

Grupo Jueves 12:00 – 14:00 semanas A

- Práctica 1 -

Autor: Inés Román Gracia

NIP: 820731

Ejercicio 5:

1. Resumen

En el fichero *ej5.* he creado los siguientes patrones para formatear los ficheros:

`^[]+`

Es para eliminar los espacios en blanco al principio de cada línea.

`[]+$`

Es para eliminar los espacios en blancos al final de cada línea.

`[]+ {printf(" ");}`

Es para eliminar uno o más espacios en blanco y añadir un solo espacio en blanco.

`^[]*\n`

Sirve para eliminar líneas vacías.

2. Pruebas

Los ficheros que he utilizado para comprobar el buen funcionamiento del programa son *entrada5* de entrada y *salida5* de salida, donde se puede ver que todos los patrones cumplen su función.

Ejercicio 6:

1. Resumen

En el fichero *ej6.* he creado los siguientes patrones para contar el número de líneas totales, de líneas en blanco, de caracteres, de líneas que contienen algún dígito y de líneas que empiezan por vocal:

`^[^0-9]*[0-9]`

Detecta líneas que empiezan o no por un dígito y las lee hasta detectar un dígito, incrementamos el contador de líneas que contienen un dígito y el de caracteres por cada carácter leído.

`^[aeiouAEIOU]`

Detecta las líneas que empiezan por una vocal, incrementamos el contador de líneas que empiezan por vocal y al de caracteres le sumamos una unidad.

`^[]*\n`

Detecta líneas vacías, incrementamos en una unidad el contador de líneas vacías y también el de líneas.

`\n`

Detecta los saltos de línea, incrementamos el contador de líneas.

`.`

Detecta el resto de caracteres, incrementamos el contador de caracteres.

2. Pruebas

Los ficheros que he utilizado para comprobar el buen funcionamiento del programa son *entrada6* de entrada y *salida6* de salida, donde se puede ver que todos los patrones cumplen su función.

Ejercicio 7:

1. Resumen

En el fichero *ej7* he escrito los siguientes comandos:

```
egrep '^[0-9].*[0-9]$\ ' t.txt
```

Este comando nos muestra las líneas del fichero *t.txt* que empiezan y acaban por un dígito, pudiendo contener cualquier carácter entre los dos dígitos.

```
egrep '^[^0-9]*$\ ' t.txt
```

Este comando nos muestra las líneas del fichero *t.txt* que no contienen ningún dígito.

```
egrep '^.*[0-9]*[02468]([0-9].*)*$' t.txt
```

Este comando nos muestra las líneas del fichero *t.txt* que contienen un número par. Al principio pueden contener cualquier carácter pero al llegar a un dígito, este debe formar un entero que acabara concatenado con un dígito par. Si la línea contiene más caracteres, el siguiente a este dígito par no podrá ser otro dígito.

2. Pruebas

El fichero que he utilizado para comprobar el funcionamiento de estos comandos es *t.txt*, además he añadido el fichero *salida7.txt*, en el que muestro los resultados que he obtenido por pantalla al ejecutar cada uno de estos comandos con el fichero *t.txt*.

Ejercicio 8:

1. Resumen

En el fichero *ej8.l* he creado los siguientes patrones:

Patrones declarados al inicio:

VECESN E*N

Este patrón me permitirá trabajar con las repeticiones del carácter *N*. No concatenamos *E** al final del patrón descrito porque sino no detectaríamos las cadenas que no contienen ninguna *N* pero que sí que contienen *E*, es decir, que estaríamos considerando que 0 no es múltiplo de 6.

MUL6 ({VECESN}{6})*E*

A partir del patrón anterior, puedo detectar las cadenas en las que *N* aparece un número múltiplo de 6. Añadimos *E** al final porque las cadenas que vamos a detectar también pueden acabar en *E*.

MUL3 ({VECESN}{3})*E*

Funciona de la misma forma que el patrón anterior, pero con múltiplos de 3.

Patrones que crean el fichero de salida:

l{MUL6}F {printf("--%s--", yytext);}

Como hemos dicho en los patrones declarados detectará las cadenas que empiezan por *l*, acaban por *F* y contengan *N* un número múltiplo de 6 veces y los modificará como se especifica en el enunciado y se ve en el fichero de salida *salida8*.

```
I{MUL3}F      {printf("-%s-", yytext);}
```

Detectará las cadenas que empiezan por *I*, acaban por *F* y que contengan *N* un número múltiplo de 3 (pero que no sea de 6) veces y los modificará como se puede ver en el fichero de salida *salida8*.

```
I{VECESN}*E*F {printf("**%s*", yytext);}
```

Detectará el resto de cadenas que empiezan por *I*, acaban por *F* y contienen cualquier patrón con *N* y *E* en el medio y los modificará como se puede ver en fichero de salida *salida8*.

2. Pruebas

Los ficheros que he utilizado para comprobar el buen funcionamiento del programa son *entrada8* de entrada y *salida8* de salida, donde se puede ver que el programa cumple su función.