



UNIVERSIDAD NACIONAL DE SAN AGUSTIN

ESCUELA PROFESIONAL DE CIENCIA DE LA COMPUTACION

COMPILADORES

GRUPO A

Laboratorio 01

Docente:

Dr. Yuber Velazco Paredes

Realizado por:

Pinto Medina, Brian

9 de mayo de 2020

1. Objetivo

Familiarizarse con la entrada y salida de datos ya sea por consola y/o por archivos de texto plano, como primera etapa en la construcción de un compilador.

2. Desarrollo:

Implementar en el lenguaje de programación C o C++ los siguientes programas:

2.1. Leer toda una instrucción por consola (Ejemplo: “int temp;”) y mostrar en pantalla letra por letra

```
1  //////////////////////////////////////
2  //                                EJERCICIO #1                                //
3  //////////////////////////////////////
4
5  #include <iostream>
6  #include <string>
7  using namespace std;
8
9  int main() {
10     string chain;
11     cout << "Ingrese una cadena:\n";
12     getline( std::cin, chain );
13
14     for( int i = 0; i < chain.size(); ++i )
15         cout << chain[ i ] << endl;
16     return 0;
17 }
```

```
pimed@horo:~/Documents/ADA$ g++ exercisel.cpp
pimed@horo:~/Documents/ADA$ ./a.out
Ingrese una cadena:
int temp;
i
n
t

t
e
m
p
;
```

2.2. Leer toda una instrucción por consola (Ejemplo: “int temp;”) y mostrar en pantalla letra por letra

```
1  //////////////////////////////////////
2  //                                EJERCICIO #2                                //
3  //////////////////////////////////////
4
5  /* Para este ejercicio utilizaremos un archivo de
6     texto plano llamado test1.txt el cual sera in_
7     cluido.
8  */
9
10 #include <stdio.h>
11 #include <stdlib.h>
12 using namespace std;
13
14 int main() {
15     FILE *file = fopen( "test1.txt", "r" );
16     char character;
17     while(( character = fgetc( file ) ) != EOF )
18         printf("%c \n", character );
19     return 0;
20 }
```

```
pimed@horo:~/Documents/ADA$ g++ exercise2.cpp
pimed@horo:~/Documents/ADA$ ./a.out
A
r
c
h
i
v
o

d
e

p
r
u
e
b
a
```

2.3. Crear un programa que cifre un pseudocódigo y otro que lo descifre, según la técnica del cifrado de César. El pseudocódigo se encuentra en un archivo de texto plano. El cifrado de César consiste en mover cada letra un determinado número de espacios en el alfabeto (puede ser 3 espacios)

```
1  //////////////////////////////////////
2  //                                EJERCICIO #3                                //
3  //////////////////////////////////////
4
5  /* Para este ejercicio se crearon dos funciones el usua_
6     debera ingresar el nombre del archivo que desea en_
7     criptar y el desplazamiento deseado para la encryp_
8     tacion de Cesar.
9     El programa creara dos archivos uno con el texto
10    encriptado y uno con el texto desencriptado, los ar_
11    chivos creados es llaman "Encrypt.txt" y
12    "Decrypt.txt".
13 */
14
15 #include <stdio.h>
16 #include <stdlib.h>
17 using namespace std;
18
19 void encryptionCesar( char *fname, int offset ) {
20     FILE *fileRead, *fileWritten;
21     fileRead = fopen( fname, "r" );
22     fileWritten = fopen( "Encrypt.txt", "wt");
23     char character;
24
25     while(( character = fgetc( fileRead )) != EOF ) {
26         character = character + offset;
27         fputc( character, fileWritten );
28     }
29     fclose( fileRead );
30     fclose( fileWritten );
31 }
32
33 void decryptionCesar( char *fname, int offset ) {
34     FILE *fileRead, *fileWritten;
35     fileRead = fopen( fname, "r" );
36     fileWritten = fopen( "Decrypt.txt", "wt");
```

```
37     char character;  
38  
39     while(( character = fgetc( fileRead )) != EOF ) {  
40         character = character - offset;  
41         fputc( character, fileWritten );  
42     }  
43     fclose( fileRead );  
44     fclose( fileWritten );  
45 }  
46  
47 int main() {  
48     int offset;  
49     char fname[ 30 ];  
50     char fencryptthed[] = "Encryptth.txt";  
51  
52     printf( "Ingrese el nombre del archivo a codificar:\n" );  
53     scanf( "%30[^\n]", fname );  
54     printf( "Ingrese el desplazamiento deseado:\n" );  
55     scanf( "%i", &offset );  
56     encryptionCesar( fname, offset );  
57     decryptionCesar( fencryptthed, offset );  
58  
59     return 0;  
60 }
```

```
pimed@horo:~/Documents/ADA$ g++ exercise3.cpp  
pimed@horo:~/Documents/ADA$ ./a.out  
Ingrese el nombre del archivo a codificar:  
test2.txt  
Ingrese el desplazamiento deseado:  
3
```

```
≡ test2.txt ×  
≡ test2.txt  
1  #include <stdio.h>  
2  
3  int main() {  
4      printf("Hello World");  
5      return 0;  
6  }
```

```

≡ Encrypth.txt X
≡ Encrypth.txt
1  &lqfoxgh#?vwglr1kA
2
3  lqw#pdlq+,#~
4  #####sulqwi+%Koor#Zruog%,>
5  #####uhwxuq#3>
6  ?

```

```

≡ Decrypth.txt X
≡ Decrypth.txt
1  #include <stdio.h>
2
3  int main() {
4      printf("Hello World");
5      return 0;
6  }

```

2.4. Leer un archivo de texto plano, letra por letra hasta encontrar un separador (espacio, tabulador o salto de línea) y mostrar en pantalla si el vocablo se trata de un número entero, una palabra o un caracter especial (“+”, “-”, “*”, “/”, “.”)

```

1  //////////////////////////////////////
2  //                               EJERCICIO #4                               //
3  //////////////////////////////////////
4
5  /* Para el ejercicio se utilizo el archivo test3.txt
6     se realizaron las limitaciones, tambien se tomo en
7     cuenta que el vocablo encontrado puede tener nume_
8     ros, palabras o caracteres especiales al mismo tiem_
9     po.
10 */
11
12 #include <stdio.h>
13 #include <stdlib.h>
14 using namespace std;

```

```
15
16 int main() {
17     FILE *file = fopen( "test.txt", "r" );
18     char character;
19     bool number = 0, word = 0, special = 0, unknown = 0;
20     while(( character = fgetc( file )) != EOF ) {
21         printf( "%c \n", character );
22         if( character == ' ' || character == '\t' ||
23             character == ' ' )
24             break;
25
26         if( character >= 48 && character <= 57 ) {
27             number = 1;
28         } else if( character >= 97 && character <= 122 ) {
29             word = 1;
30         } else if( character >= 65 && character <= 90 ) {
31             word = 1;
32         } else if( character == '+' || character == '-' ||
33                     character == '/' || character == '*' ) {
34             special = 1;
35         } else {
36             unknown = 1;
37         }
38
39         printf("El archivo contiene un/una ");
40
41         if( number )
42             printf("numero ");
43         if( word )
44             printf("palabra ");
45         if( special )
46             printf("caracter especial ");
47         if( unknown )
48             printf("caracter desconocido ");
49
50         printf("\n");
51         return 0;
52     }
```

Cuando el archivo contiene solamente palabra.

```
test3.txt x
test3.txt
1  Hola como estas
```

```
pimed@horo:~/Documents/ADA$ g++ exercise4.cpp
pimed@horo:~/Documents/ADA$ ./a.out
H
o
l
a

El archivo contiene un/una palabra
```

Cuando el archivo contiene solamente numeros.

```
test3.txt x
test3.txt
1  123 456 789
```

```
pimed@horo:~/Documents/ADA$ g++ exercise4.cpp
pimed@horo:~/Documents/ADA$ ./a.out
1
2
3

El archivo contiene un/una numero
```


Cuando el archivo contiene un caracter especial.

```
test3.txt x
test3.txt
1 Hola1234 como

pimed@horo:~/Documents/ADA$ g++ exercise4.cpp
pimed@horo:~/Documents/ADA$ ./a.out
H
o
l
a
1
2
3
4
El archivo contiene un/una numero palabra
```

Cuando el archivo contiene una palabra y un numero.

```
test3.txt x
test3.txt
1 Hola1234 como

pimed@horo:~/Documents/ADA$ g++ exercise4.cpp
pimed@horo:~/Documents/ADA$ ./a.out
H
o
l
a
1
2
3
4
El archivo contiene un/una numero palabra
```