

Grupo Miércoles 12:00-14:00 semanas A

Práctica 2

Autor: Diego Marco

Ejercicio 1.1:

1.Resumen

He creado una condición de arranque llamada fasta. Cuando detecta el patrón \n arranca esta condición, e indica que a continuación va a procesar la secuencia fasta. A continuación he añadido el patrón TT+|CC+|GG+|AA+ para que cuando detecte dos o mas TT o CC o GG o AA, saca por pantalla la letra detectada y seguidamente el número de veces que se repite. Finalmente cuando detecta el carácter > desactiva la condición de arranque fasta e inicia la condición de arranque por defecto.

2.Pruebas

Para la secuencia:

```
>AB128931.1 Homo sapiens mRNA for hypothetical protein, complete cds, Alzheimer
disease          site          specific          expressed          gene
TTCCGGCCGAGG
```

He obtenido:

```
>AB128931.1 Homo sapiens mRNA for hypothetical protein, complete cds, Alzheimer
disease          site          specific          expressed          gene
T2C2G2C2GAG2
```

Ejercicio 1.2:

1.Resumen

He creado la condición de arranque fasta. Cuando detecta el carácter \n se inicia esta condición de arranque. Seguidamente, he creado el patrón [T|C|G|A][0-9]+, para que cuando detecte T o C o G o A seguido de un numero, lo sustituye por un numero [0-9]+ de letras.

Cuando detecta el patron >, vuelve a la condición de arranque.

2.Pruebas

Para la secuencia:

```
>AB128931.1 Homo sapiens mRNA for hypothetical protein, complete cds, Alzheimer
disease          site          specific          expressed          gene
T2C2G2C2GAG2
```

He obtenido:

```
>AB128931.1 Homo sapiens mRNA for hypothetical protein, complete cds, Alzheimer
disease          site          specific          expressed          gene
TTCCGGCCGAGG
```

Ejercicio 1.2:

1.Resumen

He creado 3 condiciones de arranque para procesar los comentarios, y la condición de arranque por defecto la uso para procesar el código. El patrón “/*”.* inicia la condición de arranque coment2, el patrón “/*”.* inicia la condición de arranque coment1 si al finalizar la linea no aparecen los caracteres */, en caso contrario inicia la condición de arranque coment1. Para contar el número de lineas de los comentarios, en cada condición de arranque respectiva he incrementado un contador C. Para contar el número de lineas de código he usado un contador T que se incrementa cada vez que encuentra el carácter \n en la condición de arranque por defecto. Para contar el número de constantes he usado el patrón const.

Para hacer la media y desviación típica, he indexado el número de líneas de cada función den un vector cuyo índice representa el número total de funciones encontradas.

Finalmente para la puntuación he tenido en cuenta los casos pedidos en el enunciado y los he implementado con estructuras if .

2.Pruebas

He insertado:

```
/*Comentario
 *de
 *parrafo
 */
#include <stdio.h>
const int PI = 3.1416;
//Metodo mostrar
int mostrar()
{
    printf("Hola mundo \n")
}
/*Metodo main*/
void main()
{
    mostrar():
    if ( PI > 2 * PI )
        printf("Mala cosa \n")
}
```

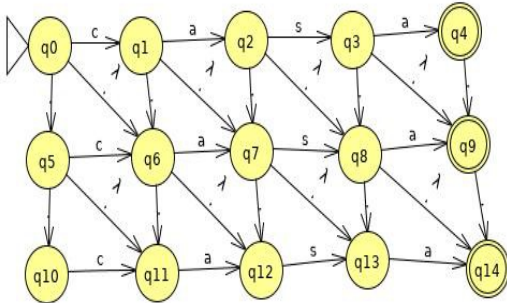
He obtenido:

T:12
C:6
CONST:1
F:2
FM:5.00
FS:1.0
P:7,5

Ejercicio 3:

1. Resumen

He creado el Afnd:



Usando el simulador Jflap, he convertido el Afnd en una expresión regular.

Posteriormente he traducido algunos trozos de la expresión regular para adaptarla a la sintaxis de flex. Por lo que me ha quedado la expresión regular:

```
casa|cas.?(ca.?(cas.?(c.?(ca.?(.c|.c)?a)s)a|(ca.?(cas.?(c.?(ca.?(.c|.c)?a)s).?(cas.?(ca.?(ca.?(c.?(cas.?(c.?(ca.?(.c|.c)?a)s)a).(((c.?(ca.?(.c|.c)?a).?(ca.?(cas.?(c.?(ca.?(.c|.c)?a)s).(((.c|.c)?.(c.?(ca.?(.c|.c)?a).(..?(.c|.c)?..c)a)s)a
```

La he usado como patrón en flex para hayar aquellas palabras cuya distancia Livenshtein de la palabra casa, es menor o igual a 2.

2. Pruebas

He insertado:

cas
ca
casas
mas
tango

He obtenido:

cas
ca
casas
mas