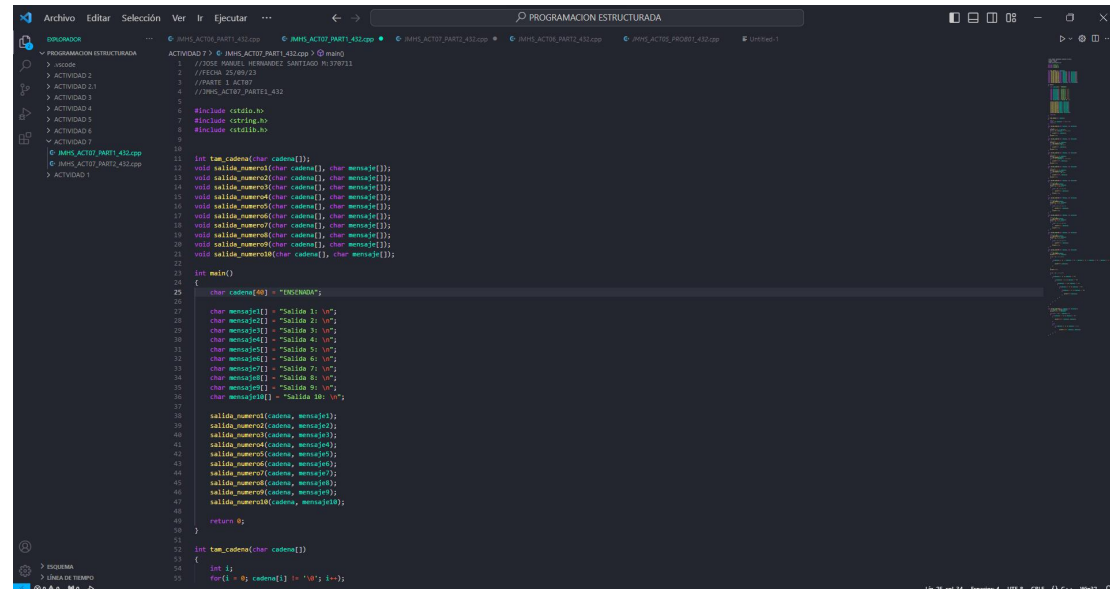
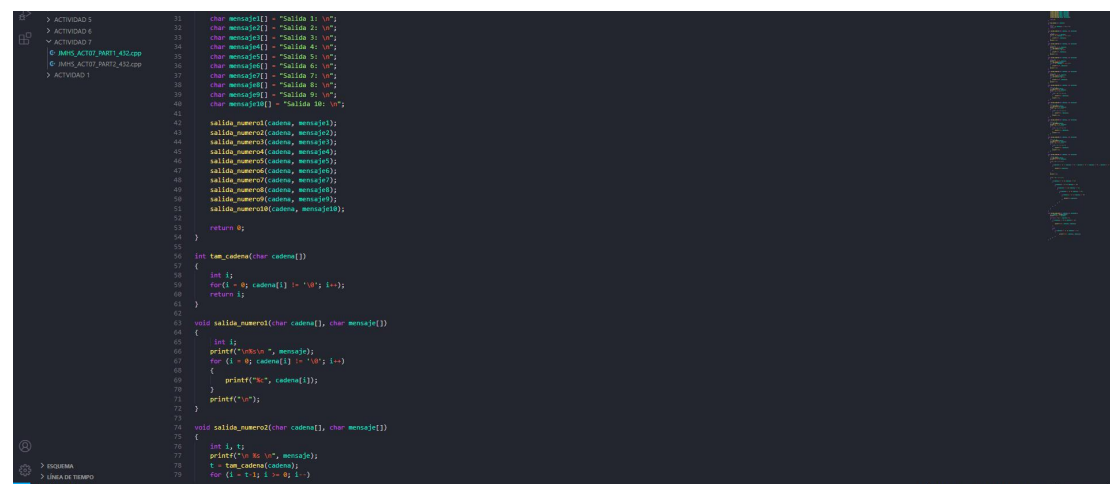


ANEXOS
ACTIVIDAD 7
PERTE 1 Y 2
JOSE MANUEL HERNANDEZ SANTIAGO M:370711
PROGRAMAION ESTRUCTURADA
GRUPO:432

PARTE 1:



```
1 //JOSE MANUEL HERNANDEZ SANTIAGO M:370711
2 //FECHA 25/09/23
3 //PARTE 1 ACT07
4 //PROG. ACT07_PARTE1_432
5
6 #include <stdio.h>
7 #include <string.h>
8 #include <stdlib.h>
9
10
11 int tam_cadena(char cadena[]);
12 void salida_numero1(char cadena[], char mensaje[]);
13 void salida_numero2(char cadena[], char mensaje[]);
14 void salida_numero3(char cadena[], char mensaje[]);
15 void salida_numero4(char cadena[], char mensaje[]);
16 void salida_numero5(char cadena[], char mensaje[]);
17 void salida_numero6(char cadena[], char mensaje[]);
18 void salida_numero7(char cadena[], char mensaje[]);
19 void salida_numero8(char cadena[], char mensaje[]);
20 void salida_numero9(char cadena[], char mensaje[]);
21 void salida_numero10(char cadena[], char mensaje[]);
22
23 int main()
24 {
25     char cadena[40] = "INCONCNUH";
26
27     char mensaje1[] = "Salida 1: v";
28     char mensaje2[] = "Salida 2: v";
29     char mensaje3[] = "Salida 3: v";
30     char mensaje4[] = "Salida 4: v";
31     char mensaje5[] = "Salida 5: v";
32     char mensaje6[] = "Salida 6: v";
33     char mensaje7[] = "Salida 7: v";
34     char mensaje8[] = "Salida 8: v";
35     char mensaje9[] = "Salida 9: v";
36     char mensaje10[] = "Salida 10: v";
37
38     salida_numero1(cadena, mensaje1);
39     salida_numero2(cadena, mensaje2);
40     salida_numero3(cadena, mensaje3);
41     salida_numero4(cadena, mensaje4);
42     salida_numero5(cadena, mensaje5);
43     salida_numero6(cadena, mensaje6);
44     salida_numero7(cadena, mensaje7);
45     salida_numero8(cadena, mensaje8);
46     salida_numero9(cadena, mensaje9);
47     salida_numero10(cadena, mensaje10);
48
49     return 0;
50 }
51
52 int tam_cadena(char cadena[])
53 {
54     int i;
55     for(i = 0; cadena[i] != '\0'; i++);
56 }
```



```
31     char mensaje1[] = "Salida 1: v";
32     char mensaje2[] = "Salida 2: v";
33     char mensaje3[] = "Salida 3: v";
34     char mensaje4[] = "Salida 4: v";
35     char mensaje5[] = "Salida 5: v";
36     char mensaje6[] = "Salida 6: v";
37     char mensaje7[] = "Salida 7: v";
38     char mensaje8[] = "Salida 8: v";
39     char mensaje9[] = "Salida 9: v";
40     char mensaje10[] = "Salida 10: v";
41
42     salida_numero1(cadena, mensaje1);
43     salida_numero2(cadena, mensaje2);
44     salida_numero3(cadena, mensaje3);
45     salida_numero4(cadena, mensaje4);
46     salida_numero5(cadena, mensaje5);
47     salida_numero6(cadena, mensaje6);
48     salida_numero7(cadena, mensaje7);
49     salida_numero8(cadena, mensaje8);
50     salida_numero9(cadena, mensaje9);
51     salida_numero10(cadena, mensaje10);
52
53     return 0;
54 }
55
56 int tam_cadena(char cadena[])
57 {
58     int i;
59     for(i = 0; cadena[i] != '\0'; i++);
60     return i;
61 }
62
63 void salida_numero1(char cadena[], char mensaje[])
64 {
65     int i;
66     printf("%s\n", mensaje);
67     for (i = 0; cadena[i] != '\0'; i++)
68     {
69         printf("%c", cadena[i]);
70     }
71     printf("\n");
72 }
73
74 void salida_numero2(char cadena[], char mensaje[])
75 {
76     int i, t;
77     printf("%s\n", mensaje);
78     t = tam_cadena(cadena);
79     for (i = 1; i <= t; i++)
80 }
```

```
Archivo Editar Selección Ver Ir Ejecutar PROGRAMACION ESTRUCTURADA
PROYECTO
PROYECTO ESTRUCTURADO
ACTIVIDAD 7
void salida_numeros(char cadena[], char mensaje[])
{
    int i, j;
    printf("%s\n", mensaje);
    t = tam_cadena(cadena);
    for (i = 0; i < t; i++)
    {
        printf("%c\n", cadena[i]);
    }
    printf("\n");
}

void salida_numeros(char cadena[], char mensaje[])
{
    int i, j;
    printf("%s\n", mensaje);
    t = tam_cadena(cadena);
    for (i = t-1; i >= 0; i--)
    {
        printf("%c\n", cadena[i]);
    }
    printf("\n");
}

void salida_numeros(char cadena[], char mensaje[])
{
    int i, j, k;
    t = tam_cadena(cadena);
    printf("%s\n", mensaje);
    for (i = 0; i < t; i++)
    {
        for (j = 0; j <= i; j++)
        {
            printf("%c", cadena[i]);
        }
        printf("\n");
    }
}

void salida_numeros(char cadena[], char mensaje[])
{
    int i, j, k;
    t = tam_cadena(cadena);
    printf("%s\n", mensaje);
    for (i = 0; i < t; i++)
    {
        for (j = t; j >= i; j--)
        {
            printf("%c", cadena[i]);
        }
        printf("\n");
    }
}

void salida_numeros(char cadena[], char mensaje[])
{
    int i, j, k;
    t = tam_cadena(cadena);
    printf("%s\n", mensaje);
    for (i = 0; i < t; i++)
    {
        if (cadena[i] == 'a' || cadena[i] == 'A' || cadena[i] == 'e' || cadena[i] == 'E' || cadena[i] == 'i' || cadena[i] == 'I' || cadena[i] == 'o' || cadena[i] == 'O' || cadena[i] == 'u' || cadena[i] == 'U')
        {
            printf("%c", cadena[i]);
        }
    }
    printf("\n");
}

for (i = 0; i < t; i++)
{
    if (cadena[i] != 'a' && cadena[i] != 'A')
    {
        if (cadena[i] != 'e' && cadena[i] != 'E')
        {
            if (cadena[i] != 'i' && cadena[i] != 'I')
            {
                if (cadena[i] != 'o' && cadena[i] != 'O')
                {
                    if (cadena[i] != 'u' && cadena[i] != 'U')
                    {
                        printf("%c", cadena[i]);
                    }
                }
            }
        }
    }
}
printf("\n");
}
```

```
Archivo Editar Selección Ver Ir Ejecutar PROGRAMACION ESTRUCTURADA
PROYECTO
PROYECTO ESTRUCTURADO
ACTIVIDAD 7
void salida_numeros(char cadena[], char mensaje[])
{
    int i, j;
    printf("%s\n", mensaje);
    t = tam_cadena(cadena);
    for (i = 0; i < t; i++)
    {
        for (j = i; j <= t; j++)
        {
            printf("%c", cadena[i]);
        }
        printf("\n");
    }
}

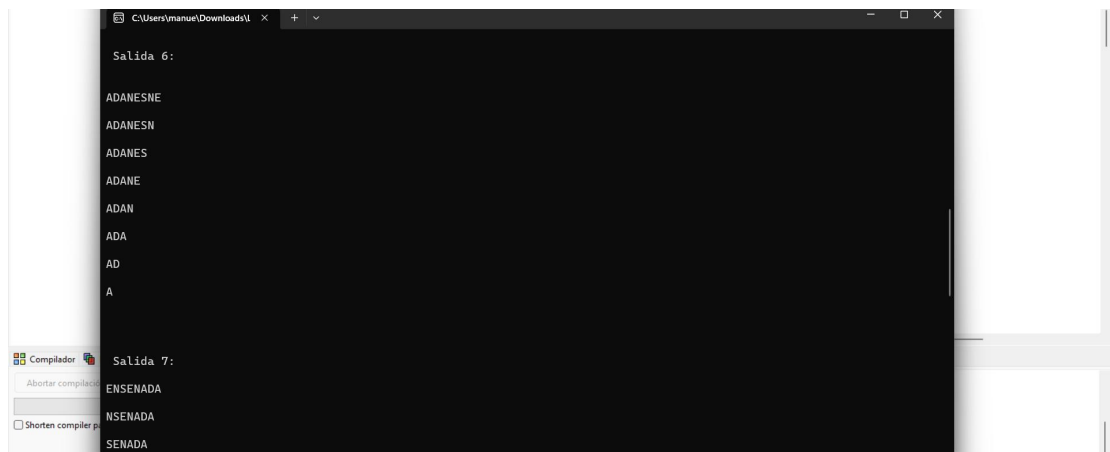
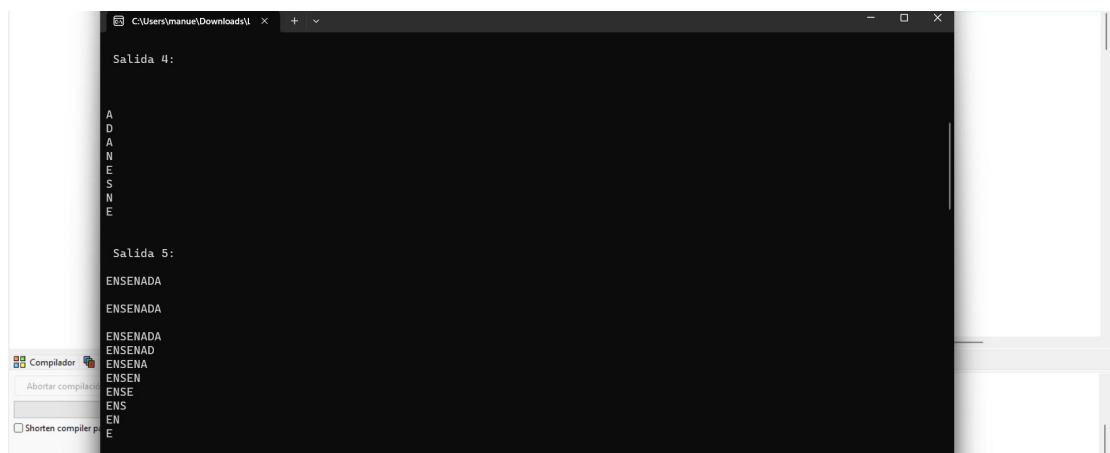
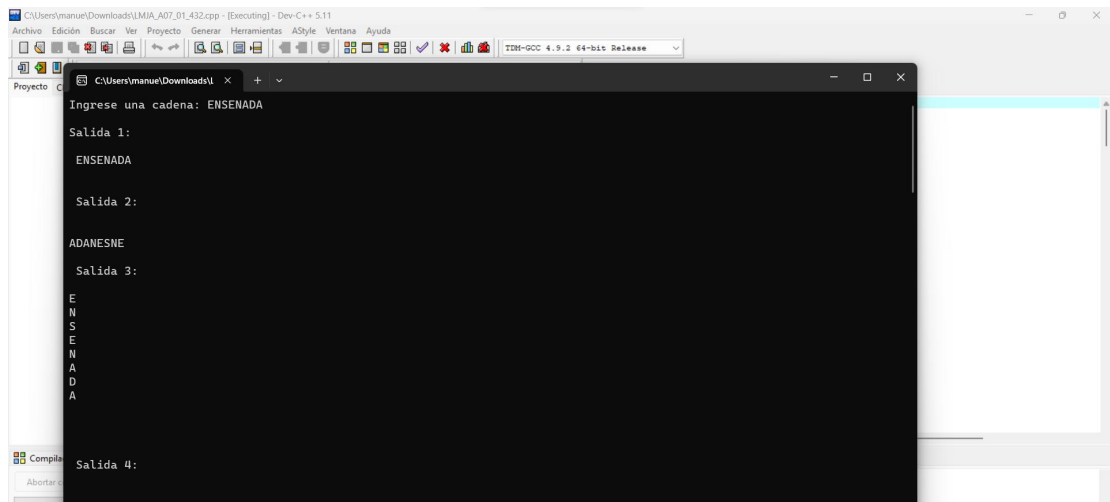
void salida_numeros(char cadena[], char mensaje[])
{
    int i, j, k;
    t = tam_cadena(cadena);
    printf("%s\n", mensaje);
    for (i = 0; i < t; i++)
    {
        for (j = t; j >= i; j--)
        {
            printf("%c", cadena[i]);
        }
        printf("\n");
    }
}

void salida_numeros(char cadena[], char mensaje[])
{
    int i, j, k;
    t = tam_cadena(cadena);
    printf("%s\n", mensaje);
    for (i = 0; i < t; i++)
    {
        if (cadena[i] != 'a' && cadena[i] != 'A')
        {
            if (cadena[i] != 'e' && cadena[i] != 'E')
            {
                if (cadena[i] != 'i' && cadena[i] != 'I')
                {
                    if (cadena[i] != 'o' && cadena[i] != 'O')
                    {
                        if (cadena[i] != 'u' && cadena[i] != 'U')
                        {
                            printf("%c", cadena[i]);
                        }
                    }
                }
            }
        }
    }
    printf("\n");
}

for (i = 0; i < t; i++)
{
    if (cadena[i] != 'a' && cadena[i] != 'A')
    {
        if (cadena[i] != 'e' && cadena[i] != 'E')
        {
            if (cadena[i] != 'i' && cadena[i] != 'I')
            {
                if (cadena[i] != 'o' && cadena[i] != 'O')
                {
                    if (cadena[i] != 'u' && cadena[i] != 'U')
                    {
                        printf("%c", cadena[i]);
                    }
                }
            }
        }
    }
}
printf("\n");
}
```

```
Archivo Editar Selección Ver Ir Ejecutar PROGRAMACION ESTRUCTURADA
PROYECTO
PROYECTO ESTRUCTURADO
ACTIVIDAD 7
void salida_numeros(char cadena[], char mensaje[])
{
    int i, j;
    printf("%s\n", mensaje);
    t = tam_cadena(cadena);
    for (i = 0; i < t; i++)
    {
        if (cadena[i] != 'a' && cadena[i] != 'A')
        {
            if (cadena[i] != 'e' && cadena[i] != 'E')
            {
                if (cadena[i] != 'i' && cadena[i] != 'I')
                {
                    if (cadena[i] != 'o' && cadena[i] != 'O')
                    {
                        if (cadena[i] != 'u' && cadena[i] != 'U')
                        {
                            printf("%c", cadena[i]);
                        }
                    }
                }
            }
        }
    }
    printf("\n");
}

for (i = 0; i < t; i++)
{
    if (cadena[i] != 'a' && cadena[i] != 'A')
    {
        if (cadena[i] != 'e' && cadena[i] != 'E')
        {
            if (cadena[i] != 'i' && cadena[i] != 'I')
            {
                if (cadena[i] != 'o' && cadena[i] != 'O')
                {
                    if (cadena[i] != 'u' && cadena[i] != 'U')
                    {
                        printf("%c", cadena[i]);
                    }
                }
            }
        }
    }
}
printf("\n");
}
```



```
C:\Users\manuel\Downloads\...
Salida 7:
ENSENADA
NSENADA
SENADA
ENADA
NADA
ADA
DA
A

Salida 8:
ADANESNE
ADANESN
ADANES
ADANE
ADAN
ADA
AD
A
```

```
ADANESNE
ADANESN
ADANES
ADANE
ADAN
ADA
AD
A

Salida 9:
EEAA
NSND

Salida 10:
EENNSSEENNAADAA
-----
Process exited after 3.887 seconds with return value 0
Presione una tecla para continuar . . . |
```

PARTE 2:

```
PROGRAMACION ESTRUCTURADA
Archivo Editar Selecci3n Ver Ir Ejecutar ...
...
ACTIVIDAD 7 > C:\Users\manuel\Downloads\...
1 //JOS3 MARCELO HERNANDEZ SANTILLANO M:3787311
2 //FECHA: 25/09/23
3 //PARTE 2 ACT07
4 //JPMH_ACT07_PART2_432.cpp
5
6 #include <stdio.h>
7 #include <stdlib.h>
8 #include <string.h>
9
10 void menu();
11
12 void mayusculas(char cadena[50]);
13 void minusculas(char cadena[50]);
14 void capital(char cadena[50]);
15 void tamam(char cadena[50]);
16 void invertir(char cadena[50]);
17 void espacios(char cadena[50]);
18 void permutar(char cadena[50]);
19 void todas(char cadena[50]);
20 void palindromo(char cadena[50]);
21
22 int main()
23 {
24     menu();
25 }
26
27 //FUNCION DE MENU PRINCIPAL //
28
29 void menu()
30 {
31     int decision, i;
32     char cadena[50];
33
34     printf("1.- Convertir todos los caracteres a mayusculas\n");
35     printf("2.- Convertir todos los caracteres a minusculas\n");
36     printf("3.- Convertir cadena en la cadena CAPITAL\n");
37     printf("4.- Contar la cantidad de caracteres de la cadena\n");
38     printf("5.- Voltear cadena de caracteres al reves\n");
39     printf("6.- Eliminar los espacios de la cadena\n");
40     printf("7.- Validar la cadena para saber si es palindromo o no\n");
41     printf("8.- Realizar 5 de las opciones anteriores juntas\n");
42     printf("9.- Imprimir si la cadena es palindromo o no\n");
43     printf("10.- Salir de la cadena que quieras\n");
44     scanf("%d", &decision);
45
46     printf("Dame una cadena de caracteres: ");
47     fflush(stdin);
48     fgets(cadena, 50, stdin);
49
50     switch(decision)
51     {
52         case 1:
53             printf("Ingresa la opcion 1");
54             mayusculas(cadena);
55     }
56 }
```

```
PROGRAMACION ESTRUCTURADA

ACTIVIDAD 7 > @ INHS_ACT07_PART2_432.cpp
103
104
105 void mayusculas(char cadena[60])
106 {
107     int i;
108     printf("\n");
109     for(i = 0; cadena[i] != '\0'; i++)
110     {
111         if (cadena[i] == 97)
112         {
113             if (cadena[i] <= 122)
114             {
115                 cadena[i] = cadena[i] - 32;
116                 printf("%c", cadena[i]);
117             }
118         }
119     }
120 }
121
122 void minusculas(char cadena[60])
123 {
124     int i;
125     printf("\n");
126     for(i = 0; cadena[i] != '\0'; i++)
127     {
128         if (cadena[i] == 65)
129         {
130             if (cadena[i] <= 90)
131             {
132                 cadena[i] = cadena[i] + 32;
133                 printf("%c", cadena[i]);
134             }
135         }
136     }
137 }
138
139 void capital(char cadena[60])
140 {
141     int i;
142     printf("\n");
143     cadena[0] = 67;
144     cadena[1] = 68;
145     cadena[2] = 84;
146     cadena[3] = 85;
147     cadena[4] = 86;
148     cadena[5] = 65;
149     cadena[6] = 76;
150     cadena[7] = '\0';
151     for(i = 0; cadena[i] != '\0'; i++)
152     {
153         printf("%c", cadena[i]);
154     }
155 }
156
157 int taman(char cadena[60])
158 {
159     printf("\n");
160     int i;
161     for(i = 0; cadena[i] != '\0'; i++)
162     {
163         return i;
164     }
165 }
166
167 void Inversa(char cadena[60])
168 {
169     int i, j;
170     printf("\n");
171     i = taman(cadena);
172     for (i = i-1; i >= 0; i--)
173     {
174         printf("%c", cadena[i]);
175     }
176 }
177
178 void espacios(char cadena[60])
179 {
180     int i, t, aux, espacio;
181     printf("\n");
182     t = taman(cadena);
183     for(i = 0; i < t; i++)
184     {
185         if (cadena[i] == 32)
186         {
187             espacio = 1;
188             aux = 0;
189             while (cadena[aux] == 32 || aux < t - 1)
190             {
191                 aux++;
192                 cadena[espacio] = cadena[aux];
193                 cadena[aux] = 32;
194             }
195         }
196     }
197     for(i = 0; cadena[i] != '\0'; i++)
198     {
199         printf("%c", cadena[i]);
200     }
201 }
```

```
PROGRAMACION ESTRUCTURADA

ACTIVIDAD 7 > @ INHS_ACT07_PART2_432.cpp
103
104
105 void mayusculas(char cadena[60])
106 {
107     int i;
108     printf("\n");
109     for(i = 0; cadena[i] != '\0'; i++)
110     {
111         if (cadena[i] == 97)
112         {
113             if (cadena[i] <= 122)
114             {
115                 cadena[i] = cadena[i] - 32;
116                 printf("%c", cadena[i]);
117             }
118         }
119     }
120 }
121
122 void minusculas(char cadena[60])
123 {
124     int i;
125     printf("\n");
126     for(i = 0; cadena[i] != '\0'; i++)
127     {
128         if (cadena[i] == 65)
129         {
130             if (cadena[i] <= 90)
131             {
132                 cadena[i] = cadena[i] + 32;
133                 printf("%c", cadena[i]);
134             }
135         }
136     }
137 }
138
139 void capital(char cadena[60])
140 {
141     int i;
142     printf("\n");
143     cadena[0] = 67;
144     cadena[1] = 68;
145     cadena[2] = 84;
146     cadena[3] = 85;
147     cadena[4] = 86;
148     cadena[5] = 65;
149     cadena[6] = 76;
150     cadena[7] = '\0';
151     for(i = 0; cadena[i] != '\0'; i++)
152     {
153         printf("%c", cadena[i]);
154     }
155 }
156
157 int taman(char cadena[60])
158 {
159     printf("\n");
160     int i;
161     for(i = 0; cadena[i] != '\0'; i++)
162     {
163         return i;
164     }
165 }
166
167 void Inversa(char cadena[60])
168 {
169     int i, j;
170     printf("\n");
171     i = taman(cadena);
172     for (i = i-1; i >= 0; i--)
173     {
174         printf("%c", cadena[i]);
175     }
176 }
177
178 void espacios(char cadena[60])
179 {
180     int i, t, aux, espacio;
181     printf("\n");
182     t = taman(cadena);
183     for(i = 0; i < t; i++)
184     {
185         if (cadena[i] == 32)
186         {
187             espacio = 1;
188             aux = 0;
189             while (cadena[aux] == 32 || aux < t - 1)
190             {
191                 aux++;
192                 cadena[espacio] = cadena[aux];
193                 cadena[aux] = 32;
194             }
195         }
196     }
197     for(i = 0; cadena[i] != '\0'; i++)
198     {
199         printf("%c", cadena[i]);
200     }
201 }
```

```
PROGRAMACION ESTRUCTURADA

ACTIVIDAD 7 > @ INHS_ACT07_PART2_432.cpp
103
104
105 void mayusculas(char cadena[60])
106 {
107     int i;
108     printf("\n");
109     for(i = 0; cadena[i] != '\0'; i++)
110     {
111         if (cadena[i] == 97)
112         {
113             if (cadena[i] <= 122)
114             {
115                 cadena[i] = cadena[i] - 32;
116                 printf("%c", cadena[i]);
117             }
118         }
119     }
120 }
121
122 void minusculas(char cadena[60])
123 {
124     int i;
125     printf("\n");
126     for(i = 0; cadena[i] != '\0'; i++)
127     {
128         if (cadena[i] == 65)
129         {
130             if (cadena[i] <= 90)
131             {
132                 cadena[i] = cadena[i] + 32;
133                 printf("%c", cadena[i]);
134             }
135         }
136     }
137 }
138
139 void capital(char cadena[60])
140 {
141     int i;
142     printf("\n");
143     cadena[0] = 67;
144     cadena[1] = 68;
145     cadena[2] = 84;
146     cadena[3] = 85;
147     cadena[4] = 86;
148     cadena[5] = 65;
149     cadena[6] = 76;
150     cadena[7] = '\0';
151     for(i = 0; cadena[i] != '\0'; i++)
152     {
153         printf("%c", cadena[i]);
154     }
155 }
156
157 int taman(char cadena[60])
158 {
159     printf("\n");
160     int i;
161     for(i = 0; cadena[i] != '\0'; i++)
162     {
163         return i;
164     }
165 }
166
167 void Inversa(char cadena[60])
168 {
169     int i, j;
170     printf("\n");
171     i = taman(cadena);
172     for (i = i-1; i >= 0; i--)
173     {
174         printf("%c", cadena[i]);
175     }
176 }
177
178 void espacios(char cadena[60])
179 {
180     int i, t, aux, espacio;
181     printf("\n");
182     t = taman(cadena);
183     for(i = 0; i < t; i++)
184     {
185         if (cadena[i] == 32)
186         {
187             espacio = 1;
188             aux = 0;
189             while (cadena[aux] == 32 || aux < t - 1)
190             {
191                 aux++;
192                 cadena[espacio] = cadena[aux];
193                 cadena[aux] = 32;
194             }
195         }
196     }
197     for(i = 0; cadena[i] != '\0'; i++)
198     {
199         printf("%c", cadena[i]);
200     }
201 }
```

```
PROGRAMACION ESTRUCTURADA
ACTIVIDAD 7 > @ JAVHS_ACTO7_PART1_432.cpp
198
199
200 }
201
202 char permutar(char cadena[60])
203 {
204     int i, j;
205     t = tam(cadena);
206     printf("t:");
207     for(i = 0; cadena[i] != '\0'; i++)
208     {
209         if (cadena[i] == ' ' || cadena[i] == ' ')
210         {
211             printf("NO PERMITIDA, debe ser una cadena sin espacios al inicio ni al final");
212             cadena[i] = '\0';
213         }
214         if (cadena[i] == 32 && cadena[i+1] == 32)
215         {
216             printf("NO PERMITIDA, debe ser cadena sin espacios dobles");
217             cadena[i] = '\0';
218         }
219         if (cadena[i] != 32 && (cadena[i] < 65 || cadena[i] > 122))
220         {
221             printf("NO PERMITIDA, solo caracteres alfabeticos");
222             cadena[i] = '\0';
223         }
224     }
225     printf("\n");
226     for(i = 0; cadena[i] != '\0'; i++)
227     {
228         printf("%c", cadena[i]);
229     }
230     return cadena[60];
231 }
232
233 void todos(char cadena[60])
234 {
235     int i;
236     char original[60], original2[60], original3[60], original4[60];
237     for(i = 0; cadena[i] != '\0'; i++)
238     {
239         original[i] = cadena[i];
240         original2[i] = cadena[i];
241         original3[i] = cadena[i];
242         original4[i] = cadena[i];
243     }
244     eliminar(cadena);
245     mayusculas(original);
246     minusculas(original2);
247     inversa(original3);
248     capital(original4);
249 }
250
251 void palindromo(char cadena[])
252 {
253     int i, j, k, g, h;
254     char es;
255     int t = tam(cadena);
256     for(i = 0; i <= t/2; i++)
257     {
258         if (cadena[i] != '\0' && cadena[t-i-1] != '\0')
259         {
260             printf("Digita caracteres, no numeros");
261             i = t;
262         }
263         if (cadena[i] != '\0')
264         {
265             if (cadena[i] == ' ')
266             {
267                 printf("No utilizar dobles espacios en la cadena");
268                 i = t;
269             }
270             else
271             {
272                 if (k == 1)
273                 {
274                     for(j = 0; j <= t/2; j++)
275                     {
276                         if (cadena[j] != '\0' && cadena[t-j-1] != '\0')
277                         {
278                             if (cadena[j] != '\0' && cadena[t-j-1] != '\0')
279                             {
280                                 if (cadena[j] == 'A' && cadena[t-j-1] == 'a')
281                                 {
282                                     es = '1';
283                                 }
284                                 else
285                                 {
286                                     printf("Digita la cadena solo con mayusculas\n");
287                                     i = t;
288                                 }
289                             }
290                             else
291                             {
292                                 printf("No utilizar dobles espacios en la cadena");
293                                 i = t;
294                             }
295                         }
296                     }
297                 }
298             }
299         }
300     }
301     if (es == '1')
302     {
303         int tam = tam(cadena);
304         for(i = 0, j = tam - 1; i < j; i++, j--)
305         {
306             if (cadena[i] != '\0' && cadena[j] != '\0')
307             {
308                 es = '0';
309             }
310             else
311             {
312                 es = '1';
313             }
314         }
315         if (es == '0')
316         {
317             printf("La cadena digitada NO es un palindromo.\n");
318         }
319         else
320         {
321             printf("La cadena digitada SI es un palindromo.\n");
322         }
323     }
324 }
```

```
PROGRAMACION ESTRUCTURADA
ACTIVIDAD 7 > @ JAVHS_ACTO7_PART1_432.cpp
242
243 {
244     original[i] = cadena[i];
245     original2[i] = cadena[i];
246     original3[i] = cadena[i];
247     original4[i] = cadena[i];
248 }
249
250 eliminar(cadena);
251 mayusculas(original);
252 minusculas(original2);
253 inversa(original3);
254 capital(original4);
255 }
256
257 void palindromo(char cadena[])
258 {
259     int i, j, k, g, h;
260     char es;
261     int t = tam(cadena);
262     for(i = 0; i <= t/2; i++)
263     {
264         if (cadena[i] != '\0' && cadena[t-i-1] != '\0')
265         {
266             printf("Digita caracteres, no numeros");
267             i = t;
268         }
269         if (cadena[i] != '\0')
270         {
271             if (cadena[i] == ' ')
272             {
273                 printf("No utilizar dobles espacios en la cadena");
274                 i = t;
275             }
276             else
277             {
278                 if (k == 1)
279                 {
280                     for(j = 0; j <= t/2; j++)
281                     {
282                         if (cadena[j] != '\0' && cadena[t-j-1] != '\0')
283                         {
284                             if (cadena[j] != '\0' && cadena[t-j-1] != '\0')
285                             {
286                                 if (cadena[j] == 'A' && cadena[t-j-1] == 'a')
287                                 {
288                                     es = '1';
289                                 }
290                                 else
291                                 {
292                                     printf("Digita la cadena solo con mayusculas\n");
293                                     i = t;
294                                 }
295                             }
296                             else
297                             {
298                                 printf("No utilizar dobles espacios en la cadena");
299                                 i = t;
300                             }
301                         }
302                     }
303                 }
304             }
305         }
306     }
307     if (es == '1')
308     {
309         int tam = tam(cadena);
310         for(i = 0, j = tam - 1; i < j; i++, j--)
311         {
312             if (cadena[i] != '\0' && cadena[j] != '\0')
313             {
314                 es = '0';
315             }
316             else
317             {
318                 es = '1';
319             }
320         }
321         if (es == '0')
322         {
323             printf("La cadena digitada NO es un palindromo.\n");
324         }
325         else
326         {
327             printf("La cadena digitada SI es un palindromo.\n");
328         }
329     }
330 }
```

```
PROGRAMACION ESTRUCTURADA
ACTIVIDAD 7 > @ JAVHS_ACTO7_PART1_432.cpp
279
280 {
281     if (cadena[i] != '\0' && cadena[t-i-1] != '\0')
282     {
283         es = '1';
284     }
285     else
286     {
287         printf("Digita la cadena solo con mayusculas\n");
288         i = t;
289     }
290     else
291     {
292         printf("No utilizar dobles espacios en la cadena");
293         i = t;
294     }
295     }
296 }
297
298 if (es == '1')
299 {
300     int tam = tam(cadena);
301     for(i = 0, j = tam - 1; i < j; i++, j--)
302     {
303         if (cadena[i] != '\0' && cadena[j] != '\0')
304         {
305             es = '0';
306         }
307         else
308         {
309             es = '1';
310         }
311     }
312     if (es == '0')
313     {
314         printf("La cadena digitada NO es un palindromo.\n");
315     }
316     else
317     {
318         printf("La cadena digitada SI es un palindromo.\n");
319     }
320 }
321
322 }
```

```
Proyecto C: C:\Users\manuel\Downloads\l...
1.- Convertir todos los caracteres a mayusculas
2.- Convertir todos los caracteres a minusculas
3.- Convertir cadena en la cadena CAPITAL
4.- Contar la cantidad de caracteres de la cadena
5.- Voltrear cadena de caracteres al reves
6.- Eliminar los espacios de la cadena
7.- Validar la cadena para saber si es permitida o no
8.- Realizar 5 de las opciones anteriores juntas
9.- Imprimir si la cadena es palindromo o no

ingresa el numero de la cadena que quieras:
1
Dame una cadena de caracteres: punta colonet

Elegiste la opcion 1
PUNTACOLONET
-----
Process exited after 55.01 seconds with return value 0
Presione una tecla para continuar . . . |
```

```
1.- Convertir todos los caracteres a mayusculas
2.- Convertir todos los caracteres a minusculas
3.- Convertir cadena en la cadena CAPITAL
4.- Contar la cantidad de caracteres de la cadena
5.- Voltrear cadena de caracteres al reves
6.- Eliminar los espacios de la cadena
7.- Validar la cadena para saber si es permitida o no
8.- Realizar 5 de las opciones anteriores juntas
9.- Imprimir si la cadena es palindromo o no

ingresa el numero de la cadena que quieras:
2
Dame una cadena de caracteres: PUNTA COLONET

Elegiste la opcion 2
punta colonet
-----
Process exited after 10.21 seconds with return value 0
Presione una tecla para continuar . . . |
```

```
C:\Users\manuel\Downloads\l...
1.- Convertir todos los caracteres a mayusculas
2.- Convertir todos los caracteres a minusculas
3.- Convertir cadena en la cadena CAPITAL
4.- Contar la cantidad de caracteres de la cadena
5.- Voltrear cadena de caracteres al reves
6.- Eliminar los espacios de la cadena
7.- Validar la cadena para saber si es permitida o no
8.- Realizar 5 de las opciones anteriores juntas
9.- Imprimir si la cadena es palindromo o no

ingresa el numero de la cadena que quieras:
5
Dame una cadena de caracteres: PUNTA COLONET

Elegiste la opcion 5

TENOLOC ATNUP
-----
Process exited after 11.88 seconds with return value 0
Presione una tecla para continuar . . . |
```