txt
  205 Métodos Numéricos 6
  207 Análisis y Diseño de Algoritmos 9
  201 Tipos Abstractos de Datos 4.5
  206 Programación Orientada a Objetos 6
- curso3.txt
  305 Sistemas Operativos 6
  302 Programación Concurrente 6
  303 Ingeniería de Requisitos 6
- curso4.txt
  403 Calidad del Software 7.5
  401 Administración de Bases de Datos 7.5
  402 Gestión de Proyectos 4.5
  406 Arquitectura de Computadores 9
  407 Procesadores de Lenguajes 7.5
- Mostar (pantalla):
  Curso 1 = 31.5 créditos
  Curso 2 = 25.5 créditos
  Curso 3 = 18 créditos
  Curso 4 = 36 créditos

Ejercicio 2

Además de los modificades + y ?, también se pueden usar los modificadores {n} y {n,m}, que nos permiten indicar el número exacto o un rango de veces que se permite que se repita el elemento anterior. La expresión regular AB{2,4}C concuerda únicamente con ABBC, ABBBC y ABBBBC. Otra caracterísitica interesante es la instrucción "next", que cuando se ejecuta fuerza a pasar a la siguiente línea del fichero de entrada, aunque todavía queden pares patrón/acción sin procesar. Por último, es posible utilizar alias predefinidos que hacen referencia a cualquier carácter dentro de un conjunto determinado. Por ejemplo, [[:alpha:]] representa cualquier carácter alfabético, ya sea mayúscula o minúscula, o sea, es como si escribiéramos [a-zA-Z]. Por otra parte, con [[:space:]] estamos queriendo expresar cualquiera de los separadores habituales de palabras, o sea [\t\n\v\f] (\v es el tabulador vertical y \f es el avance de página). En la siguiente tabla se especifican estos alias:

alias	equivalencia	significado
[[:alnum:]]	[a-zA-Z0-9]	letras o dígitos
[[:alpha:]]	[A-Za-z]	cualquier letra
[[:blank:]]	[\t]	espacio o tabulador
[[:digit:]]	[0-9]	cualquier dígito decimal
[[:punct:]]		cualquier signo de puntuación
[[:space:]]	[\t\r\n\v\f]	cualquier carácter de espacio
[[:upper:]]	[A-Z]	letras mayúsculas
[[:lower:]]	[a-z]	letras minúsculas

Además de poder usar funciones predefinidas (sin, cos, substr ...), el programador puede usar las suyas propias. Dichas funciones se definen como (el alumno debe llamar a esta función en el programa que se esboza al final del ejercicio):

```
function obtenerNota(calificacion) {
   if (calificacion < 5)
      return "SUSPENSO";
   else if(calificacion < 7)
      return "APROBADO";
   else if(calificacion < 9)
      return "NOTABLE";
   else
      return "SOBRESALIENTE";
}
```

Supongamos que tenemos un fichero a procesar donde cada línea representa la calificación de un alumno en una asignatura concreta en la que está matriculado. Para todo alumno aparece primero su DNI, seguido del código de la asignatura y finalmente su calificación numérica. Los códigos de las asignaturas son secuencias que tienen entre 3 y 5 dígitos, y en cuanto a los DNI, son secuencias de 8 dígitos más una letra (en este ejercicio supondremos que puede ser cualquiera y que para un mismo alumno puede ponerse en mayúsculas o minúsculas). **Las calificaciones serán números desde 0.0 a 10.0. Pueden tener o no punto decimal, pero si lo tienen tendrán al menos una cifra decimal (puede que más).**

Se pide hacer un programa en AWK (usando los alias detallados al principio del ejercicio) que para cada alumno con identificador válido imprima una línea con su nota calculada a partir de la calificación media, donde NOTA puede ser SUSPENSO (<5), APROBADO (>=5 y <7), NOTABLE(>=7 y <9) o SOBRESALIENTE (>=9).Existen varios posibles errores que hay que manejar: (1) Si la línea no contiene exactamente 3 campos debe indicarse un error, indicando el número de línea, el tipo de error mostrando la línea incorrecta y pasamos a la siguiente línea sin procesar la actual. (2) Si el DNI no es válido se debe escribir otra línea de error de forma parecida al punto anterior. (3) De igual forma hay que actuar si el código de la asignatura no tiene el formato válido. (4) Si el formato de la calificación fuera incorrecto también se notifica como un error y pasamos a la siguiente línea. Si una línea está afectada por más de un error, basta con informar solo de uno de ellos. El fichero a procesar será como (ver calificaciones.txt):

```
28473636R 101 4.6
34567234D 103 5.7
X38372626L 101 8
02837272A 1s02 4.5
34567234d 102 10.000
28473636R 102 3.
...
```

Si resulta que para una misma asignatura y alumno, aparecen varias líneas en el fichero, la última debe prevalecer, ignorando cualquier calificación previa. El resultado que se debe obtener es:

```
ERROR EN LÍNEA 3: El campo 1 no tiene formato de DNI válido > X38372626L
ERROR EN LÍNEA 4: El campo 2 no tiene formato de código de asignatura > 1s02
ERROR EN LÍNEA 6: El campo 3 no es una calificación numérica correcta > 3.
ERROR EN LÍNEA 9: El campo 3 no es una calificación numérica correcta > 15.7
ERROR EN LÍNEA 13: Solo se admiten líneas formadas por 3 campos > jjkijijbbb hjlljl
ERROR EN LÍNEA 14: El campo 1 no tiene formato de DNI válido > 8473636R
----------------------
CALIFICACIONES MEDIAS
----------------------
38372626L SOBRESALIENTE (9.5)
34567234D APROBADO (6.8)
28473636R APROBADO (5.1)
02837272A SUSPENSO (4.4)
57637487T NOTABLE (8.2)
```

Se proporciona el siguiente esqueleto de código para que el alumno lo complete como estime conveniente:

```
!($1!="" && $2!="" && $3!="" && $4=="") {
   printf "ERROR EN LÍNEA %d: Solo se admiten líneas formadas por 3 campos > %s\n",NR,$0;
   next
}
$1 !~ /^[[:digit:]]{8}[[:alpha:]]$/ {
   printf "ERROR EN LÍNEA %d: El campo 1 no tiene formato de DNI válido > %s\n",NR,$1;
   next
}
...
{
   calificacion=$3;
   asignatura=$2;
   alumno= ...

   if (listaNotas[asignatura,alumno]=="")
    {
      listaNotas[asignatura,alumno]= calificacion;
      ...
    }
   else
    {
      ...
    }
}
END {
   print "---------------------- ";
   print "CALIFICACIONES MEDIAS ";
   print "---------------------- ";
   ...
}
```