



**INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO**

TEORÍA COMPUTACIONAL

2CM4

PROFESOR: LUZ MARÍA SÁNCHEZ GARCÍA

PRÁCTICA 1 OPERACIONES CON CADENAS

VÁZQUEZ MORENO MARCOS OSWALDO

2016601777



FECHA DE ENTREGA: 1 DE MARZO DE 2018

INTRODUCCIÓN

En la siguiente práctica se pretende comprender algunas de las operaciones dentro de la teoría computacional, las cuales se han visto en clase y a su vez se han practicado con ejercicios complementarios, en la práctica se pretende aceptar dos cadenas de caracteres con un lenguaje: {a-z} para así formar palabras, a su vez la unión de varias palabras crea lenguajes, para fines de esta práctica se denotarán a continuación las 8 operaciones que estarán dentro de la misma:

- ✓ Palíndromo.
- ✓ Longitud.
- ✓ Concatenación.
- ✓ Potencia.
- ✓ Inversa.
- ✓ Prefijos.
- ✓ Sufijos.
- ✓ Sub cadenas.

PLANTEAMIENTO DEL PROBLEMA

Implementar el algoritmo de codificación de un programa que calcule las siguientes operaciones:

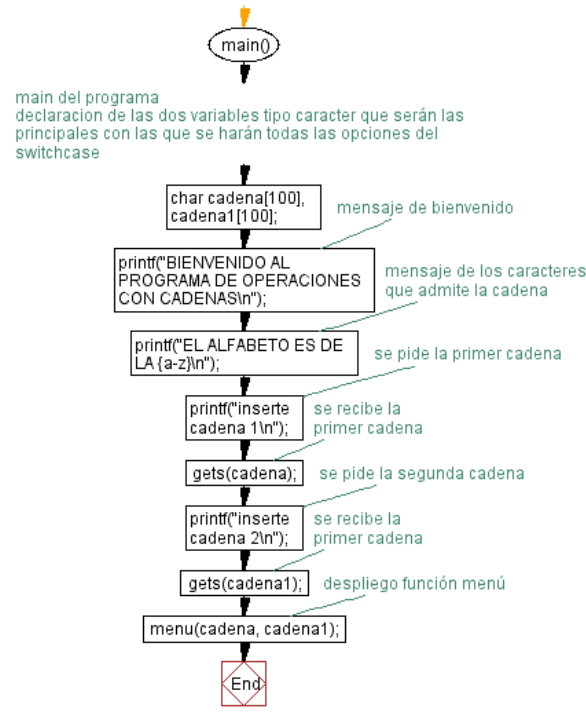
- ❖ Palíndromo.
- ❖ Longitud.
- ❖ Concatenación.
- ❖ Potencia.
- ❖ Inversa.
- ❖ Prefijos.
- ❖ Sufijos.
- ❖ Sub cadenas.

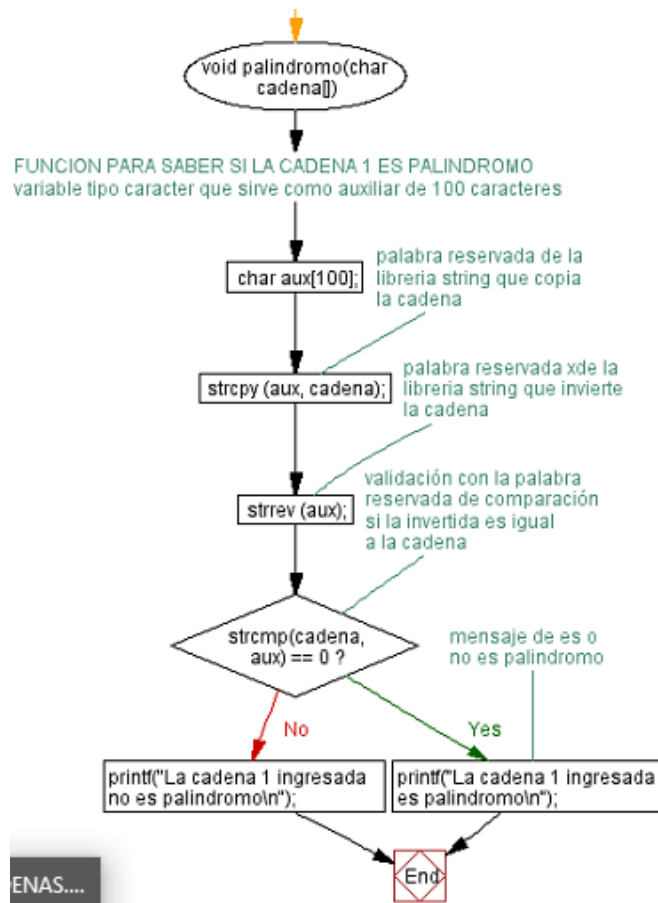
Todo esto con un determinado alfabeto (trabajando mínimo con dos cadenas), en donde el usuario va a seleccionar con cuáles cadenas va a trabajar y qué tipo de operación va a seleccionar.

DISEÑO Y FUNCIONAMIENTO DE LA SOLUCIÓN

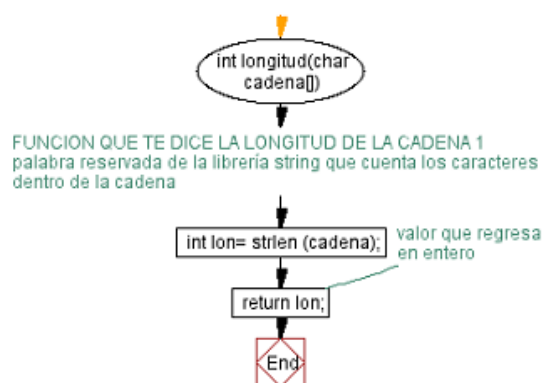
Una vez con la problemática planteada se necesitaba de poder darle sentido a la solución y es por eso por lo que se decidió crear un programa desarrollado en lenguaje C utilizando la librería <string.h> la cual iba a ser de gran ayuda puesto que en todas las operaciones se utilizaría, incluso en funciones las cuales son complementarias a las 8 opciones de operandos.

El algoritmo que se llevó a cabo consta de pedir y aceptar dos cadenas que estén dentro del lenguaje de {a-z} para posteriormente desplegar un menú que te indica las operaciones que puedes realizar, las cuales seleccionan digitando el número de la operación.

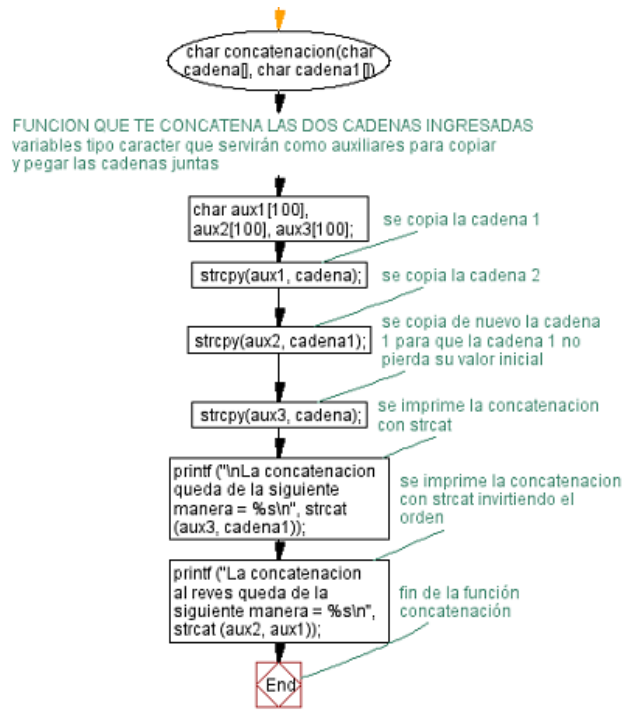




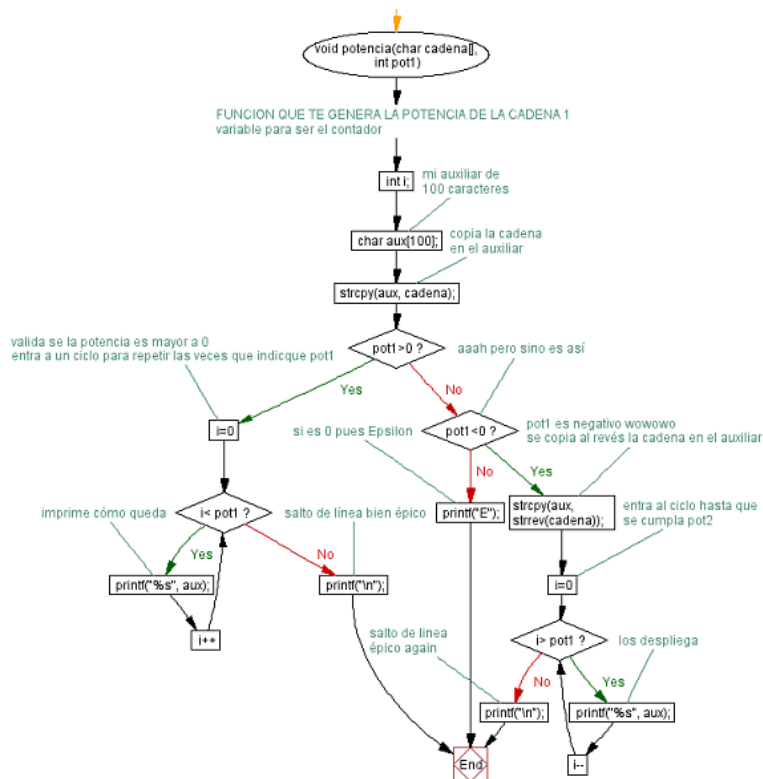
2.- Si se elige la opción de **longitud** se realiza un conteo con la palabra reservada *strlen* la cual contará los caracteres ingresados en ambas cadenas para posteriormente imprimirlas.



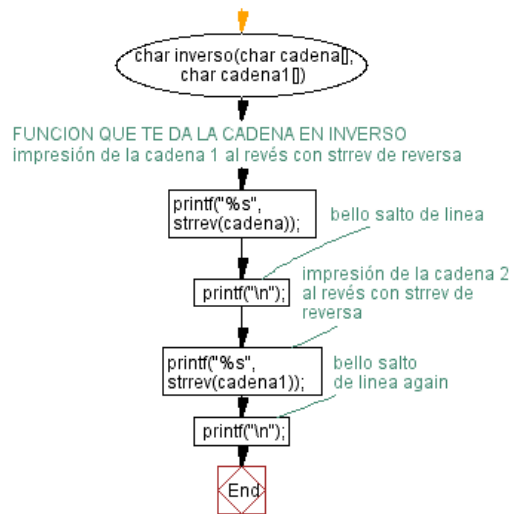
3.- Si se elige la opción de **concatenación** se realiza un copiado de ambas cadenas en dos variables auxiliares las cuales se van a imprimir juntas.



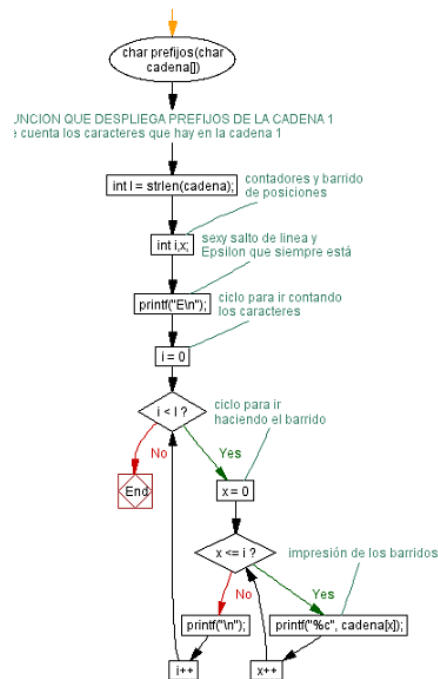
4.- Si se elige la opción de **potencia** se llama a la función potencia y se repite dicha cadena las veces que el usuario lo desee.



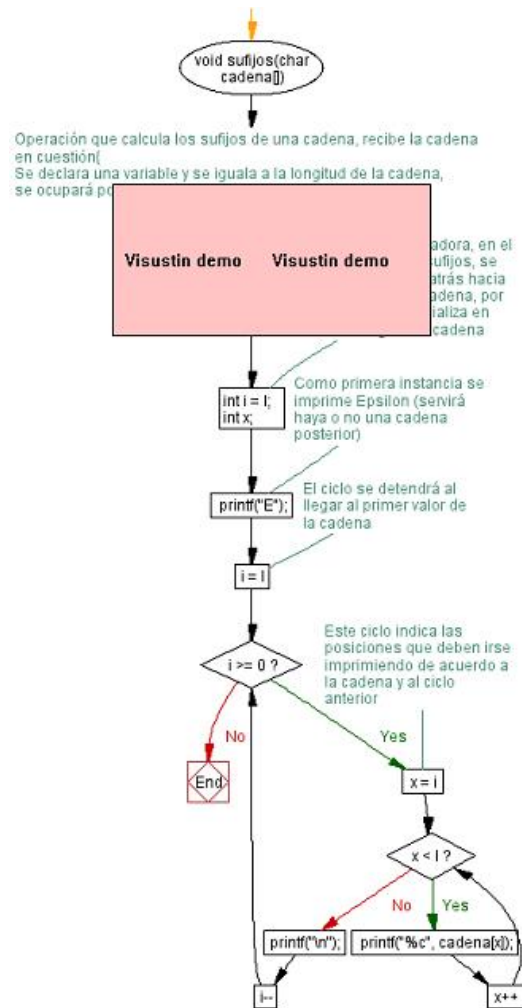
5.- Si se elige la opción de **inverso** se llama a la función inverso la cuál va a invertir la cadena, la va a copiar en otra variable auxiliar y la va a imprimir.



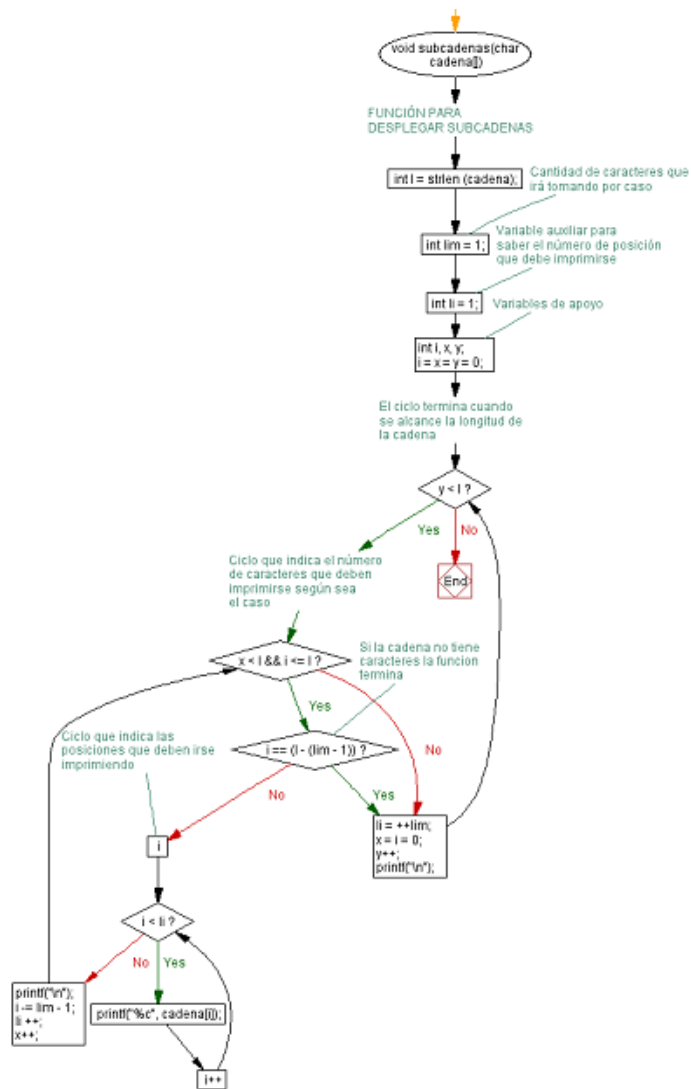
6.- Si se elige la opción de **prefijos** va generando prefijos de la cadena, por lo tanto, entra a un ciclo el cuál los va imprimiendo en orden ascendente.



7.- Si se elige la opción de **sufijos** va generando sufijos de la cadena, por lo tanto, entra a un ciclo el cuál va imprimiendo en orden descendente.



8.- Si se elige la opción de **sub cadenas** se van generando las sub cadenas de caracteres, primero el 1 con el 2, el 2 con el 3, el 3 con el 4, el 4 con el 5 y así sucesivamente.



IMPLEMENTACIÓN DE LA SOLUCIÓN

```

1  /* Hecho por Vázquez Moreno Marcos Oswaldo Boleta 2016601777
2     El 22 de Febrero de 2018
3     Materia Teoría Computacional
4     Práctica número 1 OPERACIONES CON CADENA
5     Fecha de inicio 21 de Febrero
6     Fecha de terminación 26 de Febrero
7  */
8
9  //Librerías
10 #include <stdlib.h>
11 #include <string.h>
12 #include <stdio.h>
13 #include <conio.h>
14
15 void menu(char cadena[], char cadena1[]); //funcion que despliega el menú de todas las operaciones posibles
16 void palindromo(char cadena[]); //funcion que te dice si es palindromo la cadena 1 o no
17 void palindromo1(char cadena1[]); //funcion que te dice si es palindromo la cadena 2 o no
18 void repetirmenu(char cadena[], char cadena1[]); //funcion que generé para volver a llamar a menú después de un proceso de elección
19 int longitud(char cadena[]); //función que te dice la longitud de la cadena ingresada
20 char concatenacion(char cadena[], char cadena1[]); //función que concatena las dos cadenas ingresadas
21 void potencia(char cadena[], int pot2); //funcion que te da la opción de elegir a que potencia elevar la cadena 1
22 void potencial(char cadena1[], int pot2); //funcion que te da la opción de elegir a que potencia elevar la cadena 2
23 char inverso(char cadena[], char cadena1[]); //función que te regresa la cadena en su inverso
24 char prefijos(char cadena[]); //función que te genera los prefijos de la cadena 1
25 char prefijos1(char cadena1[]); //función que te genera los prefijos de la cadena 2
26 void sufijos(char cadena[]); //función que te genera el sufijo de la cadena 1
27 void sufijos1(char cadena1[]); //función que te genera el sufijo de la cadena 2
28 void subcadenas(char cadena[]); //función que te genera la subcadena de la cadena 1
29 void subcadenas1(char cadena1[]); //función que te genera la subcadena de la cadena 1
30
31 main() { //main del programa
32     char cadena[100], cadena1[100]; //declaracion de las dos variables tipo caracter que serán las principales con las que se hará
33     printf("BIENVENIDO AL PROGRAMA DE OPERACIONES CON CADENAS\n"); //mensaje de bienvenida
34     printf("EL ALFABETO ES DE LA {a-z}\n"); //mensaje de los caracteres que admite la cadena
35     printf("inserte cadena 1\n"); //se pide la primera cadena
36     gets(cadena); //se recibe la primera cadena
37     printf("inserte cadena 2\n"); //se pide la segunda cadena
38     gets(cadena1); //se recibe la segunda cadena
39
40     menu(cadena, cadena1); //despliega función menú
41 }
42
43 void menu(char cadena[], char cadena1[]) { //función menú
44     int r=0, o, pot1, pot2; //variables para el switch, elección del menú, potencia 1 y potencia 2
45     char v; //variable para repetir o no el proceso
46     do { //comienzo del do while
47         printf("ELIJA EL NUMERO DE LA OPCION DEL PROCESO QUE QUIERE REALIZAR\n"); // mensaje
48         //desplegado de las opciones a elegir
49         printf("1.- PALINDROMO\n 2.- LONGITUD\n 3.- CONCATENACION\n 4.- POTENCIA\n 5.- INVERSO\n 6.- PREFIJOS\n 7.- SUFIJOS\n 8.- SUBC");
50         scanf("%d", &o); //recibe la opción elegida
51         fflush(stdin); //limpeza de buffer
52         switch(o) //empieza el switch dependiendo de lo elegido
53         {
54             case 1: //PALINDROMO
55                 palindromo(cadena); //Llamo a función palindromo el cual recibe la primera cadena
56                 palindromo1(cadena1); //Llamo a función palindromo el cual recibe la segunda cadena
57                 repetirmenu(cadena, cadena1); //Llamo a función repetir el menú y recibe cadena 1 y cadena 2 de nuevo
58                 break; //fin del caso 1
59             case 2: // LONGITUD
60                 printf("La longitud de la cadena 1 es: %d\n", longitud(cadena)); //imprime la longitud de la cadena 1 y ahí mismo llamo a la fi
61                 printf("La longitud de la cadena 2 es: %d\n", longitud(cadena1)); //imprime la longitud de la cadena 2 y ahí mismo llamo a la fi
62                 repetirmenu(cadena, cadena1); //Llamo a función repetir el menú y recibe cadena 1 y cadena 2 de nuevo
63                 break; //fin del caso 2
64         }
65     } while(r==0);
66 }

```

```

66     case 3:      //CONCATENACION
67         concatenacion(cadena, cadena1); //Llamo a la función concatenacion y recibe a la cadea 1 y 2
68         printf("\n"); //salto de línea bien guapo
69         repetirmenu(cadena, cadena1); //Llamo a función repetir el menu y recibe cadena 1 y cadena 2 de nuevo
70         break; //fin del caso 3
71
72
73
74     case 4:      //POTENCIA
75
76         printf("Dame la potencia para la cadena 1\n"); //pido qué potencia quiere para la cadena 1
77         scanf("%d", &pot1); //recibe la potencia
78         printf("Dame la potencia para la cadena 2\n"); //pido qué potencia quiere para la cadena 2
79         scanf("%d", &pot2); //recibe la potencia
80         potencia(cadena, pot1); //Llamo a función potencia
81         potencial(cadena1, pot2); //Llamo a función potencia 1
82         repetirmenu(cadena, cadena1); //Llamo a función repetir el menu y recibe cadena 1 y cadena 2 de nuevo
83
84         break; //fin del caso 4
85
86
87
88     case 5:      //INVERSO
89         inverso(cadena, cadena1); //funcion que invierte la cadena 1 y 2
90         repetirmenu(strrev(cadena), strrev(cadena1)); //Llamo a función repetir el menu y recibe cadena 1 y cadena 2 de nuevo pero inversas ,
91                                     //elige primero usar la opcion 4
92         break; //fin del caso 5
93
94
95
96     case 6:      //PREFIXOS
97         prefijos(cadena); //Llamo a la función prefijo
98         prefijos(cadena1); //Llamo a la función prefijo para la cadena número 2
99         repetirmenu(cadena, cadena1); //Llamo a función repetir el menu y recibe cadena 1 y cadena 2 de nuevo
100        break; //fin del caso 6
101
102
103
104     case 7:      //SUFIJOS
105         sufijos(cadena); //Llamo a la función sufijo de la cadena 1
106         sufijos(cadena1); //Llamo a la función sufijo de la cadena 2
107         repetirmenu(cadena, cadena1); //Llamo a función repetir el menu y recibe cadena 1 y cadena 2 de nuevo
108
109         break; //fin del caso 7
110
111
112     case 8:      //SUBCADENAS
113         subcadenas(cadena); //Llamo a la función subcadenas de la cadena 1
114         subcadenas1(cadena1); //Llamo a la función subcadenas de la cadena 2
115         repetirmenu(cadena, cadena1); //Llamo a función repetir el menu y recibe cadena 1 y cadena 2 de nuevo
116         break; //fin del caso 8
117     }
118 }while (r!=0); //Para que no salga del while
119 };
120
121
122 void repetirmenu(char cadena[], char cadena1[]) { //FUNCION REPETIR EL MENU
123     char v; //recibe valor para repetir el proceso
124     printf("LE GUSTARIA REPETIR EL PROCESO (Y/N)\n"); //mensaje
125     fflush(stdin); //limpieza del buffer
126     scanf("%c", &v); //recibe el valor para repetir el proceso
127     if (v == 'y' || v == 'Y') //validación para ser y o Y la llave
128         menu(cadena, cadena1); //si así se desea entrar de nuevo a menú repitiendo así el proceso
129     else //sino
130         exit(0); //matas al programa
131 } //fin de la función
132
133
134 void palindromo(char cadena[]) { //FUNCION PARA SABER SI LA CADENA 1 ES PALINDROMO
135     char aux[100]; //variable tipo caracter que sirve como auxiliar de 100 caracteres
136
137     void palindromo(char cadena[]) { //FUNCION PARA SABER SI LA CADENA 1 ES PALINDROMO
138         char aux[100]; //variable tipo caracter que sirve como auxiliar de 100 caracteres
139         strcpy(aux, cadena); //palabra reservada de la librería string que copia la cadena
140         strrev(aux); //palabra reservada xde la librería string que invierte la cadena
141         if(strcmp(cadena, aux) == 0) //validación con la palabra reservada de comparación si la invertida es igual a la cadena
142             printf("La cadena 1 ingresada es palindromo\n"); //mensaje de es o no es palindromo
143         else
144             printf("La cadena 1 ingresada no es palindromo\n");
145     }
146 }
147
148 void palindromo1(char cadena1[]) { //FUNCION QUE HACE LO MISMO QUE PPALINDROMO PERO ESTA ES PARA LA CADENA 2
149     char aux[100];
150     strcpy(aux, cadena1);
151     strrev(aux);
152     if(strcmp(cadena1, aux) == 0)
153         printf("La cadena 2 ingresada es palindromo\n");
154     else
155         printf("La cadena 2 ingresada no es palindromo\n");
156 } //fin de la función palindromo
157
158
159 int longitud(char cadena[]) { //FUNCION QUE TE DICE LA LONGITUD DE LA CADENA 1
160     int lon= strlen(cadena); //palabra reservada de la librería string que cuenta los caracteres dentro de la cadena
161     return lon; //valor que regresa en entero
162 } //fin de la función longitud
163
164
165 char concatenacion(char cadena[], char cadena1[]) { //FUNCION QUE TE CONCATENA LAS DOS CADENAS INGRESADAS
166     char aux1[100], aux2[100], aux3[100]; //variables tipo caracter que servirán como auxiliares para copiar y pegar las cadenas juntas
167     strcpy(aux1, cadena); //se copia la cadena 1
168     strcpy(aux2, cadena1); //se copia la cadena 2
169     strcpy(aux3, cadena); //se copia de nuevo la cadena 1 para que la cadena 1 no pierda su valor inicial
170     printf("\nLa concatenacion queda de la siguiente manera = %s\n", strcat(aux3, cadena1)); //se imprime la concatenacion con strcat
171     printf("\nLa concatenacion al reves queda de la siguiente manera = %s\n", strcat(aux2, aux1)); //se imprime la concatenacion con strcat in

```

```

172 void potencia(char cadena[], int pot1){ //FUNCION QUE TE GENERA LA POTENCIA DE LA CADENA 1
173     int i; //variable para ser el contador
174     char aux[100]; //mi auxiliar de 100 caracteres
175     strcpy(aux, cadena); //copia la cadena en el auxiliar
176     if(pot1>0){ //valida se la potencia es mayor a 0
177         for(i=0; i< pot1 ;i++) //entra a un ciclo para repetir las veces que indique pot1
178             printf("%s", aux); //imprime cómo queda
179             printf("\n"); //salto de línea bien épico
180     } //aahh pero sino es así
181     else if(pot1<0){ //pot1 es negativo wowowo
182         strcpy(aux, strrev(cadena)); //se copia al revés la cadena en el auxiliar
183         for(i=0; i> pot1 ;i--) //entra al ciclo hasta que se cumpla pot2
184             printf("%s", aux); //los despliega
185             printf("\n"); //salto de línea épico again
186     }
187     else // si es 0 pues Epsilon
188         printf("E");
189 }
190
191 //FUNCION QUE TE DA LA POTENCIA DE LA SEGUNDA CADENA, LO MISMO DE ARRIBA PERO PARA LA CADENA 2
192 void potencial(char cadena1[], int pot2){
193     int i;
194     char aux[100];
195     strcpy(aux, cadena1);
196     if(pot2>0){
197         for(i=0; i< pot2 ;i++)
198             printf("%s", aux);
199             printf("\n");
200     }
201     else if(pot2<0){
202         strcpy(aux, strrev(cadena1));
203         for(i=0; i> pot2 ;i--)
204             printf("%s", aux);
205             printf("\n");
206     }
207 }
208
209 //FUNCION QUE TE DA LA POTENCIA DE LA SEGUNDA CADENA, LO MISMO DE ARRIBA PERO PARA LA CADENA 2
210 void potencial(char cadena1[], int pot2){
211     int i;
212     char aux[100];
213     strcpy(aux, cadena1);
214     if(pot2>0){
215         for(i=0; i< pot2 ;i++)
216             printf("%s", aux);
217             printf("\n");
218     }
219     else if(pot2<0){
220         strcpy(aux, strrev(cadena1));
221         for(i=0; i> pot2 ;i--)
222             printf("%s", aux);
223             printf("\n");
224     }
225     else
226         printf("E");
227 }
228
229 char inverso(char cadena[], char cadena1[]){ //FUNCION QUE TE DA LA CADENA EN INVERSO
230     printf("%s", strrev(cadena)); //impresión de la cadena 1 al revés con strrev de reversa
231     printf("\n"); //bello salto de línea
232     printf("%s", strrev(cadena1)); //impresión de la cadena 2 al revés con strrev de reversa
233     printf("\n"); //bello salto de línea again
234 }
235
236 char prefijos(char cadena[]){ //FUNCION QUE DESPLIEGA PREFIJOS DE LA CADENA 1
237     int l = strlen(cadena); //se cuenta los caracteres que hay en la cadena 1
238     int i,x; //contadores y barrido de posiciones
239     printf("E\n"); //sexy salto de línea y Epsilon que siempre está
240     for (i = 0; i < l; i++) //ciclo para ir contando los caracteres
241     {

```

```

232 //FUNCIÓN QUE DESPLIEGA PREFIJOS DE LA CADENA 2, LO MISMO QUE ARRIBA
233 char prefijos1(char cadena1[]){
234     int l = strlen(cadena1);
235     int i,x;
236     printf("E\n");
237     for (i = 0; i < l; i++)
238     {
239         for (x = 0; x <= i; x++)
240             printf("%c", cadena1[x]);
241         printf("\n");
242     }
243 }
244
245
246 void sufijos(char cadena[]){ //Operación que calcula Los sufijos de una cadena, recibe La cadena en cuestión{
247     int l = strlen(cadena); //Se declara una variable y se iguala a la longitud de la cadena, se ocupará posteriormente para hacer el c
248     int i = l; //Variable contadora, en el caso de Los sufijos, se recorrerá de atrás hacia adelante la cadena, por lo que se inicializ
249     int x;
250     printf("E"); //Como primera instancia se imprime Epsilon (servirá haya o no una cadena posterior)
251     for (i = l; i >= 0; i--) //El ciclo se detendrá al llegar al primer valor de la cadena
252     {
253         for (x = i; x < l; x++) //Este ciclo indica las posiciones que deben irse imprimiendo de acuerdo a la cadena y al ciclo anterior
254             printf("%c", cadena[x]);
255         printf("\n");
256     }
257 }
258
259
260 void sufijos1(char cadena1[]){ //Operación que calcula Los sufijos de una cadena, recibe La cadena en cuestión{
261     int l = strlen(cadena1); //Se declara una variable y se iguala a la longitud de la cadena, se ocupará posteriormente para hacer el
262     int i = l; //Variable contadora, en el caso de Los sufijos, se recorrerá de atrás hacia adelante la cadena, por lo que se inicializ
263     int x;
264     printf("E"); //Como primera instancia se imprime Epsilon (servirá haya o no una cadena posterior)
265     for (i = l; i >= 0; i--) //El ciclo se detendrá al llegar al primer valor de la cadena
266     {
267         for (x = i; x < l; x++)
268             printf("%c", cadena1[x]);
269         printf("\n");
270     }
271 }
272
273 void subcadenas(char cadena[]) //FUNCIÓN PARA DESPLEGAR SUBCADENAS
274 {
275     int l = strlen(cadena);
276     int lim = 1; //Cantidad de caracteres que irá tomando por caso
277     int li = 1; //Variable auxiliar para saber el número de posición que debe imprimirse
278     int i, x, y; //Variables de apoyo
279     i = x = y = 0;
280     while(y < l) //El ciclo termina cuando se alcance la longitud de la cadena
281     {
282         while(x < l && i <= l) //Ciclo que indica el número de caracteres que deben imprimirse según sea el caso
283         {
284             if(i == (l - (lim - 1))) //Si la cadena no tiene caracteres la función termina
285                 break;
286             for(i; i < li; i++) //Ciclo que indica las posiciones que deben irse imprimiendo
287                 printf("%c", cadena[i]);
288             printf("\n");
289             i -= lim - 1;
290             li++;
291             x++;
292         }
293         li = ++lim;
294         x = i = 0;
295         y++;
296         printf("\n");
297     }
298 }
299
300
301 void subcadenas1(char cadena1[])
302 {

```

```

308 void subcadenas1(char cadena1[])
309 {
310     int l = strlen (cadena1);
311     int lim = 1; //Cantidad de caracteres que irá tomando por caso
312     int li = 1; //Variable auxiliar para saber el número de posición que debe imprimirse
313     int i, x, y; //Variables de apoyo
314     i = x = y = 0;
315     while(y < l) //El ciclo termina cuando se alcance la longitud de la cadena
316     {
317         while(x < l && i <= l) //Ciclo que indica el número de caracteres que deben imprimirse según sea el caso
318         {
319             if(i == (l - (lim - 1))) //Si la cadena no tiene caracteres la función termina
320                 break;
321
322             for(i; i < li; i++) //Ciclo que indica las posiciones que deben irse imprimiendo
323                 printf("%c", cadena1[i]);
324
325             printf("\n");
326             i -= lim - 1;
327             li++;
328             x++;
329         }
330         li = ++lim;
331         x = i = 0;
332         y++;
333         printf("\n");
334     }
335 }
336

```

FUNCIONAMIENTO

Inserción de las dos cadenas y desplegado del menú.

Cadena 1: perro.

Cadena 2: oso.

```

BIENVENIDO AL PROGRAMA DE OPERACIONES CON CADENAS
EL ALFABETO ES DE LA {a-z}
Inserte cadena 1
perro
Inserte cadena 2
oso
ELIJA EL NUMERO DE LA OPCION DEL PROCESO QUE QUIERE REALIZAR
1.- PALINDROMO
2.- LONGITUD
3.- CONCATENACION
4.- POTENCIA
5.- INVERSO
6.- PREFIJOS
7.- SUFIJOS
8.- SUBCADENAS

```

Elección de la opción 1 indicando si cada una de las cadenas es palíndromo y desplegado con opción de repetir algún otro proceso.

```

BIENVENIDO AL PROGRAMA DE OPERACIONES CON CADENAS
EL ALFABETO ES DE LA {a-z}
inserte cadena 1
perro
inserte cadena 2
oso
ELIJA EL NUMERO DE LA OPCION DEL PROCESO QUE QUIERE REALIZAR
1.- PALINDROMO
2.- LONGITUD
3.- CONCATENACION
4.- POTENCIA
5.- INVERSO
6.- PREFIJOS
7.- SUFIJOS
8.- SUBCADENAS
1
La cadena 1 ingresada no es palindromo
La cadena 2 ingresada es palindromo
LE GUSTARIA REPETIR EL PROCESO (Y/N)

```

Elección de la opción 2, longitud.

```

inserte cadena 2
oso
ELIJA EL NUMERO DE LA OPCION DEL PROCESO QUE QUIERE REALIZAR
1.- PALINDROMO
2.- LONGITUD
3.- CONCATENACION
4.- POTENCIA
5.- INVERSO
6.- PREFIJOS
7.- SUFIJOS
8.- SUBCADENAS
1
La cadena 1 ingresada no es palindromo
La cadena 2 ingresada es palindromo
LE GUSTARIA REPETIR EL PROCESO (Y/N)
y
ELIJA EL NUMERO DE LA OPCION DEL PROCESO QUE QUIERE REALIZAR
1.- PALINDROMO
2.- LONGITUD
3.- CONCATENACION
4.- POTENCIA
5.- INVERSO
6.- PREFIJOS
7.- SUFIJOS
8.- SUBCADENAS
2
La longitud de la cadena 1 es: 5
La longitud de la cadena 2 es: 3
LE GUSTARIA REPETIR EL PROCESO (Y/N)

```

Elección de la opción 3, concatenación.

```

ELIJA EL NUMERO DE LA OPCION DEL PROCESO QUE QUIERE REALIZAR
1.- PALINDROMO
2.- LONGITUD
3.- CONCATENACION
4.- POTENCIA
5.- INVERSO
6.- PREFIJOS
7.- SUFIJOS
8.- SUBCADENAS
2
La longitud de la cadena 1 es: 5
La longitud de la cadena 2 es: 3
LE GUSTARIA REPETIR EL PROCESO (Y/N)
y
ELIJA EL NUMERO DE LA OPCION DEL PROCESO QUE QUIERE REALIZAR
1.- PALINDROMO
2.- LONGITUD
3.- CONCATENACION
4.- POTENCIA
5.- INVERSO
6.- PREFIJOS
7.- SUFIJOS
8.- SUBCADENAS
3
La concatenación queda de la siguiente manera = perrooso
La concatenación al revés queda de la siguiente manera = oso perro
LE GUSTARIA REPETIR EL PROCESO (Y/N)

```

Elección de la opción 4, potencia.

3

Elección de la opción 5, inverso.

4

Elección de la opción 6, prefijos.

Cambio de cadenas.

Cadena 1: osoperesoso.

Cadena 2: perroacurrucado.


```
C:\Users\marco\Documents\ESCOM\CUARTO SEMESTRE\TEORÍA COMPUTACIONAL\PRÁCTICA 1 OPERACIONES CON CADENAS\OPERACIONES CON...
e
o
so
oso
ososo
esoso
eresoso
peresoso
operesoso
soperesoso
osoperesoso
E
o
do
ado
cado
ucado
rucado
rrucado
urnrucado
curnrucado
acurnrucado
pacurnrucado
roacurnrucado
rroacurnrucado
eruoacurnrucado
peruoacurnrucado
¿ LE GUSTARÍA REPETIR EL PROCESO? (Y/N)
```

Elección de la operación 7, sufijos.

```
C:\Users\marco\Documents\ESCOM\CUARTO SEMESTRE\TEORÍA COMPUTACIONAL\PRÁCTICA 1 OPERACIONES CON CADENAS\OPERACIONES CON...
e
o
so
oso
ososo
esoso
eresoso
peresoso
operesoso
soperesoso
osoperesoso
E
o
do
ado
cado
ucado
rucado
rrucado
urnrucado
curnrucado
acurnrucado
pacurnrucado
roacurnrucado
rroacurnrucado
eruoacurnrucado
peruoacurnrucado
¿ LE GUSTARÍA REPETIR EL PROCESO? (Y/N)
```

Elección de la operación 8, sub cadenas.

```
C:\Users\marco\Documents\ESCOM\CUARTO SEMESTRE\TEORÍA COMPUTACIONAL\PRÁCTICA 1 OPERACIONES CON CADENAS\OPERACIONES CON...
s
o
p
e
r
e
s
o
s
o
os
so
op
pe
er
re
es
so
os
so
oso
sop
ope
per
ere
res
eso
sos

C:\Users\marco\Documents\ESCOM\CUARTO SEMESTRE\TEORÍA COMPUTACIONAL\PRÁCTICA 1 OPERACIONES CON CADENAS\OPERACIONES CON...
oper
pere
eres
reso
esos
soso
osope
soper
opere
peres
ereso
resos
esoso
osoper
sopere
operes
pereso
eresos
resoso
osopere
soperes
opereso
peresos
eresoso
osoperes
sopereso
operesoso

Seleccionar C:\Users\marco\Documents\ESCOM\CUARTO SEMESTRE\TEORÍA COMPUTACIONAL\PRÁCTICA 1 OPERACIONES CON CADENAS\OPERAC...
operesos
peresoso
osopereso
soperesos
operesoso
osoperesos
soperesoso
osoperesoso
p
e
r
o
a
c
u
r
u
c
a
d
o
pe
er
```

```
Seleccionar C:\Users\marco\Documents\ESCOM\CUARTO SEMESTRE\TEOR-A COMPUTACIONAL\PR-CTICA 1 OPERACIONES CON CADENAS\OPERAC...
rr
ro
oa
ac
cu
un
rr
ru
uc
ca
ad
do

per
err
rro
roa
oac
acu
cur
urr
rru
ruc
uca
cad
ado

per
err
erro

Seleccionar C:\Users\marco\Documents\ESCOM\CUARTO SEMESTRE\TEOR-A COMPUTACIONAL\PR-CTICA 1 OPERACIONES CON CADENAS\OPERAC...
per
erro
erro
roa
roac
oacu
acur
curr
unru
rruc
ruca
ucad
ucado

perro
erroa
erroac
roacu
oacur
acurr
curru
urruc
rruca
rucad
rucado

perroa
erroac
erroacu
roacur
oacurr
acurruc
curruca
urrucad
rrucado

perroa
erroa
erroac
roacu
oacur
acurruc
curruca
urrucad
rrucado

perroa
erroa
erroac
roacu
oacur
acurruc
curruca
urrucad
rrucado
```

CONCLUSIONES

En conclusión, la práctica número 1 me agradó bastante, además me ayudó más que nada a practicar de nuevo mi destreza en la programación ya que tenía un rato sin programar en lenguaje C y creo que ya estaba algo.

Por otro lado, creo que ha sido demasiado bueno el hecho de ejercitar un poco los 8 procesos que se pueden llevar a cabo en la ejecución del programa, más allá de que algunos algoritmos son fáciles de llevar a cabo te das cuenta de que tiene un correcto funcionamiento sin ser la gran cosa en el código y que algunas funciones puedes ocuparlas en otros procesos.

Me quedo con lo muy bueno de esta práctica lo cual fue el diseño del programa y el volver a ponerme a codificar ya que creo que estaba un poco fuera de práctica, también me quedo con la elaboración de diagramas de flujo dentro de un software de aplicación cuyo nombre es “Visustin” y es bastante sencillo de utilizar.

BIBLIOGRAFÍA

Elaboración autentica de Marcos Oswaldo Vázquez Moreno con ayuda del software de programación Visustin v8 y de Devc.