

Proyecto Final

Mark Brian Paredes Zavala

8 de febrero de 2022

Procesador de Textos.

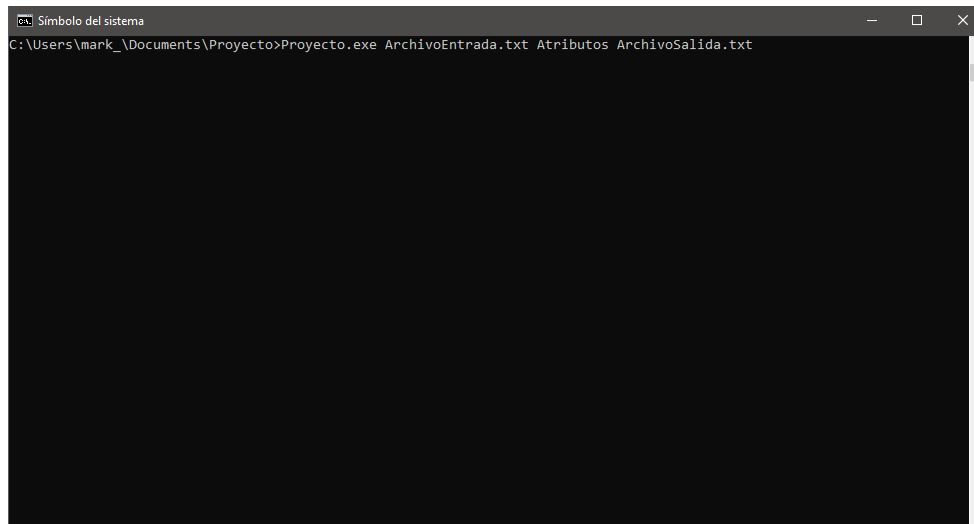
Este proyecto fue realizado en el lenguaje de C en el compilador de (DEV-C++)

El programa es un procesador de textos, mediante un archivo .txt que le llamaremos (ArchivoEntrada) y mediante una serie de comandos escritos, que realizaran una determinada acción, las cuales son:

1. L_ d1_ d2 _ dn: Atributo de la linea L imprimirá solamente las lineas indicadas por d1_ d2 _ dn... etc donde _ es el separador de dígitos.
2. C_ d1_ d2_ dn: Atributo columna imprimirá solamente las columnas indicadas por d1_ d2 _ dn... etc donde _ es el separador de digitos.
3. ILV_ d1_ d2_ dn: El atributo intercalar meterá un salto de linea en las lineas indicadas por d1_ d2 _ dn... etc donde _ es el separador de digitos.
4. IC_ d1_ d2_ dn: El atributo Intercalar columnas IC meterá un tabulador solamente en las lineas indicadas por d1_ d2 _ dn... etc donde _ es el separador de dígitos
5. CC (cadena) : Contador de cadenas, contar únicamente el numero de ocurrencias en cada cadena encontrada indicando el numero de ocurrencias en cada linea:
6. CP (cadena): El atributo contador de prefijos contara únicamente el numero de ocurrencias de la subcadena dada que sea prefijo de cada cadena del archivo, indicando el numero de ocurrencias de cada linea.
7. CS (cadena): El atributo contador de sufijos contara únicamente el numero de ocurrencias de la subcadena dada que sea sufijo de cada cadena del archivo, indicando el numero de ocurrencias de cada linea.
8. WC: Cuenta cadenas, caracteres y numero de lineas.
9. WNUM: Cuenta el numero de ocurrencia de enteros y flotantes.
10. INC: Insertara el numero de fila, numerando cada linea del archivo.

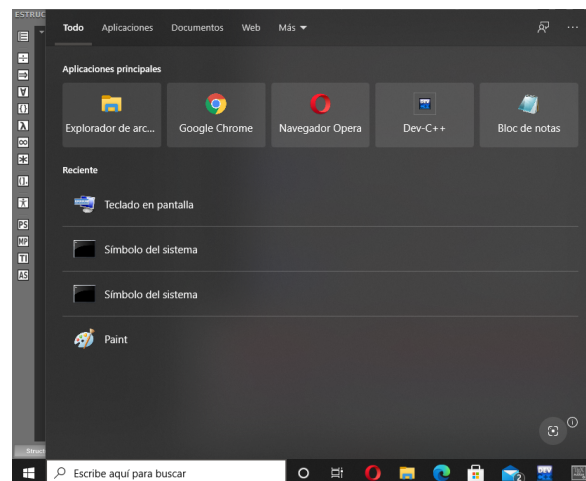
En consola realizaremos la ejecución del programa mediante la siguiente sintaxis:

"Programa Ejecutable.exe" "ArchivoEntrada.txt" "Atributo" "ArchivoSalida.txt"



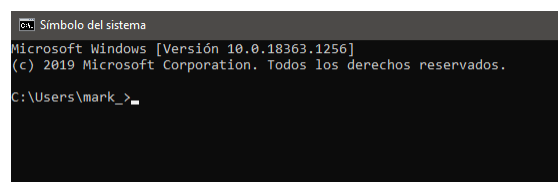
Así es como se vería el ingreso de la sintaxis en el cmd mejor conocido como "Símbolo del sistema". Antes tenemos que especificar al cmd la ruta donde tenemos nuestros archivos que utilizaremos para este programa

Primero tendremos que buscar el cmd en nuestro ordenador:

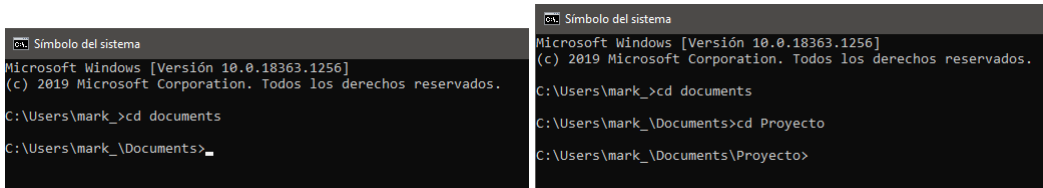


En mi caso ya lo tengo por predeterminado en mi inicio por lo que ingresare.

Cuando hayamos ingresado se nos abrirá el cmd de esta forma:



Dentro de cmd tendremos que especificar la ruta de donde tenemos nuestro proyecto:



De esta forma tendremos acceso a los archivos a utilizar, de igual forma, ejecutar nuestro programa desde cmd donde recibirá argumentos que vamos a mencionar mas adelante.

Ejecución del programa

De acuerdo a las acciones de los atributos y el hecho de ejecutar el programa en cmd se diseñó un algoritmo donde resolveremos las situaciones mencionadas anteriormente.

En primer lugar tendremos en cuenta que se recibirán argumentos así que, vamos a definir ¿Cómo vamos a pasar comando desde cmd?

Cuando queramos recibir comandos tendremos que ejecutar en nuestro `int main()` con argumentos donde serán los siguientes:

```
1 int main(int argc, char *argv[]) {  
2     //  
3 }
```

El `argc` (argument count) es considerado el número de argumentos por lo que se ingresa como argumento del `main` como `int`, `argv` (argument vectors) es un array de vectores que va a contener los elementos de `argc`

Vamos a considerar como `argv[0]` como el programa ejecutable, entonces si se ingresa un total de tres argumentos, el valor de `argc` será de 4 puesto y `argv[0]` tendrá como contenido el programa ejecutable (.exe)

Entonces, al tener el concepto de recibir argumentos por medio de cmd procedemos a realizar el algoritmo que sería idéntico a la última imagen mostrada.

De la misma manera, procedemos a deducir el algoritmo para identificar por medio de la línea de argumentos que atributo se eligió, en este programa al recibir 4 argumentos tendremos algo así:

`"argv[0]" "argv[1]" "argv[2]" "argv[3]"`

Esta sintaxis la utilizaremos de los ejemplos 1 al 4 así como del 8 al 10.

Caso contrario, cuando vayamos a ingresar los atributos del ejemplo 5 al 7 donde ingresaremos un argumento de más:

`"argv[0]" "argv[1]" "argv[2]" "argv[3]" "argv[4]"`

A continuación se mostraran la ejecución final del programa:

```
Archivo de entrada: Bloc de notas
Archivo  Edición  Formato  Ver  Ayuda
Kakarotto tu eres un guerrero admirable me acabo de dar cuenta de que yo no sirvo para pelear
con Majin Boo tu eres el unico que puede derrotarlo. Kakarotto la primera vez que te vi estaba
en busca de planetas con un excelente ambiente para despues venderlos.
Solamente vivia para cumplir mi objetivo superar los poderes de Kakarotto
Contaba la leyenda de que un Super Sayajin apareceria entre mi y si la leyenda seria cierta
yo seria el unico en convertirme en ese ser 23, Super dotado. Kakarotto es un guerrero de clase baja
y seria absurdo de que el fuera el Super Sayajin.
Yo pude despertar el Super Sayajin que habia en mi al haber destado mi furia estaba emocionado
pense que finalmente habia superado a Kakarotto y esos momentos que tube alguna vez como el
principe Sayajin habian regresado pero a fin de cuentas no 2.456 logre superarte.
Al principio pense que era tu obligacion proteger a tus seres queridos y a causa de eso un poder
totalmente desconocido brotaba en tu corazon quiza tenga razon yo tengo el mismo deber
Antes yo peleaba para que todo se hiciera a mi voluntad, por diversion era una dimisia para
mi matar a la gente y sobre todo para fortalecer mi orgullo 2.4
Pero Kakarotto tu eres diferente no peleabas solamente para ganar siempre sobrepasaste el
limite de tus fuerzas para no perder ante nadie, por eso tu nunca te atreviste a matar a tus
oponentes, lo se porque tu nunca te atreviste a matarme 2 parece ser que finalmente he comprendido
que en mi corazon hay un poco de sentimiento que suelen tener 23 los humanos, pero no puedo soportar
la idea de que existe un Sayajin 1 generoso que le gusta pelear.
Tu puedes Kakarotto eres el numero 1
C:\Users\mark_\Documents\Proyecto>
```

```
Simbolo del sistema
C:\Users\mark_\Documents\Proyecto>Projecto.exe ArchivoEntrada.txt I_1_2 ArchivoSalida.txt
Kakarotto tu eres un guerrero admirable me acabo de dar cuenta de que yo no sirvo para pelear
con Majin Boo tu eres el unico que puede derrotarlo. Kakarotto la primera vez que te vi estaba
en busca de planetas con un excelente ambiente para despues venderlos.
Solamente vivia para cumplir mi objetivo superar los poderes de Kakarotto
Contaba la leyenda de que un Super Sayajin apareceria entre mi y si la leyenda seria cierta
yo seria el unico en convertirme en ese ser 23, Super dotado. Kakarotto es un guerrero de clase baja
y seria absurdo de que el fuera el Super Sayajin.
Yo pude despertar el Super Sayajin que habia en mi al haber destado mi furia estaba emocionado
pense que finalmente habia superado a Kakarotto y esos momentos que tube alguna vez como el
principe Sayajin habian regresado pero a fin de cuentas no 2.456 logre superarte.
Al principio pense que era tu obligacion proteger a tus seres queridos y a causa de eso un poder
totalmente desconocido brotaba en tu corazon quiza tenga razon yo tengo el mismo deber
Antes yo peleaba para que todo se hiciera a mi voluntad, por diversion era una dimisia para
mi matar a la gente y sobre todo para fortalecer mi orgullo 2.4
Pero Kakarotto tu eres diferente no peleabas solamente para ganar siempre sobrepasaste el
limite de tus fuerzas para no perder ante nadie, por eso tu nunca te atreviste a matar a tus
oponentes, lo se porque tu nunca te atreviste a matarme 2 parece ser que finalmente he comprendido
que en mi corazon hay un poco de sentimiento que suelen tener 23 los humanos, pero no puedo soportar
la idea de que existe un Sayajin 1 generoso que le gusta pelear.
Tu puedes Kakarotto eres el numero 1
C:\Users\mark_\Documents\Proyecto>
```

```
Simbolo del sistema
C:\Users\mark_\Documents\Proyecto>Projecto.exe ArchivoEntrada.txt ILV_1_2 ArchivoSalida.txt
Kakarotto tu eres un guerrero admirable me acabo de dar cuenta de que yo no sirvo para pelear
con Majin Boo tu eres el unico que puede derrotarlo. Kakarotto la primera vez que te vi estaba
en busca de planetas con un excelente ambiente para despues venderlos.
Solamente vivia para cumplir mi objetivo superar los poderes de Kakarotto
Contaba la leyenda de que un Super Sayajin apareceria entre mi y si la leyenda seria cierta
yo seria el unico en convertirme en ese ser 23, Super dotado. Kakarotto es un guerrero de clase baja
y seria absurdo de que el fuera el Super Sayajin.
Yo pude despertar el Super Sayajin que habia en mi al haber destado mi furia estaba emocionado
pense que finalmente habia superado a Kakarotto y esos momentos que tube alguna vez como el
principe Sayajin habian regresado pero a fin de cuentas no 2.456 logre superarte.
Al principio pense que era tu obligacion proteger a tus seres queridos y a causa de eso un poder
totalmente desconocido brotaba en tu corazon quiza tenga razon yo tengo el mismo deber
Antes yo peleaba para que todo se hiciera a mi voluntad, por diversion era una dimisia para
mi matar a la gente y sobre todo para fortalecer mi orgullo 2.4
Pero Kakarotto tu eres diferente no peleabas solamente para ganar siempre sobrepasaste el
limite de tus fuerzas para no perder ante nadie, por eso tu nunca te atreviste a matar a tus
oponentes, lo se porque tu nunca te atreviste a matarme 2 parece ser que finalmente he comprendido
que en mi corazon hay un poco de sentimiento que suelen tener 23 los humanos, pero no puedo soportar
la idea de que existe un Sayajin 1 generoso que le gusta pelear.
Tu puedes Kakarotto eres el numero 1
C:\Users\mark_\Documents\Proyecto>
```

```
C:\Users\mark_\Documents\Proyecto>Projecto.exe ArchivoEntrada.txt CC Kakarotto ArchivoSalida.txt
El numero de ocurrencias de la cadena [Kakarotto] es de: 7
Usted ingreso la opcion 5
```

```
C:\Users\mark_\Documents\Proyecto>Projecto.exe ArchivoEntrada.txt CS Tito ArchivoSalida.txt
El numero de ocurrencias de la cadena [tito] es de: 7
Usted ingreso la opcion 7
```

```
C:\Users\mark_\Documents\Proyecto>Projecto.exe ArchivoEntrada.txt MMUM ArchivoSalida.txt
El numero de enteros que existen es: 5
El numero de flotantes que existen es: 2
Usted ingreso la opcion 8
```

Podemos observar el texto contenido en el archivo se conocerá como "Archivo de entrada", el cual ocuparemos para todo el proyecto, no sera modificando con la ejecución de cada comando, si no el modificado sera el "Archivo de Salida".

```
Simbolo del sistema
C:\Users\mark_\Documents\Proyecto>Projecto.exe ArchivoEntrada.txt C_1_2 ArchivoSalida.txt
Kakarotto tu eres un guerrero admirable me acabo de dar cuenta de que yo no sirvo para pelear
con Majin Boo tu eres el unico que puede derrotarlo. Kakarotto la primera vez que te vi estaba
en busca de planetas con un excelente ambiente para despues venderlos.
Solamente vivia para cumplir mi objetivo superar los poderes de Kakarotto
Contaba la leyenda de que un Super Sayajin apareceria entre mi y si la leyenda seria cierta
yo seria el unico en convertirme en ese ser 23, Super dotado. Kakarotto es un guerrero de clase baja
y seria absurdo de que el fuera el Super Sayajin.
Yo pude despertar el Super Sayajin que habia en mi al haber destado mi furia estaba emocionado
pense que finalmente habia superado a Kakarotto y esos momentos que tube alguna vez como el
principe Sayajin habian regresado pero a fin de cuentas no 2.456 logre superarte.
Al principio pense que era tu obligacion proteger a tus seres queridos y a causa de eso un poder
totalmente desconocido brotaba en tu corazon quiza tenga razon yo tengo el mismo deber
Antes yo peleaba para que todo se hiciera a mi voluntad, por diversion era una dimisia para
mi matar a la gente y sobre todo para fortalecer mi orgullo 2.4
Pero Kakarotto tu eres diferente no peleabas solamente para ganar siempre sobrepasaste el
limite de tus fuerzas para no perder ante nadie, por eso tu nunca te atreviste a matar a tus
oponentes, lo se porque tu nunca te atreviste a matarme 2 parece ser que finalmente he comprendido
que en mi corazon hay un poco de sentimiento que suelen tener 23 los humanos, pero no puedo soportar
la idea de que existe un Sayajin 1 generoso que le gusta pelear.
Tu puedes Kakarotto eres el numero 1
C:\Users\mark_\Documents\Proyecto>
```

```
Simbolo del sistema
C:\Users\mark_\Documents\Proyecto>Projecto.exe ArchivoEntrada.txt IC_1_2 ArchivoSalida.txt
El numero de caracteres encontrados es: 1370
El numero de lineas encontradas es: 28
El numero de cadenas encontradas es: 301
Usted ingreso la opcion 8
```

```
C:\Users\mark_\Documents\Proyecto>Projecto.exe ArchivoEntrada.txt CP Kak ArchivoSalida.txt
El numero de ocurrencias de la cadena [kak] es de: 7
Usted ingreso la opcion 6
```

```
C:\Users\mark_\Documents\Proyecto>Projecto.exe ArchivoEntrada.txt MC ArchivoSalida.txt
El numero de caracteres encontrados es: 1370
El numero de lineas encontradas es: 28
El numero de cadenas encontradas es: 301
Usted ingreso la opcion 8
```

```
Simbolo del sistema
C:\Users\mark_\Documents\Proyecto>Projecto.exe ArchivoEntrada.txt IMC ArchivoSalida.txt
Kakarotto tu eres un guerrero admirable me acabo de dar cuenta de que yo no sirvo para pelear
con Majin Boo tu eres el unico que puede derrotarlo. Kakarotto la primera vez que te vi estaba
en busca de planetas con un excelente ambiente para despues venderlos.
Solamente vivia para cumplir mi objetivo superar los poderes de Kakarotto
Contaba la leyenda de que un Super Sayajin apareceria entre mi y si la leyenda seria cierta
yo seria el unico en convertirme en ese ser 23, Super dotado. Kakarotto es un guerrero de clase baja
y seria absurdo de que el fuera el Super Sayajin.
Yo pude despertar el Super Sayajin que habia en mi al haber destado mi furia estaba emocionado
pense que finalmente habia superado a Kakarotto y esos momentos que tube alguna vez como el
principe Sayajin habian regresado pero a fin de cuentas no 2.456 logre superarte.
Al principio pense que era tu obligacion proteger a tus seres queridos y a causa de eso un poder
totalmente desconocido brotaba en tu corazon quiza tenga razon yo tengo el mismo deber
Antes yo peleaba para que todo se hiciera a mi voluntad, por diversion era una dimisia para
mi matar a la gente y sobre todo para fortalecer mi orgullo 2.4
Pero Kakarotto tu eres diferente no peleabas solamente para ganar siempre sobrepasaste el
limite de tus fuerzas para no perder ante nadie, por eso tu nunca te atreviste a matar a tus
oponentes, lo se porque tu nunca te atreviste a matarme 2 parece ser que finalmente he comprendido
que en mi corazon hay un poco de sentimiento que suelen tener 23 los humanos, pero no puedo soportar
la idea de que existe un Sayajin 1 generoso que le gusta pelear.
Tu puedes Kakarotto eres el numero 1
Usted ingreso la opcion 10
C:\Users\mark_\Documents\Proyecto>
```

Ahora se mostrara la lo que se escribió en el ArchivoSalida de acuerdo a cada ejecución:

```
ArchivoSalida: Bloc de notas
Archivo  Edición  Formato  Ver  Ayuda
Kakarotto tu eres un guerrero admirable me acabo de dar cuenta de que yo no sirvo para pelear
con Majin Boo tu eres el unico que puede derrotarlo. Kakarotto la primera vez que te vi estaba
```

```
ArchivoSalida: Bloc de notas
Archivo  Edición  Formato  Ver  Ayuda
Ka
co
en
So
Co
yo
y
Yo
pe
pr
Al
to
An
mi
Pe
li
op
qu
la
Tu
```

```
ArchivoSalida: Bloc de notas
Archivo  Edición  Formato  Ver  Ayuda
Kakarotto tu eres un guerrero admirable me acabo de dar cuenta de que yo no sirvo para pelear
con Majin Boo tu eres el unico que puede derrotarlo. Kakarotto la primera vez que te vi estaba

en busca de planetas con un excelente ambiente para despues venderlos.
Solamente vivia para cumplir mi objetivo superar los poderes de Kakarotto
Contaba la leyenda de que un Super Sayajin apareceria entre mi y si la leyenda seria cierta
yo seria el unico en convertirme en ese ser 23, Super dotado. Kakarotto es un guerrero de clase baj
y seria absurdo de que el fuera el Super Sayajin.
Yo pude despertar el Super Sayajin que habia en mi al haber destado mi furia estaba emocionado
pense que finalmente habia superado a Kakarotto y esos momentos que tube alguna vez como el
principe Sayajin habian regresado pero a fin de cuentas no 2.456 logre superarte.
Al principio pense que era tu obligacion proteger a tus seres queridos y a causa de eso un poder
totalmente desconocido brotaba en tu corazon quiza tenga razon yo tengo el mismo deber
Antes yo peleaba para que todo se hiciera a mi voluntad, por diversion era una dimisia para
mi matar a la gente y sobre todo para fortalecer mi orgullo 2.4
Pero Kakarotto tu eres diferente no peleabas solamente para ganar siempre sobrepasaste el
limite de tus fuerzas para no perder ante nadie, por eso tu nunca te atreviste a matar a tus
oponentes, lo se porque tu nunca te atreviste a matarme 2 parece ser que finalmente he comprendido
que en mi corazon hay un poco de sentimiento que suelen tener 23 los humanos, pero no puedo soporta
la idea de que existe un Sayajin 1 generoso que le gusta pelear.
Tu puedes Kakarotto eres el numero 1
```

```
ArchivoSalida: Bloc de notas
Archivo  Edición  Formato  Ver  Ayuda
K a
karotto tu eres un guerrero admirable me acabo de dar cuenta de que yo no sirvo para pelear
c o n Majin Boo tu eres el unico que puede derrotarlo. Kakarotto la primera vez que te vi estaba
e n busca de planetas con un excelente ambiente para despues venderlos.
s o lamente vivia para cumplir mi objetivo superar los poderes de kakarotto
c o ntaba la leyenda de que un Super Sayajin apareceria entre mi y si la leyenda seria cierta
y o seria el unico en convertirme en ese ser 23, Super dotado. kakarotto es un guerrero de clase baja
y seria absurdo de que el fuera el Super Sayajin.
Y o pude despertar el Super Sayajin que habia en mi al haber destado mi furia estaba emocionado
p e nse que finalmente habia superado a kakarotto y esos momentos que tube alguna vez como el
p r incipe Sayajin habian regresado pero a fin de cuentas no 2.456 logre superarte.
A l principio pense que era tu obligacion proteger a tus seres queridos y a causa de eso un poder
t o talmente desconocido brotaba en tu corazon quiza tenga razon yo tengo el mismo deber
A n tes yo peleaba para que todo se hiciera a mi voluntad, por diversion era una dimisia para
m i matar a la gente y sobre todo para fortalecer mi orgullo 2.4
P e ro kakarotto tu eres diferente no peleabas solamente para ganar siempre sobrepasaste el
l i mite de tus fuerzas para no perder ante nadie, por eso tu nunca te atreviste a matar a tus
o p onentes, lo se porque tu nunca te atreviste a matarme 2 parece ser que finalmente he comprendido
q u e en mi corazon hay un poco de sentimiento que suelen tener 23 los humanos, pero no puedo soportar
l a idea de que existe un Sayajin 1 generoso que le gusta pelear.
T u puedes kakarotto eres el numero 1
```

```
ArchivoSalida: Bloc de notas
Archivo  Edición  Formato  Ver  Ayuda
El numero de ocurrencias de la cadena [Kakarotto] es de 7:
```

```
ArchivoSalida: Bloc de notas
Archivo  Edición  Formato  Ver  Ayuda
El numero de ocurrencias de la cadena [Kak] es de: 7
```

```
ArchivoSalida: Bloc de notas
Archivo  Edición  Formato  Ver  Ayuda
El numero de ocurrencias de la cadena [tto] es de: 7
```

```
ArchivoSalida: Bloc de notas
Archivo  Edición  Formato  Ver  Ayuda
El numero de caracteres encontrados es: 1370
El numero de lineas encontradas es: 20
El numero de cadenas encontradas es: 301
```

```
ArchivoSalida: Bloc de notas
Archivo  Edición  Formato  Ver  Ayuda
El numero de enteros que existen es: 5
El numero de flotantes que existen es: 2
```

```
ArchivoSalida: Bloc de notas
Archivo  Edición  Formato  Ver  Ayuda
1 - kakarotto tu eres un guerrero admirable me acabo de dar cuenta de que yo no sirvo para pelear
2 - con Majin Boo tu eres el unico que puede derrotarlo. kakarotto la primera vez que te vi estaba
3 - en busca de planetas con un excelente ambiente para despues venderlos.
4 - Solamente vivia para cumplir mi objetivo superar los poderes de kakarotto
5 - Contaba la leyenda de que un Super Sayajin apareceria entre mi y si la leyenda seria cierta
6 - yo seria el unico en convertirme en ese ser 23, Super dotado. kakarotto es un guerrero de clase baja
7 - y seria absurdo de que el fuera el Super Sayajin.
8 - Yo pude despertar el Super Sayajin que habia en mi al haber destado mi furia estaba emocionado
9 - pense que finalmente habia superado a kakarotto y esos momentos que tube alguna vez como el
10 - principe Sayajin habian regresado pero a fin de cuentas no 2.456 logre superarte.
11 - Al principio pense que era tu obligacion proteger a tus seres queridos y a causa de eso un poder
12 - totalmente desconocido brotaba en tu corazon quiza tenga razon yo tengo el mismo deber
13 - Antes yo peleaba para que todo se hiciera a mi voluntad, por diversion era una dimisia para
14 - mi matar a la gente y sobre todo para fortalecer mi orgullo 2.4
15 - Pero kakarotto tu eres diferente no peleabas solamente para ganar siempre sobrepasaste el
16 - limite de tus fuerzas para no perder ante nadie, por eso tu nunca te atreviste a matar a tus
17 - oponentes, lo se porque tu nunca te atreviste a matarme 2 parece ser que finalmente he comprendido
18 - que en mi corazon hay un poco de sentimiento que suelen tener 23 los humanos, pero no puedo soportar
19 - la idea de que existe un Sayajin 1 generoso que le gusta pelear.
20 - Tu puedes kakarotto eres el numero 1
```

Como se pudo observar el funcionamiento de código fue de manera correcta, así mismo la salida de la información hacia el Archivo de salida.

Bibliotecas y funciones a utilizar en este programa.

A continuacion se muestran las funciones usadas para realizar este proyecto, mediante las cuales se adentrara en especifico a lo que aportaran a este programa:

Bibliotecas

1. `# include <stdio.h>`: Nos permite manipular con funciones las cuales podremos manipular así como el manejo de archivos:

Entrada y salida de funciones

- a) `fopen`: Abre un archivo, en el archivo se especifica si se leera, modificara, leer y modificar, etc.
- b) `fclose`: Cierra un archivo con el valor asociado.

Manejo de archivos

- a) `feof`: Comprueba si al final de un archivo pasa determinado flujo
 - b) `fgets`: obtiene una cadena desde el archivo y termina en una nueva linea o al final de archivo.
 - c) `fgetc`: Devuelve el carácter de un archivo.
 - d) `fscanf`: Obtiene el contenido del archivo (hasta el primer espacio en blanco) y los asigna en una variable.
 - e) `fprintf`: Imprime en un archivo.
 - f) `printf`: Imprime en consola.
 - g) `sscanf`: Obtiene los datos del programa y las asigna en una variable.
2. `# include<string.h>`: Biblioteca para el uso de funciones que manipulan cadenas.
 - a) `strcpy(cadena1,cadena2)`: Copia el valor de `cadena1` en `cadena2`
 - b) `strcmp(cadena1,cadena2)`: Devuelve 0 si `cadena1` y `cadena2` son iguales, de lo contrario devolverá cualquier otro valor.
 - c) `strstr(cadena1,cadena2)`: Busca si `cadena2` es subcadena de `cadena1`.
 3. `# include<stdlib.h>` : Contiene prototipos de funciones para gestionar memoria dinámica y otros procesos.
 4. `# include<stdbool.h>` : Biblioteca para el manejo de valores booleanos.

Construcción del programa

Como primer punto a resaltar, para el desarrollo de programa se utilizaron diferentes recursos, como lo es el libro 'Programación Estructurada a fondo' del autor de Ing Pablo Augusto, así como fuentes del internet.

Tras analizar el problema y realizar un algoritmo para poderlo resolver, se inicio sabiendo el ¿Como se realizara la acción que el usuario requiere? tras esto, el uso de una estructura de decisión fue la elegida, por el hecho de saber que acción realizar de acuerdo al comando, por lo que el empleo de if fue necesario; esto va requerir de comparar cadenas, se hace uso de la biblioteca <string.h> para poder utilizar strcmp(), así como strcpy().

Así que el código fue el siguiente:

```
int main(int argc, char *argv[]) {
    const char *archivoEnt = argv[1];
    const char *archivoSal = argv[argc - 1];
    char argumento[256];
    char comando[256];
    strcpy(argumento, argv[2]);

    if(strcmp(argumento, "CC")==0) {
        contarCadena(argv[1], argv[3], argv[4]);
        printf("Usted ingreso la opcion 5");
    }
    else if(strcmp(argumento, "CP")==0) {
        contarSubPre(argv[1], argv[3], argv[4]);
        printf("Usted ingreso la opcion 6");
    }
    else if(strcmp(argumento, "CS")==0) {
        contarSubSuf(argv[1], argv[3], argv[4]);
        printf("Usted ingreso la opcion 7");
    }
    else if(strcmp(argumento, "MC")==0) {
        FILE *archivoSalida=fopen(archivoSal, "w");
        printf("El numero de caracteres encontrados es: %d\n", contarCaracteres(archivoEnt));
        printf("El numero de lineas encontradas es: %d\n", contarLineas(archivoEnt));
        printf("El numero de cadenas encontradas es: %d\n", contarCadenas(archivoEnt));
        //Aquí mandamos a imprimir en el archivo de salida la información
        fprintf(archivoSalida, "El numero de caracteres encontrados es: %d\n", contarCaracteres(archivoEnt));
        fprintf(archivoSalida, "El numero de lineas encontradas es: %d\n", contarLineas(archivoEnt));
        fprintf(archivoSalida, "El numero de cadenas encontradas es: %d\n", contarCadenas(archivoEnt));
        printf("Usted ingreso la opcion 8\n");
        fclose(archivoSalida);
    }
}
```

```
    else if(strcmp(argumento, "MC")==0) {
        FILE *archivoSalida=fopen(archivoSal, "w");
        printf("El numero de caracteres encontrados es: %d\n", contarCaracteres(archivoEnt));
        printf("El numero de lineas encontradas es: %d\n", contarLineas(archivoEnt));
        printf("El numero de cadenas encontradas es: %d\n", contarCadenas(archivoEnt));
        //Aquí mandamos a imprimir en el archivo de salida la información
        fprintf(archivoSalida, "El numero de caracteres encontrados es: %d\n", contarCaracteres(archivoEnt));
        fprintf(archivoSalida, "El numero de lineas encontradas es: %d\n", contarLineas(archivoEnt));
        fprintf(archivoSalida, "El numero de cadenas encontradas es: %d\n", contarCadenas(archivoEnt));
        printf("Usted ingreso la opcion 8\n");
        fclose(archivoSalida);
    }
    else if(strcmp(argumento, "MUN")==0) {
        FILE *archivoSalida=fopen(archivoSal, "w");
        printf("El numero de enteros que existen es: %d\n", contarEnteros(archivoEnt));
        printf("Usted ingreso la opcion 9");
        //Aquí mandamos a imprimir en el archivo de salida la información
        fprintf(archivoSalida, "El numero de enteros que existen es: %d\n", contarEnteros(archivoEnt));
        fprintf(archivoSalida, "El numero de flotantes que existen es: %d\n", contarFlotantes(archivoEnt));
        fclose(archivoSalida);
    }
    else if(strcmp(argumento, "TMC")==0) {
        printf("Usted ingreso la opcion 10");
    }
}
```

Primero, usamos strcpy para copiar el contenido de argv[2] que tiene dentro el comando a utilizar a la variable 'argumento' esto nos permitirá saber que decisión de los if tomar.

Aquí es importante señalar que estamos haciendo uso de las anteriores funciones para los puntos 5 al 10 donde tendremos un formato de sintaxis diferente, como se explico anteriormente.

Se puede observar que cada decisión tiene dentro una función diferente en ella se encuentran diferentes argumentos en especifico en 4 de las 6 decisiones que fueron mostraron reciben 3 argumentos:

"ArchivoEntrada.txt" "Argumento" "ArchivoSalida.txt"

Que a su vez son;

"argv[1]" "argv[2]" "argv[3]"

Ahora vamos adentrarnos en cada una de las funciones que se mandan a llamar dentro de cada if y else if

A continuación se hablara del punto 5 al 7 donde vamos a utilizar la siguiente sintaxis, ya mencionada en la pagina 4:

"argv[0]" "argv[1]" "argv[2]" "argv[3]" "argv[4]"

Donde la cadena a utilizar sera "argv[3]"

5. CC - Contador de Cadenas

Esta es una función de tipo void por lo que la función no retornara nada.

```
void contarCadena(const char *entrada, const char *cadena, const char *salida){
    int contPalabIguales=0;
    char cadenaTemporal[256];
    FILE* archivoEntrada=fopen(entrada,"r");

    while(!feof(archivoEntrada)){
        fscanf(archivoEntrada,"%s",cadenaTemporal);

        if(strcmp(cadenaTemporal,cadena)==0){
            contPalabIguales=contPalabIguales+1;
        }
    }

    fclose(archivoEntrada);
    FILE* archivoSalida=fopen(salida,"w");
    fprintf(archivoSalida,"El numero de ocurrencias de la cadena [%s] es de %d\n",cadena,contPalabIguales);
    printf("El numero de ocurrencias de la cadena [%s] es de: %d\n",cadena,contPalabIguales);
    fclose(archivoSalida);
}
```

Declaramos una función con el nombre "contarCadenas" y sera de tipo entero, dentro de ella declaramos dos variables; una sera de tipo int, va actuar como el contador de palabras, la otra de tipo char donde vamos almacenar el contenido del archivo.

Abrimos el archivo de entrada que es donde tenemos toda nuestra información guardada, así que vamos a leer su contenido.

Después de haber leído el archivo, realizaremos un ciclo donde, mientras que el archivo de entrada no llegue al "end of file", extraiga en contenido del archivo, lo guarde en la variable tipo char llamada "cadenaTemporal" y mediante una decisión usando la función strcmp de la cual ya hablamos anteriormente, el retorno de la comparación de dos cadenas en este caso la que el usuario escriba y el contenido del archivo, si estas son iguales retornara un 0 y el contador va ir en incremento. Cerramos el archivo de entrada, ahora vamos a abrir el archivo de salida donde vamos a escribir el numero de ocurrencias que tenemos de la palabra indicada por el usuario, así que mandaremos a imprimir las veces que el contador fue en incremento, al terminar de escribir en el archivo de salida procedemos a cerrar el archivo de salida.

6. CP - Contador de subcadena prefijos

Esta es una función de tipo void por lo que la función no retornara nada.

```
void contarSubCPre(const char *entrada, const char *cadena, const char *salida){
    int contarSubC=0;
    char cadenaTemporal[256];
    FILE* archivoEntrada=fopen(entrada,"r");

    while(!feof(archivoEntrada)){
        fscanf(archivoEntrada,"%s",cadenaTemporal);

        if(strstr(cadenaTemporal,cadena)==cadenaTemporal && strcmp(cadenaTemporal,cadena)!=0){
            contarSubC=contarSubC+1;
        }
    }

    fclose(archivoEntrada);
    FILE* archivoSalida=fopen(salida,"w");
    printf("El numero de ocurrencias de la cadena [%s] es de: %d\n",cadena,contarSubC);
    fprintf(archivoSalida,"El numero de ocurrencias de la cadena [%s] es de: %d\n",cadena,contarSubC);
    fclose(archivoSalida);
}
```

tarSubPre" y sera de tipo entero, dentro de ella declaramos dos variables; una de tipo entero y la otra de tipo char, la primera sera un contador, y la ultima para almacenar el contenido del fichero. Abrimos el archivo de entrada para leer su contenido. Ahora realizamos un ciclo donde mientras no sea "end of file" almacenaremos el contenido del archivo dentro de la variable char declarada anteriormente, después mediante una condición

Declaramos una función con el nombre "con-

"Si es igual la subcadena de cadena y cadenaTemporal con cadenaTemporal y a su vez que el retorno de la comparación entre cadenaTemporal y cadena sea diferente de 0" esto porque lo que queremos es encontrar el prefijo, que este siempre va a iniciar al principio de cadena, de igual forma consideramos que si queremos saber si perro es prefijo de perro el programa indique que esto es error, ya que el que sean iguales palabras no los hace subcadena de otra. Al cumplir con los parámetros anteriores empezamos a hacer un conteo de palabras que cumplen con el mismo. Cerramos el archivo de entrada y abrimos el archivo de salida que sera el de escritura, mandamos a imprimir en el archivo de salida las veces que el contador aumento mediante un fprintf y por ultimo cerramos el archivo de salida

7. CS - Contador de subcadena sufixo

Esta es una función de tipo void por lo que no retornara nada.

```
void contarSubSuf(const char *entrada, const char *cadena, const char *salida){
    int contarSubC=0;
    char cadenaTemporal[256];
    FILE* archivoEntrada=fopen(entrada,"r");

    while(!feof(archivoEntrada)){
        fscanf(archivoEntrada,"%s",cadenaTemporal);
        if(strstr(cadenaTemporal,cadena)!=&cadenaTemporal[0] && strstr(cadenaTemporal,cadena)){
            contarSubC=contarSubC+1;
        }
    }
    fclose(archivoEntrada);
    FILE* archivoSalida=fopen(salida,"w");
    printf("El numero de ocurrencias de la cadena [%s] es de: %d \n",cadena,contarSubC);
    fprintf(archivoSalida,"El numero de ocurrencias de la cadena [%s] es de: %d \n",cadena,contarSubC);
    fclose(archivoSalida);
}
```

Declaramos una función con el nombre "con-

tarSubSuf" y sera de tipo entero, dentro de ella declaramos dos variables; una de tipo entero y la otra de tipo char, la primera sera un contador, y la ultima para almacenar el contenido del fichero. Abrimos el archivo de entrada para leer su contenido. Ahora realizamos un ciclo donde mientras no sea "end of file" almacenaremos el contenido del archivo dentro de la variable char declarada anteriormente, despues mediante una condicion

"Si el retorno de la función strstr() es diferente del apuntador en la primera posición de la variable cadena y si la variable cadena es subcadena de la variable cadenaTemporal" ¿Por qué no queremos que nos retorne el apuntador de la primera posición de la variable cadena? porque no que queremos es encontrar el sufixo por lo que este no se va encontrar en el inicio de la palabra, sera en otra posición de la palabra y a su vez es una subcadena de la otra.

Cerramos el archivo de entrada y abrimos el archivo de salida, mandamos a imprimir el valor del contador mediante un fprintf en el archivo de salida y por ultimo cerramos este ultimo.

8. WC - Contador de cadenas, caracteres y numero de lineas

Contador numero de lineas

```
int contarLineas(const char *entrada){
    int numeroLineas=0;
    int cLineas=0;
    char cadenaTemporal[256];
    FILE* archivoEntrada=fopen(entrada,"r");

    while(!feof(archivoEntrada)){
        fgets(cadenaTemporal,256,archivoEntrada);
        numeroLineas=numeroLineas+1;
    }

    fclose(archivoEntrada);
    return numeroLineas;
}
```

Esta es una función de tipo int puesto que retornara un valor al final Declaramos una función con el nombre "contarLineas" y sera de tipo entero, dentro de ella declaramos dos variables, una sera el contador de lineas que existen en el archivo de entrada, la otra sera de tipo char, donde almacenara el contenido del archivo. Abrimos el archivo de entrada para lectura, usamos un ciclo while donde

"Mientras no sea "end of file" del archivo de entrada" con un fgets obtendremos el contenido del archivo y lo almacenaremos en la variable cadenaTemporal. Ahora bien, una gran ventaja que tenemos al usar el fgets es que este almacenara linea por linea el contenido de archivo, por lo que al hacerlo, el contador ira en incremento puesto que asi lo tenemos señalado, y cada que realice ese ciclo hasta que se llegue a final del archivo permitirá saber cuantas lineas tenemos.

Por ultimo retornamos el numero de conteos que haya realizado el contador.

En estas funciones donde tendremos que retornar un valor tendremos que mandarlo a imprimir desde la función main() en su respectivo punto.

Contador caracteres

```
int contarCaracteres(const char *entrada){
    int numeroCaracteres=0;
    int cCaracteres=0;
    FILE* archivoEntrada=fopen(entrada,"r");

    while((cCaracteres=fgetc(archivoEntrada))!=EOF){
        if(cCaracteres!=' ' && cCaracteres!='\n' && cCaracteres!='\t' )
            numeroCaracteres=numeroCaracteres+1;
    }

    fclose(archivoEntrada);
    return numeroCaracteres;
}
```

Esta es una función de tipo int puesto que retor-

nara un valor al final. Declaramos una función con el nombre "contarCaracteres" y sera de tipo entero, dentro de ella declaramos dos variables, una sera el contador de veces encuentra un carácter en el archivo, la segunda sera de tipo char, donde almacenara el contenido del archivo. Abrimos el archivo de entrada para lectura, usamos un ciclo while donde

"Mientras el valor de cCaracteres (este tendra como contenido carácter por carácter del archivo de entrada) sea diferente del "end of file" " entonces decimos mediante una condición, "Si cCaracteres es diferente de un espacio vacío y si es diferente de salto de linea (\n) y si es diferente de tabulador (\t)" entonces realizara un conteo de los caracteres existentes. Esta condición fue determinada ya que la función fgetc() también toma como el espacio vacío, salto de linea y el tabulador como caracteres, en nuestro caso, nosotros no los tomaremos en cuenta. Por ultimo retornamos el valor del contador.

Contador cadenas

```
int contarCadenas(const char *entrada){
    int numeroCadenas=0;
    int cCadenas=0;
    char cadenaTemporal[256];
    FILE* archivoEntrada=fopen(entrada,"r");

    while(!feof(archivoEntrada)){
        fscanf(archivoEntrada,"%s",cadenaTemporal);
        numeroCadenas++;
    }

    fclose(archivoEntrada);
    return numeroCadenas;
}
```

Esta es una función de tipo int puesto que retor-

nara un valor al final. Declaramos una función con el nombre "contarCadenas" y sera de tipo entero, dentro de ella declaramos dos variables, una sera el contador de cuantas cadenas existen, la segunda sera de tipo char, donde almacenara el contenido del archivo de entrada. Abrimos el archivo de entrada para lectura, usamos un ciclo while donde

"Mientras no sea "end of file del archivo de entrada" vamos a obtener el contenido del archivo y se almacenara en la variable cadenaTemporal, después empezamos a incrementar el contador ¿por que? La función fscanf obtiene solo las palabras, los caracteres de espacio vacío y tabuladores no los tomara en en cuenta. Por ultimo retornamos el valor del contador.

Salida

```
else if(strcmp(argumento,"WC")==0){
    FILE *archivoSalida=fopen(archivoSal,"w");
    printf("El numero de caracteres encontrados es: %d\n",contarCaracteres(archivoEnt));
    printf("El numero de lineas encontradas es: %d\n",contarLineas(archivoEnt));
    printf("El numero de cadenas encontradas es: %d\n",contarCadenas(archivoEnt));
    //Aqui mandamos a imprimir en el archivo de salida la informacion
    fprintf(archivoSalida,"El numero de caracteres encontrados es: %d\n",contarCaracteres(archivoEnt));
    fprintf(archivoSalida,"El numero de lineas encontradas es: %d\n",contarLineas(archivoEnt));
    fprintf(archivoSalida,"El numero de cadenas encontradas es: %d\n",contarCadenas(archivoEnt));
    printf("Usted ingreso la opcion %s\n");
    fclose(archivoSalida);
}
```

Aqui se muestra como mandamos a llamar a cada función al main() y su respectivo else if, así mismo, abrimos el archivo de salida donde le indicamos que vamos a escribir sobre el, mandamos a imprimir mediante un fprintf llamamos la función respectiva en este mismo, puesto que retorna un valor.

9. WNUM - Contador de enteros y flotantes

Contador enteros

```
int contarEnteros(const char* entrada){
    int conteoDeEnteros=0;
    char cadenaTemporal[256];
    FILE* archivoEntrada=fopen(entrada,"r");
    while(!feof(archivoEntrada)){
        fscanf(archivoEntrada,"%s",cadenaTemporal);
        if(es_entero(cadenaTemporal))
            conteoDeEnteros++;
    }
    return conteoDeEnteros;
}
```

```
bool es_entero(const char *cadena){
    int i;
    int conteoNumeros = 0;
    int conteoPermitidos = 0;
    int conteoOtros = 0;

    for(i = 0; cadena[i] != '\0'; i++){
        if (cadena[i] >= '0' && cadena[i] <= '9')
            conteoNumeros++;
        else if (cadena[i] == '.' || cadena[i] == ',' || cadena[i] == ';')
            conteoPermitidos++;
        else
            conteoOtros++;
    }

    if(conteoOtros == 0){
        if (conteoPermitidos == 0)
            return true;
        else if ((conteoPermitidos == 1) && (cadena[i-1] == '.' || cadena[i-1] == ',' || cadena[i-1] == ';'))
            return true;
        else
            return false;
    }
    return false;
}
```

Mediante dos funciones una de tipo int y otra booleana donde una retornara un valor entero y la otra retornara un valor booleano (True o False). La función de tipo booleano se llamará `es_entero` y se encarga de determinar si un número encontrado es un entero mediante una serie de condiciones, en ella declaramos 4 variables donde todas serán contadores. Un número entero se caracteriza por:

1. No tiene caracteres.
2. No tiene punto (.) y si llegara a tener será al final del mismo número de esta forma (9.) con la cual concluimos que es punto final de un texto.
3. No tiene coma (,) y si llegara a tener será al final del mismo número de esta forma (9,) con la cual concluimos que es una pausa del texto.
4. No tiene punto y coma (;) y si llegara a tener será al final del mismo número de esta forma (9;) con la cual concluimos que es una pausa intermedia del texto.

Con una sentencia de control analizaremos cadena por cadena hasta que llegue al final de cadena que es `\0` y al reconocer los caracteres que no queremos hará un conteo de ellos, así mismo llevaremos un conteo de números que existan en la cadena. Por último mediante los conteos de los caracteres prohibidos diremos si cumple con los aquellos que hacen un número entero, entonces retorna un valor 'true', de lo contrario 'false'.

Ahora con la función `int` que se llamará `contarEnteros` declaramos dos variables, una será el contador de veces que el contador incremento de acuerdo al número de enteros existentes en el archivo de entrada, la segunda será de tipo `char`, donde almacenará el contenido del archivo, obtendremos el contenido del archivo de entrada para después realizar con la función booleana las condiciones para saber si es un entero, por lo que se le pasará la cadena donde guardamos el contenido del archivo. Abrimos el archivo de entrada para lectura, usamos un ciclo `while` donde "mientras no sea "end of file" del archivo de entrada" decimos que si "se cumple el valor de retorno de la función booleana" ya que como se mencionó la función booleana retornará un valor de `true` o `false` dependiendo de si se cumple la función, por lo que si retorna valor de `true` de acuerdo a las condiciones mencionadas el contador irá en incremento, de lo contrario no incrementará.

Contador flotantes

```
int contarFlotantes(const char* entrada){
    int numFlot=0;
    char cadenaTemporal[256];
    FILE* archivoEntrada=fopen(entrada,"r");
    while(!feof(archivoEntrada)){
        fscanf(archivoEntrada,"%s",cadenaTemporal);
        if(es_flotante(cadenaTemporal)){
            numFlot++;
        }
    }
    return numFlot;
}
```

```
bool es_flotante(const char *cadena){
    int i, conteoFlotantes=0, conteoPuntos=0, conteoPermitidos=0, conteoOtros=0;
    for(i=0; cadena[i]!='\0'; i++){
        if(cadena[i]=='0' && cadena[i]!='9')
            conteoFlotantes++;
        else if(cadena[i]!='.')
            conteoPuntos++;
        else if(cadena[i]!='.' || cadena[i]!=';')
            conteoPermitidos++;
        else
            conteoOtros++;
    }
    if(conteoOtros==0){
        if(conteoPuntos==1 && conteoPermitidos==0)
            return true;
        else if ((conteoPuntos==1) && (conteoPermitidos==1 && (cadena[i-1]!='.' || cadena[i-1]!=';'))))
            return true;
        else if ((conteoPuntos==2) && (conteoPermitidos==0 && cadena[i-1] == '.'))
            return true;
        else
            return false;
    }
    else
        return false;
}
```

Mediante dos funciones una de tipo int y otra booleana donde una retornara un valor entero y la otra retornara un valor booleano (True o False). La función de tipo booleano se llamara es _ flotante y se encargara de determinar si un numero encontrado es un entero mediante una serie de condiciones en ella declaramos 4 variables donde todas serán contadores. Un numero flotante se caracteriza por:

1. No tiene caracteres.
2. Si tiene punto (.)
3. Si tiene dos puntos, uno tendrá que estar forzosamente al final de este, de esta forma (9.98.) con la cual concluimos que es punto final de un texto.
4. No tiene coma (,) y si llegara a tener sera al final del mismo numero de esta forma (9.98,) con la cual concluimos que es una pausa del texto.
5. No tiene punto y coma (;) y si llegara a tener sera al final del mismo numero de esta forma (9.98;) con la cual concluimos que es una pausa intermedia del texto.

Con una sentencia de control analizaremos cadena por cadena hasta que llegue al final de cadena que es \0 y al reconocer los caracteres que no queremos hará un conteo de ellos, así mismo llevaremos un conteo de numero que existan en la cadena. Por ultimo mediante los conteo de los caracteres prohibidos diremos si cumple con los aquellos que hacen un numero flotante, entonces retorna un valor 'true', de lo contrario 'false'. Ahora con la función int que se llamara contar-Flotantes declaramos dos variables, una sera el contador de veces que el contador incremento de acuerdo al numero de enteros existentes en el archivo de entrada, la segunda sera de tipo char, donde almacenara el contenido del archivo, obtendremos el contenido del archivo de entrada para después realizar con la función booleana las condiciones para saber si es un entero, por lo que se le pasara la cadena donde guardamos el contenido del archivo. Abrimos el archivo de entrada para lectura, usamos un ciclo while donde "mientras no sea "end of file" del archivo de entrada" decimos que si "se cumple el valor de retorno de la función booleana" ya que como se menciono la función booleana retornara un valor de true o false dependiendo se si se cumple la funcion, por lo que si retorna valor de true de acuerdo a las condiciones mencionadas el contador ire en incremento, de lo contrario no incrementara.

Salida

```
else if(strcmp(argumento,"NUM")!=0){
    FILE *archivoSalida=fopen(archivoSal,"w");
    printf("El numero de enteros que existen es: %d\n",contarEnteros(archivoEnt));
    printf("El numero de flotantes que existen es: %d\n",contarFlotantes(archivoEnt));
    printf("Usted ingreso la opcion %d");
    //Aqui mandamos a imprimir en el archivo de salida la informacion
    fprintf(archivoSalida,"El numero de enteros que existen es: %d\n",contarEnteros(archivoEnt));
    fprintf(archivoSalida,"El numero de flotantes que existen es: %d\n",contarFlotantes(archivoEnt));
    fclose(archivoSalida);
}
```

Ahora mandamos a llamar las funciones de tipo int donde las vamos a andar a imprimir mediante un fprintf puesto que estas retornan un valor.

10. INC - Imprimir lineas de un archivo con su numero de linea.

Esta es una función de tipo void por lo que la función no retornara nada.

```
void imprimirLinea(const char *entrada, const char *salida){
    int i=1;
    char lineas[256];
    FILE* archivoEntrada=fopen(entrada,"r");
    FILE* archivoSalida=fopen(salida,"w");

    while(!feof(archivoEntrada)){
        fgets(lineas,256,archivoEntrada);
        fprintf(archivoSalida,"%d - %s\n",i,lineas);
        printf("%d-%s\n",i,lineas);
        i++;
    }
    fclose(archivoEntrada);
    fclose(archivoSalida);
}
```

Declaramos una función con el nombre "imprimirLineas" y sera de tipo entero, dentro de ella declaramos dos variables; una sera de tipo int y llevara el conteo de numero de linea, la otra sera de tipo char, almacenara el contenido del archivo. Abrimos el archivo de entrada y abrimos el archivo de salida; en el primero es lectura, el ultimo escribiremos lo requerido.

Con un ciclo while decimos que "mientras no sea "end of file" del archivo de entrada" entonces obtenemos el contenido del archivo linea por linea con fgets y lo almacenaremos en la variable declarada anteriormente. Mandamos a imprimir cada linea y realizamos un conteo hasta que lleguemos al final del archivo. Por ultimo cerramos el archivo de entrada y salida.

Manejo de los primeros 4 puntos.

Ahora hablaremos de los puntos 1 al 4 donde en la parte anterior se saltaron, puesto que esta es la estructura que sigue el programa. A continuación de explicara el porque.

De acuerdo a los requerimientos del programa, del punto 1 al 4 pide ingresar un argumento seguido de un guion bajo y un numero, de acuerdo a la preferencia del usuario, por ejemplo, esta seria un argumento referente al punto 1 L_ 1_2 _3, significa que el usuario eligió la opcion 1 que es L, reconoceremos los caracteres que estan antes del primer guion bajo como "Comando" y desea manejar las lineas 1, 2 y 3 del archivo de enterada.

Para identificar el comando y realice una determinada acción haremos uso de sscanf donde vamos extraer del argumento que va escribir el usuario el comando con "%[^_]" decimos que solo obtenga todo lo que este antes del primer _ donde se almacenara en la variable "comando". asi mismo vamos a obtener lo que sigue hacia la derecha del primer _ donde eso lo vamos a almacenar en la misma variable "argumento", por lo que al final de la función sscanf el valor de argumento

```
int main(int argc, char *argv[]) {  
  
    const char *archivoEnt = argv[1];  
    const char *archivoSal = argv[argc - 1];  
    char argumento[256];  
    char comando[256];  
    strcpy(argumento, argv[2]);
```

```
else {  
    // Obtenemos el comando y los valores dados puesto que se trata de un atributo tipo comando_valor1_valor2.....  
    sscanf(argumento, "%[^_]%s", comando, argumento); // obtenemos comando y argumento se queda con el resto  
    bool tablaDeValores[1000] = {false};  
    int valor;  
    char caracter;  
    int k=0;  
    do{  
        valor = 0;  
        while (('0' <= argumento[k] && (argumento[k] <= '9')){ // Ciclamos la cadena 'argumento' mientras sea numero  
            if (valor==0) // numero nuevo: lo copiamos  
                valor = argumento[k] - '0';  
            else // subsecuente digito: recorremos un lugar decimal el que llevabamos y lo metemos  
                valor = valor * 10 + (argumento[k] - '0');  
            k++;  
        } // encuentra algo distinto y termina el ciclo  
        tablaDeValores[valor - 1] = true; // asignamos TRUE a la tabla de valores justo donde nos indico el usuario en sus argumentos  
    } while (argumento[k++] != '_'); // mientras aun haya valores por leer del usuario
```

En la imagen de la izquierda tenemos las variables del int main() y son: char *archivoEnt = argv[1], char *archivoSal = argv[argc - 1], char argumento[256], char comando[256]; Las dos primeras variables que son punteros serán para representar los archivos de entrada y salida, las ultimas dos serán para reconocer argumento y comando como ya lo mencionamos anteriormente, así que almacenaran su contenido. De primeras diremos que argumento sera lo escrito por el usuario, así que se copia el contenido de 'argv[2]' a la variable 'argumento'.

En la imagen de la derecha es el código con el que obtendremos el argumento y comando así mismo empezamos el desarrollo de los 4 puntos faltantes. Comenzamos con las variables a utilizar y serán 4; bool tablaDeValores[1000] = false, int valor, char carácter, int k=0. El propósito de la variable bool es para construir una tabla con valores false, 'valor' sera un contador que va recorrer dicha tabla y 'k' también actuara como un contador que va a recorrer la cadena original.

Así que mediante un ciclo do-while vamos a recorrer el ultimo valor que le dimos a la variable "argumento", recordemos que tiene almacenado del primer _ hacia la derecha. Al recorrer argumento mediante ciclos y sean números los que vaya encontrado los almacenara en la variable 'valor' por medio de una resta de ascii, y si el numero llegara a ser de dos dígitos este hará una operación donde permita convertirlo a un numero de dos digitos.

Ahora ese valor que vaya tomando la variable valor lo ira almacenando en la tabla de valores, donde le diremos que en, la casilla donde nosotros tenemos como valor le asigne el valor de 'true' de esta forma podremos manipular cuando el usuario indique un cierto numero de linea, parrafo etc.

1. L_ Imprimir líneas indicadas

```
void imprimir_lineas_seleccionadas(const char *entrada, bool tablaDeRegiones[], const char *salida){
    char texto[256];
    // Si el valor de nuestra tabla de líneas a imprimir es 'true' la imprimimos del original, sino, solo imprimimos línea en blanco
    FILE *archivoEntrada = fopen(entrada, "r");
    FILE *archivoSalida = fopen(salida, "w");
    int i = 0;
    while (fgets(texto, 256, archivoEntrada) != NULL){
        if (tablaDeRegiones[i] == true){
            printf("%s", texto);
            fprintf(archivoSalida, "%s", texto);
        }
        else{
            printf("\n");
            fprintf(archivoSalida, "\n");
        }
        i++;
    }
    fclose(archivoEntrada);
    fclose(archivoSalida);
}
```

Declaramos una función de tipo void llamada 'imprimir_lineas_seleccionadas' donde no va-

Ahora diremos que "mientras que sea diferente de null el obtener el contenido de el archivo en la variable texto" observemos que haremos uso de fgets donde obtendremos línea por línea mientras se realice el ciclo. procedemos a decir que "si en la tabla de valores (esta se ira recorriendo por el contador) existe el valor de verdad, que en ese numero de línea la imprima en el archivo de salida, de no ser asi no imprimirá nada y solo serán saltos de línea.

Por ultimo cerramos los dos archivos.

2. C_ Imprimir columnas indicadas

```
void imprimir_columnas_seleccionadas(const char *entrada, bool tablaDeRegiones[], const char *salida){
    char texto[256];
    int i;
    FILE *archivoEntrada = fopen(entrada, "r");
    FILE *archivoSalida = fopen(salida, "w");
    while (fgets(texto, 256, archivoEntrada) != NULL) { // ciclamos cada línea
        i = 0;
        while (texto[i] != '\0') { // mientras no llegue al final de la línea
            if (tablaDeRegiones[i] == true) { // si el usuario nos indico esa línea para imprimir
                printf("%c", texto[i]); // imprimimos el caracter
                fprintf(archivoSalida, "%c", texto[i]);
            }
            else{
                printf(" "); // espacio en blanco
                fprintf(archivoSalida, " ");
            }
            i++;
        }
        printf("\n");
        fprintf(archivoSalida, "\n");
    }
    fclose(archivoEntrada);
    fclose(archivoSalida);
}
```

mos a retornar nada, dentro de ella declaramos dos variables una de tipo char; almacenaremos el contenido del archivo en ella y la otra de tipo int que sera encargada de ser contador.

Abriremos los archivos de entrada y salida, uno sera de lectura y el otro sera donde escribiremos.

Declaramos una función de tipo void llamada 'imprimir_columnas_seleccionadas' donde no vamos a retornar nada, dentro de ella declaramos dos variables una de tipo char; almacenaremos el contenido del archivo en ella y la otra de tipo int que sera encargada de ser contador.

Abriremos los archivos de entrada y salida, uno sera de lectura y el otro sera donde escribiremos.

Ahora diremos que "mientras que sea diferente de null el obtener el contenido de el archivo en la variable texto" observemos que haremos uso de fgets donde obtendremos línea por línea mientras se realice el ciclo, utilizaremos otro ciclo while para recorrer cada cadena y "mientras no llegue a final de cadena" "si en la tabla de valores (esta se ira recorriendo por el contador) existe el valor de verdad, que en ese caracter lo imprima en el archivo de salida, de no ser asi no imprimira nada Por ultimo cerramos los dos archivos.

3. ILV_ Imprimir salto de líneas en las líneas indicadas

```
void insertar_saltos(const char *entrada, bool tablaDeRegiones[], const char *salida){
    char texto[256];
    // Si el valor de nuestra tabla de líneas a imprimir es 'true' la imprimimos del original, sino, solo imprimimos línea en blanco
    FILE *archivoEntrada = fopen(entrada, "r");
    FILE *archivoSalida = fopen(salida, "w");
    int i = 0;
    while (fgets(texto, 256, archivoEntrada) != NULL){
        if (tablaDeRegiones[i] == true){
            printf("%s\n", texto);
            fprintf(archivoSalida, "%s\n", texto);
        }
        else{
            printf("%s", texto);
            fprintf(archivoSalida, "%s", texto);
        }
        i++;
    }
    fclose(archivoEntrada);
    fclose(archivoSalida);
}
```

Declaramos una función de tipo void llamada 'insertar_saltos' donde no vamos a retornar nada, dentro de ella declaramos dos variables una de tipo char; almacenaremos el contenido del archivo en ella y la otra de tipo int que sera encargada de ser contador.

Abriremos los archivos de entrada y salida, uno sera de lectura y el otro sera donde escribiremos.

Ahora diremos que "mientras que sea diferente de null el obtener el contenido de el archivo en la variable texto" observemos que haremos uso de fgets donde obtendremos linea por linea mientras se realice el ciclo. Procedemos a decir que "si en la tabla de valores (esta se ira recorriendo por el contador) existe el valor de verdad, que en ese numero de linea la imprima junto con un salto de linea en el archivo de salida, de no ser asi, solo imprimirá las lineas sin alto de linea. Por ultimo cerramos los dos archivos.

4. IC_ Imprimir tabulador en las columnas indicadas

```
void insertar_tabuladores(const char *entrada, bool tablaDeRenglones[], const char *salida){
    char texto[256];
    int i;

    FILE *archivoEntrada = fopen(entrada, "r");
    FILE *archivoSalida = fopen(salida, "w");
    while (fgets(texto, 256, archivoEntrada) != NULL) { // ciclamos cada linea
        i = 0;
        while (texto[i] != '\0') { // mientras no llegue al final de la linea
            if (tablaDeRenglones[i] == true) { // si el usuario nos indico esa linea para imprimir
                printf("%c\t", texto[i]); // imprimimos el caracter con tabulador
                fprintf(archivoSalida, "%c\t", texto[i]);
            }
            else{
                printf("%c", texto[i]); // solo imprimimos el caracter
                fprintf(archivoSalida, "%c", texto[i]);
            }
            i++;
        }
        printf("\n");
        fprintf(archivoSalida, "\n");
    }
    fclose(archivoEntrada);
    fclose(archivoSalida);
}
```

Declaramos una función de tipo void llamada 'insertar_tabulador' donde no vamos a retornar nada, dentro de ella declaramos dos variables una de tipo char; almacenaremos el contenido del archivo en ella y la otra de tipo int que sera encargada de ser contador.

Abriremos los archivos de entrada y salida, uno sera de lectura y el otro sera donde escribiremos.

Ahora diremos que "mientras que sea diferente de null el obtener el contenido de el archivo en la variable texto" observemos que haremos uso de fgets donde obtendremos linea por linea mientras se realice el ciclo. Procedemos a decir que "si en la tabla de valores (esta se ira recorriendo por el contador) existe el valor de verdad, que en ese numero de carácter con un tabulador la imprima en el archivo de salida, de no ser así no imprimirá los demás caracteres de manera normal Por ultimo cerramos los dos archivos.