



# **Universidad Autónoma de Baja California**

## **Facultad de Ingeniería, Arquitectura y Diseño**



Ingeniero en Computación

**Asignatura:**

Programación Estructurada

**Actividad 7:**

**Brayan Arturo Rocha Meneses**

**Ensenada Baja California 3 de Marzo del 2024**

ACTIVIDADES

EXPLORER

ACTIVIDADES

Act95 > RMBA\_ACT7\_01.cpp > salida9(char[])

```
1 //Brayan Arturo Rocha Menses Matricula:371049
2 //Fecha 11/03/2024 19/03/2024
3 //Parte 1
4 //RMBA_ACT7_01_432
5
6
7 // Librerias
8 #include <stdio.h>
9 #include <stdlib.h>
10
11 // Menu Principal
12 int mses();
13 void menu();
14
15 //Funcion para la pedir de datos
16 void datos(char nombre[]);
17
18 // Validacion
19 int validar(char mensj[], int r1, int rf);
20
21 // Declarar funciones
22 //Ejercicios de la Parte 1
23 void salida1(char nombre[]);
24 void salida2(char nombre[]);
25 void salida3(char nombre[]);
26 void salida4(char nombre[]);
27 void salida5(char nombre[]);
28 void salida6(char nombre[]);
29 void salida7(char nombre[]);
30 void salida8(char nombre[]);
31 void salida9(char nombre[]);
32 void salida10(char nombre[]);
33
34 // Funcion main
35 int main()
36 {
37     menu();
38     return 0;
39 }
40
41 // Menu que se muestra al usuario
42 int mses()
43 {
44     int op;
45
46     system("CLS");
47     printf("\n MENU PARA LAS DIFERENTES SALIDAS DE LA CADENA \n");
48     printf("\n OPCIONES: \n");
49     printf("1.- Salida 1: MAYUSCULAS\n");
50     printf("2.- Salida 2: MAYUSCULAS AL REVES\n");
51     printf("3.- Salida 3: UNA LETRA POR RENGLON\n");
```

Ln 445, Col 2 Spaces: 4 UTF-8 CRLF {} C++ Win32 Prettier

ACTIVIDADES

EXPLORER

Act95 > RMBA\_ACT7\_01.cpp > salida9(char[])

```
41 // Menu que se muestra al usuario
42 int mses()
43 {
44     int op;
45
46     system("CLS");
47     printf("\n MENU PARA LAS DIFERENTES SALIDAS DE LA CADENA \n");
48     printf("\n OPCIONES: \n");
49     printf("1.- Salida 1: MAYUSCULAS\n");
50     printf("2.- Salida 2: MAYUSCULAS AL REVES\n");
51     printf("3.- Salida 3: UNA LETRA POR RENGLON\n");
52     printf("4.- Salida 4: UNA LETRA POR RENGLON AL REVES\n");
53     printf("5.- Salida 5: LA PALABRA MENOS UNA LETRA\n");
54     printf("6.- Salida 6: LA PALABRA MENOS UNA LETRA AL REVES\n");
55     printf("7.- Salida 7: LA PALABRA MENOS UNA LETRA DEL INICIO\n");
56     printf("8.- Salida 8: LA PALABRA MENOS UNA LETRA DEL INICIO AL REVES\n");
57     printf("9.- Salida 9: SOLO CONSONANTES\n");
58     printf("10.- Salida 10: SOLO VOCALES\n");
59     printf("0.- SALIR \n");
60     op = validar("ESCOGE UNA OPCION: ", 0, 10);
61
62     return op;
63 }
64
65 // Casos depende de la seleccion del usuario
66 void menu()
67 {
68     system("CLS");
69     int op;
70     char nombre[50];
71     do {
72         op = mses();
73         switch(op)
74         {
75             case 1:
76                 datos(nombre);
77                 salida1(nombre);
78                 break;
79             case 2:
80                 datos(nombre);
81                 salida2(nombre);
82                 break;
83             case 3:
84                 datos(nombre);
85                 salida3(nombre);
86                 break;
87             case 4:
88                 datos(nombre);
89                 salida4(nombre);
90                 break;
```

Ln 445, Col 2 Spaces: 4 UTF-8 CRLF {} C++ Win32 Prettier

ACTIVIDADES

EXPLORER

Act95 > RMBA\_ACT7\_01.cpp > salida9(char [])

```
67 {
68     system("CLS");
69     int op;
70     char nombre[50];
71     do {
72         op = msges();
73         switch(op)
74         {
75             case 1:
76                 datos(nombre);
77                 salida1(nombre);
78                 break;
79             case 2:
80                 datos(nombre);
81                 salida2(nombre);
82                 break;
83             case 3:
84                 datos(nombre);
85                 salida3(nombre);
86                 break;
87             case 4:
88                 datos(nombre);
89                 salida4(nombre);
90                 break;
91             case 5:
92                 datos(nombre);
93                 salida5(nombre);
94                 break;
95             case 6:
96                 datos(nombre);
97                 salida6(nombre);
98                 break;
99             case 7:
100                 datos(nombre);
101                 salida7(nombre);
102                 break;
103             case 8:
104                 datos(nombre);
105                 salida8(nombre);
106                 break;
107             case 9:
108                 datos(nombre);
109                 salida9(nombre);
110                 break;
111             case 10:
112                 datos(nombre);
113                 salida10(nombre);
114                 break;
115             case 0:
116                 printf("Saliste del programa.\n");
117                 break;
```

Ln 445, Col 2 Spaces: 4 UTF-8 CRLF ( ) C++ Win32 Prettier

ACTIVIDADES

EXPLORER

Act95 > RMBA\_ACT7\_01.cpp > salida9(char [])

```
106         break;
107     case 9:
108         datos(nombre);
109         salida9(nombre);
110         break;
111     case 10:
112         datos(nombre);
113         salida10(nombre);
114         break;
115     case 0:
116         printf("Saliste del programa.\n");
117         break;
118     default:
119         printf("Opcion Invalida.\n");
120         break;
121     }
122     while(op != 0);
123     system("PAUSE");
124 }
125
126 // Validacion
127 int validar(char mensj[], int ri, int rf)
128 {
129     int num;
130     // Cadena que va a leer el mensaje ingresado
131     char cadena[50];
132     // Proceso y desarrollo
133     do
134     {
135         printf("%s", mensj);
136         // Borrar basura
137         fflush(stdin);
138         gets(cadena);
139         num = atoi(cadena);
140     } while (num < ri || num > rf);
141     // Retorna el valor que num, entre los rangos dados
142     return num;
143 }
144
145 //Pedir datos
146 void datos(char nombre[])
147 {
148     system("CLS");
149     printf("Introduzca su nombre: ");
150     // Borrar basura
151     fflush(stdin);
152     gets(nombre);
153 }
154
155 // Salidas diferentes de la cadena
156 // SALIDA 1
```

Ln 445, Col 2 Spaces: 4 UTF-8 CRLF ( ) C++ Win32 Prettier

```
154 // Salidas diferentes de la cadena
155 // SALIDA 1
156 void salida1(char nombre[])
157 {
158     system("CLS");
159     //Imprimir nombre de la salida
160     printf("Salida 1: MAYUSCULAS\n");
161     // Recorre todos los caracteres uno por uno
162     for (int i = 0; nombre[i] != '\0'; i++)
163     {
164         // Comprobar si estan en minusculas
165         if (nombre[i] >= 'a')
166         {
167             if (nombre[i] <= 'z')
168             {
169                 // Si las condiciones son verdaderas se hace la conversion a mayusculas
170                 nombre[i] = nombre[i] - ('a' - 'A');
171             }
172         }
173         // Imprime la palabra en mayusculas o igual
174         printf("%c", nombre[i]);
175     }
176     printf("\n");
177     system("PAUSE");
178 }
179
180 // SALIDA 2
181 void salida2(char nombre[])
182 {
183     system("CLS");
184     //Imprimir nombre de la salida
185     printf("Salida 2: MAYUSCULAS AL REVES\n");
186     // Declarar variable para la longitud/largo
187     int length = 0;
188     // Calcular
189     while (nombre[length] != '\0')
190     {
191         length++;
192     }
193     // Recorre todos los caracteres uno por uno
194     for (int i = length - 1; i >= 0; i--)
195     {
196         // Comprobar si estan en minusculas
197         if (nombre[i] >= 'a')
198         {
199             if (nombre[i] <= 'z')
200             {
201                 // Si las condiciones son verdaderas se hace la conversion a mayusculas
202                 nombre[i] = nombre[i] - ('a' - 'A');
203             }
204         }
205     }
206     // Imprime la palabra en mayusculas o igual pero al reves
207     printf("%c", nombre[i]);
208 }
209 printf("\n");
210 system("PAUSE");
211 }
```

```
212 // SALIDA 3
213 void salida3(char nombre[])
214 {
215     system("CLS");
216     // Imprimir nombre de la salida
217     printf("Salida 3: UNA LETRA POR RENGLON\n");
218     // Recorre todos los caracteres uno por uno
219     for (int i = 0; nombre[i] != '\0'; i++)
220     {
221         if (nombre[i] >= 'a')
222         {
223             if (nombre[i] <= 'z')
224             {
225                 nombre[i] = nombre[i] - ('a' - 'A');
226             }
227         }
228         // Imprime la cadena y al usar \n podemos sacar una por renglon
229         printf("%c\n", nombre[i]);
230     }
231     // Para que la palabra se imprima en cada linea
232     printf("\n");
233     system("PAUSE");
234 }
235
236 // SALIDA 4
237 void salida4(char nombre[])
238 {
239     system("CLS");
240     // Imprimir nombre de la salida
241     printf("Salida 4: UNA LETRA POR RENGLON AL REVES\n");
242     // Declarar variable para la longitud/largo
243     int length = 0;
244     // Calcular
245     while (nombre[length] != '\0')
246     {
247         length++;
248     }
249     // Recorre todos los caracteres uno por uno y hace que salga al reves la palabra
250     for (int i = length - 1; i >= 0; i--)
251     {
252         if (nombre[i] >= 'a')
253         {
254             if (nombre[i] <= 'z')
255             {
256                 nombre[i] = nombre[i] - ('a' - 'A');
257             }
258         }
259         // Imprime la cadena y al usar \n podemos sacar una por renglon
260         printf("%c\n", nombre[i]);
261     }
262     // Para que la palabra se imprima en cada linea
263     printf("\n");
264     system("PAUSE");
265 }
```

EXPLORES...WelcomeC curp.hRelease Notes: 1.85.2LFOA\_Act07\_01\_432.cppLFOA\_Act07\_02\_432.cppRMBA\_ACT7\_01.cpp

ACTIVIDADES

259  
260  
261  
262  
263  
264  
265  
266  
267  
268  
269  
270  
271  
272  
273  
274  
275  
276  
277  
278  
279  
280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309

```
    }  
    // Imprime la cadena y al usar \n podemos sacar una por renglon  
    printf("%c\n", nombre[i]);  
}  
// Para que la palabra se imprima en cada linea  
printf("\n");  
system("PAUSE");  
}  
// SALIDA 5  
void salida5(char nombre[])  
{  
    system("CLS");  
    // Imprimir nombre de la salida  
    printf("Salida 5: LA PALABRA MENOS UNA LETRA POR RENGLON\n");  
    // Declarar variable para la longitud/largo  
    int length = 0;  
    // Calcular  
    while (nombre[length] != '\0')  
    {  
        length++;  
    }  
    // La longitud de la palabra y va disminuyendo hasta llegar a 1  
    for (int i = length; i >= 1; i--)  
    {  
        // Recorre cada letra y convierte la letra actual a mayusculas  
        for (int j = 0; j < i; j++)  
        {  
            if (nombre[j] >= 'a')  
            {  
                if (nombre[j] <= 'z')  
                {  
                    nombre[j] = nombre[j] - ('a' - 'A');  
                }  
            }  
        }  
        // Imprime la palabra  
        for (int j = 0; j < i; j++)  
        {  
            printf("%c", nombre[j]);  
        }  
        // Para que la palabra se imprima en cada linea  
        printf("\n");  
    }  
    printf("\n");  
    system("PAUSE");  
}
```

OUTLINE  
TIMELINE

Ln 445, Col 2 Spaces: 4 UTF-8 CRLF ( ) C++ Win32 Prettier

EXPLORES...WelcomeC curp.hRelease Notes: 1.85.2LFOA\_Act07\_01\_432.cppLFOA\_Act07\_02\_432.cppRMBA\_ACT7\_01.cpp

ACTIVIDADES

309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321  
322  
323  
324  
325  
326  
327  
328  
329  
330  
331  
332  
333  
334  
335  
336  
337  
338  
339  
340  
341  
342  
343  
344  
345  
346  
347  
348  
349  
350  
351  
352  
353  
354  
355  
356

```
    }  
    // SALIDA 6  
    void salida6(char nombre[])  
    {  
        system("CLS");  
        // Imprimir nombre de la salida  
        printf("Salida 6: LA PALABRA MENOS UNA LETRA AL REVES\n");  
        // Declarar variable para la longitud/largo  
        int length = 0;  
        // Calcular  
        while (nombre[length] != '\0')  
        {  
            length++;  
        }  
        // Declarar la copia y asignarle un tamaño  
        char copia_nombre[50];  
        // Copiar el nombre a la copia  
        for (int i = 0; i < length; i++)  
        {  
            copia_nombre[i] = nombre[i];  
        }  
        // Que el ultimo caracter sea nulo  
        copia_nombre[length] = '\0';  
        // Muestra la cadena gradualmente  
        while (length > 1)  
        {  
            // Imprimir la cadena copiada en mayusculas al reves  
            for (int i = length - 1; i >= 0; i--)  
            {  
                if (copia_nombre[i] >= 'a')  
                {  
                    if (copia_nombre[i] <= 'z')  
                    {  
                        copia_nombre[i] = copia_nombre[i] - ('a' - 'A');  
                    }  
                }  
                printf("%c", copia_nombre[i]);  
            }  
            // Para pasar a la siguiente linea  
            printf("\n");  
            // Acortar la cadena copiada eliminando el primer carácter  
            for (int i = 0; i < length - 1; i++)  
            {  
                copia_nombre[i] = copia_nombre[i + 1];  
            }  
            // Para que el nuevo ultimo caracter si sea el nulo  
            copia_nombre[length - 1] = '\0';  
            // Para que se vea que la cadena ya no tiene un caracter  
            length--;  
        }  
        system("PAUSE");  
    }
```

OUTLINE  
TIMELINE

Ln 445, Col 2 Spaces: 4 UTF-8 CRLF ( ) C++ Win32 Prettier

EXPLORES...Welcomecurp.hRelease Notes: 1.85.2LFOA\_Act07\_01\_432.cppLFOA\_Act07\_02\_432.cppRMBA\_ACT7\_01.cpp

ACTIVIDADES

359 // SALIDA 7  
360 void salida7(char nombre[])  
361 {  
362 system("CLS");  
363 // Imprimir nombre de la salida  
364 printf("Salida 7: LA PALABRA MENOS UNA LETRA DEL INICIO\n");  
365 // Declarar variable para la longitud/largo  
366 int length = 0;  
367 // Calcular  
368 while (nombre[length] != '\0')  
369 {  
370 length++;  
371 }  
372 // Declarar la copia y asignarle un tamaño  
373 char copia\_nombre[50];  
374 // Copiar el nombre a la copia  
375 for (int i = 0; i < length; i++)  
376 {  
377 copia\_nombre[i] = nombre[i];  
378 }  
379 // Que el ultimo caracter sea nulo  
380 copia\_nombre[length] = '\0';  
381 // Itera y muestra la cadena gradualmente  
382 while (length > 1)  
383 {  
384 // Imprimir la cadena copiada en mayúsculas  
385 for (int i = 0; i < length; i++)  
386 {  
387 if (copia\_nombre[i] >= 'a' && copia\_nombre[i] <= 'z')  
388 {  
389 copia\_nombre[i] = copia\_nombre[i] - ('a' - 'A');  
390 }  
391 printf("%c", copia\_nombre[i]);  
392 }  
393 printf("\n");  
394 // Acortar la cadena copiada eliminando el primer carácter  
395 for (int i = 0; i < length - 1; i++)  
396 {  
397 copia\_nombre[i] = copia\_nombre[i + 1];  
398 }  
399 // Para que el nuevo ultimo caracter si sea el nulo  
400 copia\_nombre[length - 1] = '\0';  
401 // Para que se vea que la cadena ya no tiene un caracter  
402 length--;  
403 }  
404 printf("\n");  
405 system("PAUSE");  
406 }  
407  
408 // SALIDA 8  
409 void salida8(char nombre[]

Ln 445, Col 2 Spaces: 4 UTF-8 CRLF ( ) C++ Win32 Prettier

EXPLORES...Welcomecurp.hRelease Notes: 1.85.2LFOA\_Act07\_01\_432.cppLFOA\_Act07\_02\_432.cppRMBA\_ACT7\_01.cpp

ACTIVIDADES

406 }  
407  
408 // SALIDA 8  
409 void salida8(char nombre[])  
410 {  
411 system("CLS");  
412 // Imprimir nombre de la salida  
413 printf("Salida 8: LA PALABRA MENOS UNA LETRA DEL INICIO AL REVES\n");  
414 // Declarar variable para la longitud/largo  
415 int length = 0;  
416 // Calcular  
417 while (nombre[length] != '\0')  
418 {  
419 length++;  
420 }  
421 // Leer una cadena al revés hasta llegar al primer elemento que es el 0  
422 for (int i = length - 1; i >= 0; i--)  
423 {  
424 // Convertir la letra actual a mayúsculas  
425 for (int j = i; j >= 0; j--)  
426 {  
427 if (nombre[j] >= 'a' && nombre[j] <= 'z')  
428 {  
429 nombre[j] = nombre[j] - ('a' - 'A');  
430 }  
431 }  
432 // Imprimir la palabra actual al revés  
433 for (int j = i; j >= 0; j--)  
434 {  
435 printf("%c", nombre[j]);  
436 }  
437 // Para que la palabra se imprima en cada línea  
438 printf("\n");  
439 }  
440 system("PAUSE");  
441 }  
442  
443 // SALIDA 9  
444 void salida9(char nombre[])  
445 {  
446 system("CLS");  
447 // Imprimir nombre de la salida  
448 printf("Salida 9: SOLO CONSONANTES\n");  
449 // Imprimir el nombre en mayúsculas  
450 for (int i = 0; nombre[i] != '\0'; i++)  
451 {  
452 if (nombre[i] >= 'a')  
453 {  
454 if (nombre[i] <= 'z')  
455 {  
456 nombre[i] = nombre[i] - ('a' - 'A');

Ln 445, Col 2 Spaces: 4 UTF-8 CRLF ( ) C++ Win32 Prettier

ACTIVIDADES

EXPLORER

ACTIVIDADES

Act95 > RMBA\_ACT7\_01.cpp > salida9(char[])

```
467 if ( nombre[i] != 'I')
468 {
469     if ( nombre[i] != 'O')
470     {
471         if ( nombre[i] != 'U')
472         {
473             printf ( "%c", nombre[i] );
474         }
475     }
476 }
477 }
478 }
479 }
480 printf("\n");
481 system("PAUSE");
482 }
483 }
484 // SALIDA 10
485 void salida10(char nombre[])
486 {
487     system ("CLS");
488     //Imprimir nombre de la salida
489     printf("Salida 10: SOLO VOALES\n");
490     // Imprimir el nombre en mayúsculas
491     for (int i = 0; nombre[i] != '\0'; i++)
492     {
493         if (nombre[i] >= 'a')
494         {
495             if(nombre[i] <= 'z')
496             {
497                 nombre[i] = nombre[i] - ('a' - 'A');
498             }
499         }
500     }
501     // Imprimir solo las vocales
502     for (int i = 0; nombre[i] != '\0'; i++)
503     {
504         if ( nombre[i] == 'A')
505         {
506             printf ( "%c", nombre[i] );
507         }
508         if (nombre[i] == 'E')
509         {
510             printf ( "%c", nombre[i] );
511         }
512         if ( nombre[i] == 'I')
513         {
514             printf ( "%c", nombre[i] );
515         }
516         if ( nombre[i] == 'O')
517         {
518             printf ( "%c", nombre[i] );
519         }
520     }
521     printf("\n");
522     system("PAUSE");
523 }
```

Ln 445, Col 2 Spaces: 4 UTF-8 CRLF ( ) C++ Win32 Prettier

ACTIVIDADES

EXPLORER

ACTIVIDADES

Act95 > RMBA\_ACT7\_01.cpp > salida10(char[])

```
485 void salida10(char nombre[])
486 {
487     system ("CLS");
488     //Imprimir nombre de la salida
489     printf("Salida 10: SOLO VOALES\n");
490     // Imprimir el nombre en mayúsculas
491     for (int i = 0; nombre[i] != '\0'; i++)
492     {
493         if (nombre[i] >= 'a')
494         {
495             if(nombre[i] <= 'z')
496             {
497                 nombre[i] = nombre[i] - ('a' - 'A');
498             }
499         }
500     }
501     // Imprimir solo las vocales
502     for (int i = 0; nombre[i] != '\0'; i++)
503     {
504         if ( nombre[i] == 'A')
505         {
506             printf ( "%c", nombre[i] );
507         }
508         if (nombre[i] == 'E')
509         {
510             printf ( "%c", nombre[i] );
511         }
512         if ( nombre[i] == 'I')
513         {
514             printf ( "%c", nombre[i] );
515         }
516         if ( nombre[i] == 'O')
517         {
518             printf ( "%c", nombre[i] );
519         }
520         if ( nombre[i] == 'U')
521         {
522             printf ( "%c", nombre[i] );
523         }
524     }
525     printf("\n");
526     system("PAUSE");
527 }
528 }
```

Ln 510, Col 40 Spaces: 4 UTF-8 CRLF ( ) C++ Win32 Prettier

```
4 //RMBA_ACT7_02_432
5
6 // Librerias
7 #include <stdio.h>
8 #include <stdlib.h>
9
10 // Menu Principal
11 int mses();
12 void menu();
13
14 // Validacion:
15 // Declaracion de la función es_numero
16 int es_numero(char caracter);
17 // Declaracion de la función es_espacio
18 int es_espacio(char caracter);
19 // Funcion de validacion
20 int validarCadena(const char *cadena);
21 // Validar la cadena
22 void validar(char cadena[]);
23
24 //Funcion para la pedir de datos
25 void datos(char cadena[]);
26
27 // Ejercicios de la Parte 2
28 void ej1_pt2(char cadena[]);
29 void ej2_pt2(char cadena[]);
30 void ej3_pt2(char cadena[]);
31 int ej4_pt2(char cadena[]);
32 int ej5_pt2(char cadena[]);
33 void ej6_pt2(char cadena[]);
34 int ej7_pt2(char cadena[]);
35 void ej8_pt2(char cadena[]);
36 void ej9_pt2(char cadena[]);
37
38 // Funcion main
39 int main()
40 {
41     menu();
42     return 0;
43 }
44
45 // Menu que se muestra al usuario
46 int mses()
47 {
48     int op;
49
50     system("CLS");
51     printf("\n MENU DE LA ACTIVIDAD 7 \n");
52     printf("\n PARTE 2 \n");
53     printf("1.- Mayusculas \n");
54     printf("2.- Minusculas \n");
55     printf("3.- Capital \n");
56     printf("4.- Cantidad \n");
57     printf("5.- Inversa \n");
58     printf("6.- Sin espacios \n");
59     printf("7.- Alfabeto \n");
60     printf("8.- Todo \n");
61     printf("9.- Palindromo \n");
62     printf("0.- SALIR \n");
63     printf("ESCOGE UNA OPCION: ");
64     scanf("%d",&op);
65
66     return op;
67 }
68
69 // Casos depende la seleccion del usuario
70 void menu()
71 {
72     int op;
73     char cadena[50];
74     do{
75         op=mses();
76         switch (op)
77         {
78             case 1:
79                 datos(cadena);
80                 ej1_pt2(cadena);
81                 break;
82             case 2:
83                 datos(cadena);
84                 ej2_pt2(cadena);
85                 break;
86             case 3:
87                 datos(cadena);
88                 ej3_pt2(cadena);
89                 break;
90             case 4:
91                 datos(cadena);
92                 ej4_pt2(cadena);
93                 break;
```

```
43 }
44
45 // Menu que se muestra al usuario
46 int mses()
47 {
48     int op;
49
50     system("CLS");
51     printf("\n MENU DE LA ACTIVIDAD 7 \n");
52     printf("\n PARTE 2 \n");
53     printf("1.- Mayusculas \n");
54     printf("2.- Minusculas \n");
55     printf("3.- Capital \n");
56     printf("4.- Cantidad \n");
57     printf("5.- Inversa \n");
58     printf("6.- Sin espacios \n");
59     printf("7.- Alfabeto \n");
60     printf("8.- Todo \n");
61     printf("9.- Palindromo \n");
62     printf("0.- SALIR \n");
63     printf("ESCOGE UNA OPCION: ");
64     scanf("%d",&op);
65
66     return op;
67 }
68
69 // Casos depende la seleccion del usuario
70 void menu()
71 {
72     int op;
73     char cadena[50];
74     do{
75         op=mses();
76         switch (op)
77         {
78             case 1:
79                 datos(cadena);
80                 ej1_pt2(cadena);
81                 break;
82             case 2:
83                 datos(cadena);
84                 ej2_pt2(cadena);
85                 break;
86             case 3:
87                 datos(cadena);
88                 ej3_pt2(cadena);
89                 break;
90             case 4:
91                 datos(cadena);
92                 ej4_pt2(cadena);
93                 break;
```



ACTIVIDADES

EXPLORER

Act95 > RMBA\_ACT7\_02.cpp > ...

```
88         ej3_pt2(cadena);
89         break;
90     case 4:
91         datos(cadena);
92         ej4_pt2(cadena);
93         break;
94     case 5:
95         datos(cadena);
96         ej5_pt2(cadena);
97         break;
98     case 6:
99         datos(cadena);
100        ej6_pt2(cadena);
101        break;
102    case 7:
103        datos(cadena);
104        ej7_pt2(cadena);
105        break;
106    case 8:
107        datos(cadena);
108        ej8_pt2(cadena);
109        break;
110    case 9:
111        datos(cadena);
112        ej9_pt2(cadena);
113        break;
114    case 0:
115        printf("Saliste del programa.\n");
116        break;
117    default:
118        printf("Opcion Invalida.\n");
119        break;
120    }
121    }while (op != 0);
122    }
123
124    // Validacion
125    int validarCadena(const char cadena[])
126    {
127        // Declarar variables e inicializarlas en 0
128        int contieneNumero = 0;
129        int contieneDobleEspacio = 0;
130        int contieneMinuscula = 0;
131        // Recorre cada caracter hasta encontrar el nulo
132        for (int i = 0; cadena[i] != '\0'; i++)
133        {
134            char caracter = cadena[i];
135            // Si encuentra que tiene un numero se establece en 1 para indicarlo
136            if (caracter >= '0')
137            {
138                if(caracter <= '9')
```

Go to Line/Column

Ln 487, Col 1 Spaces: 4 UTF-8 CRLF (1) C++ Win32 Prettier

ACTIVIDADES

EXPLORER

Act95 > RMBA\_ACT7\_02.cpp > ...

```
122    }
123
124    // Validacion
125    int validarCadena(const char cadena[])
126    {
127        // Declarar variables e inicializarlas en 0
128        int contieneNumero = 0;
129        int contieneDobleEspacio = 0;
130        int contieneMinuscula = 0;
131        // Recorre cada caracter hasta encontrar el nulo
132        for (int i = 0; cadena[i] != '\0'; i++)
133        {
134            char caracter = cadena[i];
135            // Si encuentra que tiene un numero se establece en 1 para indicarlo
136            if (caracter >= '0')
137            {
138                if(caracter <= '9')
139                {
140                    contieneNumero = 1;
141                }
142            }
143            if (caracter == ' ')
144            {
145                if (cadena[i + 1] == ' ')
146                {
147                    contieneDobleEspacio = 1;
148                }
149            }
150            if (caracter >= 'a')
151            {
152                if(caracter <= 'z')
153                {
154                    contieneMinuscula = 1;
155                }
156            }
157        }
158        // Verifica si alguna de las dos se establecio en 1
159        if (contieneNumero || contieneDobleEspacio || contieneMinuscula)
160        {
161            // La cadena no cumple con los requisitos
162            return 0;
163        }
164        else
165        {
166            // La cadena cumple con los requisitos
167            return 1;
168        }
169    }
170
171    int es_numero(char caracter)
172    {
```

Go to Line/Column

Ln 487, Col 1 Spaces: 4 UTF-8 CRLF (1) C++ Win32 Prettier

ACTIVIDADES

EXPLORER

Act95 > RMBA\_ACT7\_02.cpp > ...

```
109
170
171 int es_numero(char caracter)
172 {
173     if (caracter >= '0')
174     {
175         if (caracter <= '9')
176         {
177             return 1; // Es numero
178         }
179     }
180     return 0; // No es numero
181 }
182
183 int es_espacio(char caracter)
184 {
185     return (caracter == ' ');
186 }
187
188 int es_minuscula(char caracter)
189 {
190     if (caracter >= 'a')
191     {
192         if (caracter <= 'z')
193         {
194             return 1; // Es minuscula
195         }
196     }
197     return 0; // No es minuscula
198 }
199
200 //Pedir datos
201 void datos(char cadena[])
202 {
203     system("CLS");
204     printf("Introduzca una palabra: ");
205     fflush(stdin);
206     gets(cadena);
207 }
208
209 // Validar la cadena
210 void validar(char cadena[])
211 {
212     // Validar la cadena después de que el usuario la ingrese
213     if (!validarCadena(cadena))
214     {
215         printf("La cadena NO cumple con los requisitos.\n");
216         printf("No uses dobles espacios, minusculas y no ingreses numeros.\n");
217         // Borra la cadena si no es valida
218         cadena[0] = '\0';
219     }
220 }
```

Ln 487, Col 1 Spaces: 4 UTF-8 CRLF {} C++ Win32 Prettier

ACTIVIDADES

EXPLORER

Act95 > RMBA\_ACT7\_02.cpp > ...

```
219
220     }
221     else
222     {
223         printf("La cadena SI cumple con los requisitos.\n");
224     }
225 }
226
227 // PARTE 2
228 // Ejercicio 1 Parte 2
229 // Función que reciba como parámetro una cadena y la convierta a MAYUSCULAS
230 void eji_pt2(char cadena[])
231 {
232     system("CLS");
233     // Titulo
234     printf("CONVERTIR A MAYUSCULAS\n");
235     // Recorre todos los caracteres uno por uno
236     for (int i = 0; cadena[i] != '\0'; i++)
237     {
238         // Comprobar si estan en minusculas
239         if (cadena[i] >= 'a')
240         {
241             if (cadena[i] <= 'z')
242             {
243                 // Si las condiciones son verdaderas se hace la conversion a mayusculas
244                 cadena[i] = cadena[i] - ('a' - 'A');
245             }
246             // Imprime la palabra en mayusculas o igual
247             printf("%c", cadena[i]);
248         }
249         printf("\n");
250         getchar();
251     }
252 }
253
254 // Ejercicio 2 Parte 2
255 // Función Que reciba como parámetro una cadena y la convierta a MINUSCULAS
256 void ej2_pt2(char cadena[])
257 {
258     system("CLS");
259     // Titulo
260     printf("CONVERTIR A MINUSCULAS\n");
261     // Recorre todos los caracteres uno por uno
262     for (int i = 0; cadena[i] != '\0'; i++)
263     {
264         // Comprobar si estan en mayusculas
265         if (cadena[i] >= 'A')
266         {
267             if (cadena[i] <= 'Z')
268             {
269                 // Si las condiciones son verdaderas se hace la conversion a minusculas
270                 cadena[i] = cadena[i] - ('A' - 'a');
271             }
272         }
273     }
274 }
```

Ln 487, Col 1 Spaces: 4 UTF-8 CRLF {} C++ Win32 Prettier

ACTIVIDADES

EXPLORER

Act95 > RMBA\_ACT7\_02.cpp > ...

```
256
257
258     system("CLS");
259     // Titulo
260     printf("CONVERTIR A MINUSCULAS\n");
261     // Recorre todos los caracteres uno por uno
262     for (int i = 0; cadena[i] != '\0'; i++)
263     {
264         // Comprobar si estan en mayusculas
265         if (cadena[i] >= 'A')
266         {
267             if (cadena[i] <= 'Z')
268             {
269                 // Si las condiciones son verdaderas se hace la conversion a minusculas
270                 cadena[i] = cadena[i] - ('A' - 'a');
271             }
272         }
273         // Imprime la palabra en mayusculas o igual
274         printf("%c", cadena[i]);
275     }
276     printf("\n");
277     getchar();
278
279 // Ejercicio 3 Parte 2
280 // Función que reciba como parámetro una cadena y la convierta a CAPITAL
281 void ej3_pt2(char cadena[])
282 {
283     system("CLS");
284     // Titulo
285     printf("CONVERTIR A CAPITAL\n");
286     // Bandera para indicar que estamos en la primera letra
287     int primera_letra = 1;
288     // Recorre todos los caracteres uno por uno
289     for (int i = 0; cadena[i] != '\0'; i++)
290     {
291         // Comprobar si estan en minusculas
292         if (cadena[i] >= 'a')
293         {
294             if (cadena[i] <= 'z')
295             {
296                 // Si es la primera letra
297                 if (primera_letra)
298                 {
299                     // Convierte en mayUscula
300                     cadena[i] = cadena[i] - ('a' - 'A');
301                     // Ya no estamos en la primera letra
302                     primera_letra = 0;
303                 }
304             }
305         }
306         else
307         {
308             // Imprime la palabra en mayusculas o igual
309             printf("%c", cadena[i]);
310         }
311     }
312     printf("\n");
313     getchar();
314 }
315
316 // Ejercicio 4 Parte 2
317 // Función que reciba como parámetro una cadena y retorne la cantidad de
318 // caracteres que tiene la cadena.
319 int ej4_pt2(char cadena[])
320 {
321     system("CLS");
322     // Titulo
```

Ln 487, Col 1 Spaces: 4 UTF-8 CRLF (1) C++ Win32 Prettier

ACTIVIDADES

EXPLORER

Act95 > RMBA\_ACT7\_02.cpp > ...

```
278
279 // Ejercicio 3 Parte 2
280 // Función que reciba como parámetro una cadena y la convierta a CAPITAL
281 void ej3_pt2(char cadena[])
282 {
283     system("CLS");
284     // Titulo
285     printf("CONVERTIR A CAPITAL\n");
286     // Bandera para indicar que estamos en la primera letra
287     int primera_letra = 1;
288     // Recorre todos los caracteres uno por uno
289     for (int i = 0; cadena[i] != '\0'; i++)
290     {
291         // Comprobar si estan en minusculas
292         if (cadena[i] >= 'a')
293         {
294             if (cadena[i] <= 'z')
295             {
296                 // Si es la primera letra
297                 if (primera_letra)
298                 {
299                     // Convierte en mayUscula
300                     cadena[i] = cadena[i] - ('a' - 'A');
301                     // Ya no estamos en la primera letra
302                     primera_letra = 0;
303                 }
304             }
305         }
306         else
307         {
308             if(cadena[i] == ' ')
309             {
310                 primera_letra = 1;
311             }
312         }
313         // Imprime la palabra en mayusculas o igual
314         printf("%c", cadena[i]);
315     }
316     printf("\n");
317     getchar();
318 }
319
320 // Ejercicio 4 Parte 2
321 // Función que reciba como parámetro una cadena y retorne la cantidad de
322 // caracteres que tiene la cadena.
323 int ej4_pt2(char cadena[])
324 {
325     system("CLS");
326     // Titulo
```

Ln 487, Col 1 Spaces: 4 UTF-8 CRLF (1) C++ Win32 Prettier

ACTIVIDADES

EXPLORER

Act95 > RMBA\_ACT7\_02.cpp > ...

```
322 // Función que reciba como parámetro una cadena y retorne la cantidad de
323 // caracteres que tiene la cadena.
324 int ej4_pt2(char cadena[])
325 {
326     system("CLS");
327     // Titulo
328     printf("CANTIDAD DE CARACTERES\n");
329     // Declarar variable para la longitud/largo
330     int length = 0;
331     // Calcular
332     while (cadena[length] != '\0')
333     {
334         // Incremento
335         length++;
336     }
337     // Imprimir tamaño de la cadena
338     printf("Su tamaño es de: %d\n", length);
339     getch();
340     return 0;
341 }
342
343 // Ejercicio 5 Parte 2
344 // Función que reciba como parámetro una cadena y retorne una cadena con sus
345 // caracteres acomodados de forma inversa (al revés)
346 int ej5_pt2(char cadena[])
347 {
348     system("CLS");
349     // Titulo
350     printf("CADENA INVERSA\n");
351     // Declarar variable para la longitud/largo
352     int length = 0;
353     // Calcular
354     while (cadena[length] != '\0')
355     {
356         // Incrementa longitud
357         length++;
358     }
359     // Recorre todos los caracteres uno por uno
360     for (int i = length - 1; i >= 0; i--)
361     {
362         // Imprime la cadena al revés
363         printf("%c", cadena[i]);
364     }
365     printf("\n");
366     getch();
367     return 0;
368 }
369
370 // Ejercicio 6 Parte 2
371 // Función que reciba como parámetro una cadena y genere una nueva cadena
372 // basada en la original pero sin espacios.
```

Ln 487, Col 1 Spaces: 4 UTF-8 CRLF (1) C++ Win32 Prettier

ACTIVIDADES

EXPLORER

Act95 > RMBA\_ACT7\_02.cpp > ...

```
370 // Ejercicio 6 Parte 2
371 // Función que reciba como parámetro una cadena y genere una nueva cadena
372 // basada en la original pero sin espacios.
373 void ej6_pt2(char cadena[])
374 {
375     system("CLS");
376     // Titulo
377     printf("SIN ESPACIOS\n");
378     // Declarar una nueva cadena para almacenar el resultado sin espacios
379     char nuevaCadena[50];
380     // Inicializar la longitud de la nueva cadena
381     int nuevaLongitud = 0;
382     // Recorre la cadena original
383     for (int i = 0; cadena[i] != '\0'; i++)
384     {
385         if (cadena[i] != ' ')
386         {
387             // Copia los caracteres que no sean espacios a la nueva cadena
388             nuevaCadena[nuevaLongitud] = cadena[i];
389             nuevaLongitud++;
390         }
391     }
392     // Agrega el caracter nulo al final de la nueva cadena para finalizarla
393     nuevaCadena[nuevaLongitud] = '\0';
394     // Imprime la nueva cadena sin espacios
395     printf("%s\n", nuevaCadena);
396     getch();
397 }
398
399 // Ejercicio 7 Parte 2
400 // Función que sirva para leer una cadena y solo permita caracteres alfabéticos
401 // (A...Z) y el espacio, donde una cadena no puede comenzar o terminar con espacio,
402 // no debe tener dos espacios seguidos. retornar la cadena ya sea como parámetro o variable.
403 int ej7_pt2(char cadena[])
404 {
405     system("CLS");
406     // Declarar variables booleanas para seguir el estado de la cadena
407     bool tieneEspaciosAlPrincipio = false;
408     bool terminaConEspacios = false;
409     bool tieneNumeros = false;
410     int i = 0;
411     while (cadena[i] != '\0')
412     {
413         // Para los espacios
414         if (cadena[i] == ' ')
415         {
416             if (i == 0)
417             {
418                 tieneEspaciosAlPrincipio = true; // Tiene espacios al principio
419             }
420         }
421     }
422 }
```

Ln 487, Col 1 Spaces: 4 UTF-8 CRLF (1) C++ Win32 Prettier

