

# UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA



Ingenieria en Computacion

Programación Estructurada

## **Anexos: Actividad 7**

\*\*\*

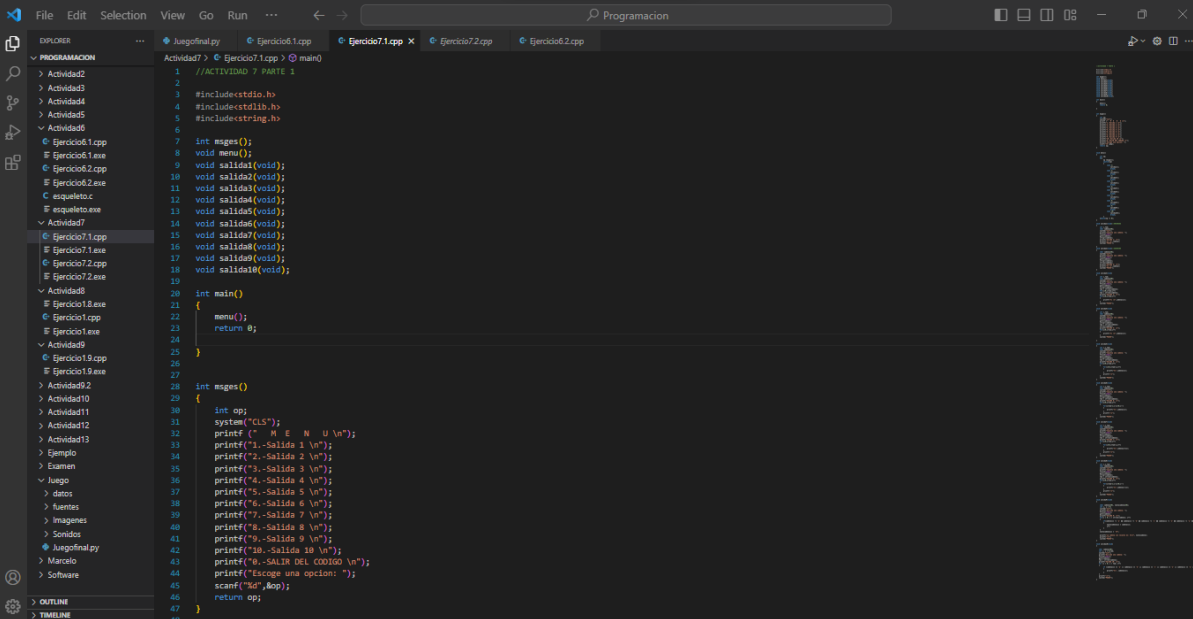
ALUMNO: Isai Alexis Arredondo Urbalejo

MATRÍCULA: 368747

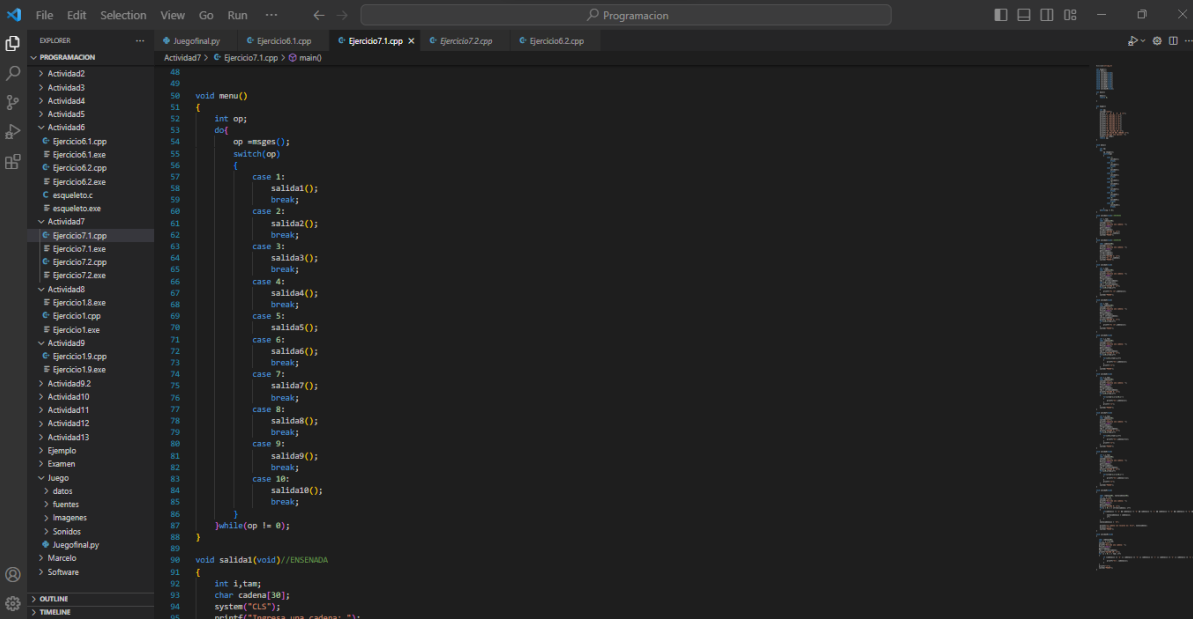
GRUPO: 932

PROFESOR: Nuñez Yepi Pedro

# Parte 1



```
1 //ACTIVIDAD 7 PARTE 1
2
3 #include<stdio.h>
4 #include<stdlib.h>
5 #include<string.h>
6
7 int mages();
8 void menu();
9 void salida1(void);
10 void salida2(void);
11 void salida3(void);
12 void salida4(void);
13 void salida5(void);
14 void salida6(void);
15 void salida7(void);
16 void salida8(void);
17 void salida9(void);
18 void salida10(void);
19
20 int main()
21 {
22     menu();
23     return 0;
24 }
25
26
27 int mages()
28 {
29     int op;
30     system("CLS");
31     printf("E N U \n");
32     printf("1.-Salida 1 \n");
33     printf("2.-Salida 2 \n");
34     printf("3.-Salida 3 \n");
35     printf("4.-Salida 4 \n");
36     printf("5.-Salida 5 \n");
37     printf("6.-Salida 6 \n");
38     printf("7.-Salida 7 \n");
39     printf("8.-Salida 8 \n");
40     printf("9.-Salida 9 \n");
41     printf("10.-Salida 10 \n");
42     printf("0.-SALIR del CODIGO \n");
43     printf("Escriba una opcion: ");
44     scanf("%d",&op);
45     return op;
46 }
47
48
```



```
48
49 void menu()
50 {
51     int op;
52     do{
53         op = mages();
54         switch(op)
55         {
56             case 1:
57                 salida1();
58                 break;
59             case 2:
60                 salida2();
61                 break;
62             case 3:
63                 salida3();
64                 break;
65             case 4:
66                 salida4();
67                 break;
68             case 5:
69                 salida5();
70                 break;
71             case 6:
72                 salida6();
73                 break;
74             case 7:
75                 salida7();
76                 break;
77             case 8:
78                 salida8();
79                 break;
80             case 9:
81                 salida9();
82                 break;
83             case 10:
84                 salida10();
85                 break;
86         }
87     }while(op != 0);
88 }
89
90 void salida1(void)//ENSEÑADA
91 {
92     int i;
93     char cadena[30];
94     system("CLS");
95     printf("Escriba una cadena: ");
96 }
97
98
```

The screenshot shows the Visual Studio Code editor with a C++ project named 'Programacion'. The Explorer sidebar on the left shows a file tree with folders like 'Actividad2' through 'Actividad13', and files like 'Ejercicio6.1.cpp', 'Ejercicio6.1.exe', 'Ejercicio7.1.cpp', 'Ejercicio7.2.cpp', 'Ejercicio7.2.exe', 'Ejercicio8.1.exe', 'Ejercicio9.1.cpp', 'Ejercicio9.1.exe', 'Ejercicio9.2', 'Ejercicio10', 'Ejercicio11', 'Ejercicio12', 'Ejercicio13', 'Ejemplo', 'Examen', 'Juego', 'datos', 'fuentes', 'imagenes', 'Sonidos', 'JuegoFinal.py', 'Marcelo', 'Software'. The main editor window shows the code for 'Ejercicio7.1.cpp' at line 100. The code includes headers for `system`, `printf`, `fflush`, `gets`, `strupr`, `strlen`, `strrev`, and `strcat`. It defines a function `void salida2(void)` that prompts the user to 'Ingresar una cadena', reads the input, converts it to uppercase, reverses it, and prints it. It also defines a function `void salida1(void)` that prompts for a string, calculates its length, and prints it. The `main` function calls `salida2` and `salida1`. The status bar at the bottom indicates 'Ln 14, Col 5', 'Tab Size: 4', 'UTF-8', 'C++', and 'Win32'.

```
100 #include <iostream>
101 #include <conio.h>
102 #include <string.h>
103 #include <ctype.h>
104 #include <stdlib.h>
105 #include <stdio.h>
106 #include <string.h>
107 #include <conio.h>
108 #include <string.h>
109 #include <string.h>
110 #include <string.h>
111 #include <string.h>
112 #include <string.h>
113 #include <string.h>
114 #include <string.h>
115 #include <string.h>
116 #include <string.h>
117 #include <string.h>
118 #include <string.h>
119 #include <string.h>
120 #include <string.h>
121 #include <string.h>
122 #include <string.h>
123 #include <string.h>
124 #include <string.h>
125 #include <string.h>
126 #include <string.h>
127 #include <string.h>
128 #include <string.h>
129 #include <string.h>
130 #include <string.h>
131 #include <string.h>
132 #include <string.h>
133 #include <string.h>
134 #include <string.h>
135 #include <string.h>
136 #include <string.h>
137 #include <string.h>
138 #include <string.h>
139 #include <string.h>
140 #include <string.h>
141 #include <string.h>
```

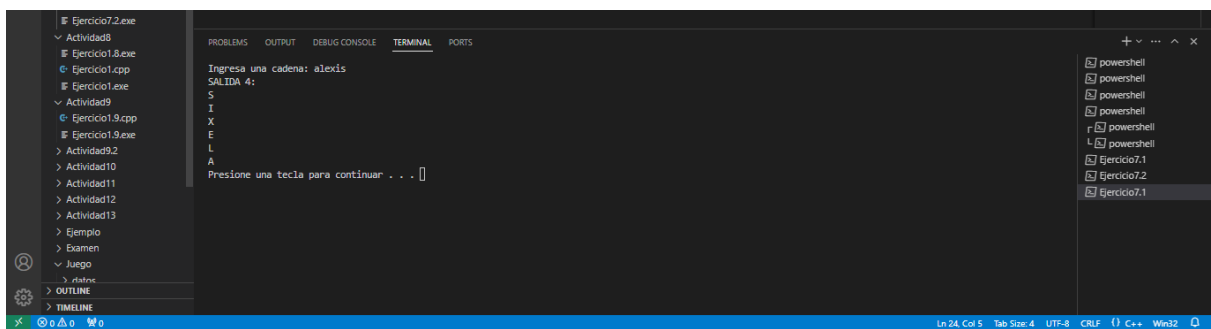
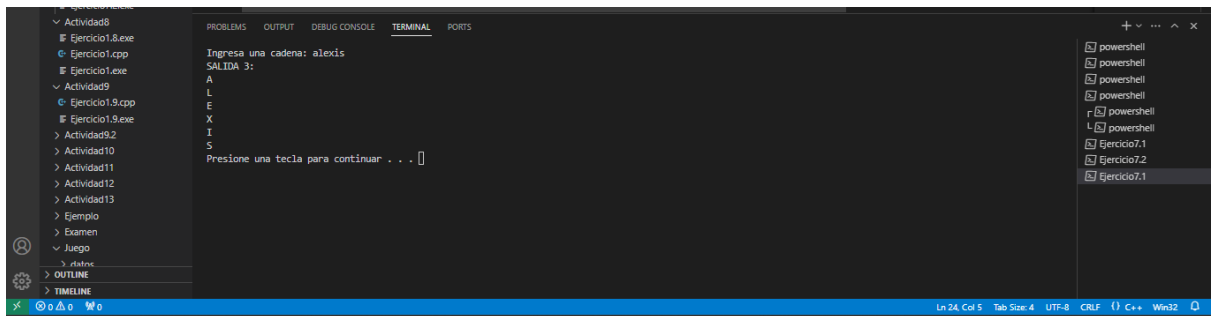
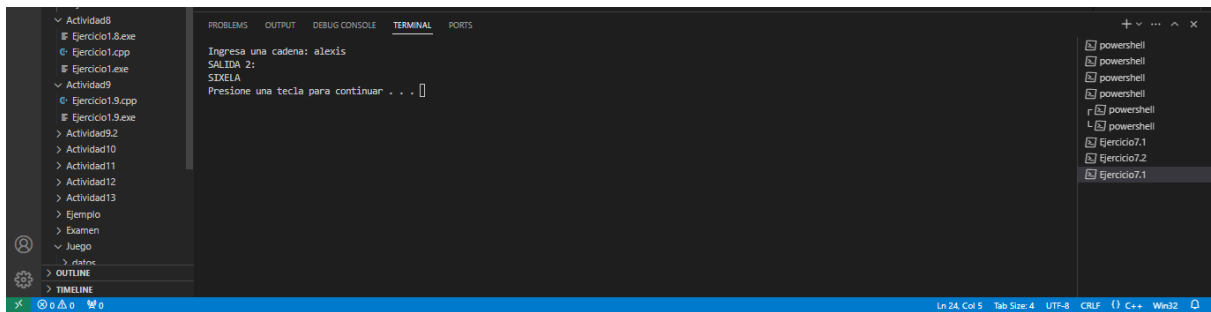
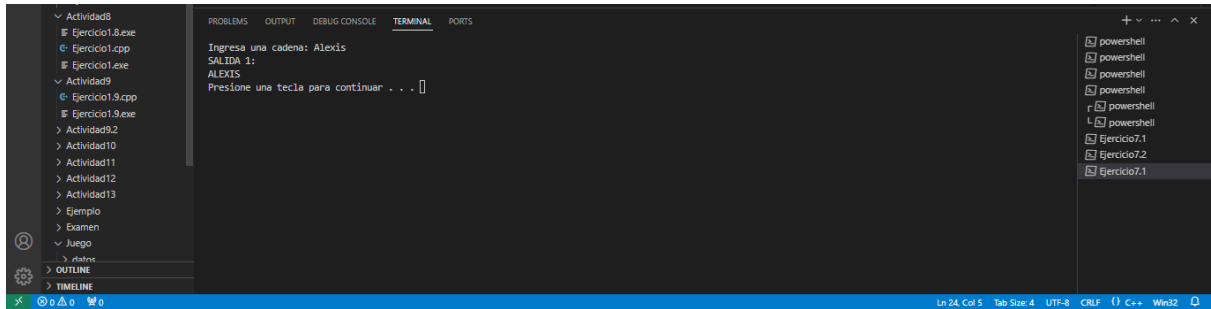
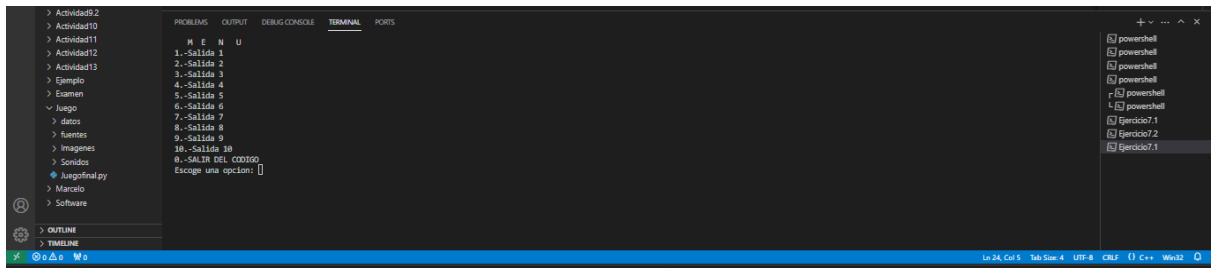
The screenshot shows the Visual Studio Code editor with a C++ project named 'Programacion'. The Explorer sidebar on the left shows a file tree with folders like 'Actividad2' through 'Actividad13', and files like 'Ejercicio6.1.cpp', 'Ejercicio6.1.exe', 'Ejercicio7.1.cpp', 'Ejercicio7.2.cpp', 'Ejercicio7.2.exe', 'Ejercicio8.1.exe', 'Ejercicio9.1.cpp', 'Ejercicio9.1.exe', 'Ejercicio9.2', 'Ejercicio10', 'Ejercicio11', 'Ejercicio12', 'Ejercicio13', 'Ejemplo', 'Examen', 'Juego', 'datos', 'fuentes', 'imagenes', 'Sonidos', 'JuegoFinal.py', 'Marcelo', 'Software'. The main editor window shows the code for 'Ejercicio7.1.cpp' at line 142. The code includes headers for `system`, `printf`, `fflush`, `gets`, `strupr`, `strlen`, `strrev`, and `strcat`. It defines a function `void salida5(void)` that prompts the user to 'Ingresar una cadena', reads the input, converts it to uppercase, reverses it, and prints it. It also defines a function `void salida6(void)` that prompts for a string, calculates its length, and prints it. The `main` function calls `salida5` and `salida6`. The status bar at the bottom indicates 'Ln 14, Col 5', 'Tab Size: 4', 'UTF-8', 'C++', and 'Win32'.

```
142 #include <iostream>
143 #include <conio.h>
144 #include <string.h>
145 #include <ctype.h>
146 #include <stdlib.h>
147 #include <stdio.h>
148 #include <string.h>
149 #include <conio.h>
150 #include <string.h>
151 #include <string.h>
152 #include <string.h>
153 #include <string.h>
154 #include <string.h>
155 #include <string.h>
156 #include <string.h>
157 #include <string.h>
158 #include <string.h>
159 #include <string.h>
160 #include <string.h>
161 #include <string.h>
162 #include <string.h>
163 #include <string.h>
164 #include <string.h>
165 #include <string.h>
166 #include <string.h>
167 #include <string.h>
168 #include <string.h>
169 #include <string.h>
170 #include <string.h>
171 #include <string.h>
172 #include <string.h>
173 #include <string.h>
174 #include <string.h>
175 #include <string.h>
176 #include <string.h>
177 #include <string.h>
178 #include <string.h>
179 #include <string.h>
180 #include <string.h>
181 #include <string.h>
182 #include <string.h>
183 #include <string.h>
184 #include <string.h>
185 #include <string.h>
186 #include <string.h>
187 #include <string.h>
188 #include <string.h>
```

```
189 tam = strlen(cadena);
190 printf("SALIDA 6: \n");
191 for(j=0;j<tam;j++)
192 {
193     for(i=tam-1;i-j>=0;i--)
194     {
195         printf("%c",cadena[i]);
196     }
197     printf("\n");
198 }
199 system("PAUSE");
200
201
202 void salida7(void)
203 {
204     int i,j,tam;
205     char cadena[30];
206     system("CLS");
207     printf("Ingresa una cadena: ");
208     fflush(stdin);
209     gets(cadena);
210     strcpy(cadena);
211     tam = strlen(cadena);
212     printf("SALIDA 7: \n");
213     for(j=0;j<tam;j++)
214     {
215         for(i=0;i<tam-j;i++)
216         {
217             printf("%c",cadena[i+j]);
218         }
219         printf("\n");
220     }
221     system("PAUSE");
222 }
223
224 void salida8(void)
225 {
226     int i,j,tam;
227     char cadena[30];
228     system("CLS");
229     printf("Ingresa una cadena: ");
230     fflush(stdin);
231     gets(cadena);
232     strcpy(cadena);
233     tam = strlen(cadena);
234     printf("SALIDA 8: \n");
235     for(j=0;j<tam;j++)
```

```
236 {
237     for(i=tam-1;i-j>=0;i--)
238     {
239         printf("%c",cadena[i-j]);
240     }
241     printf("\n");
242 }
243 system("PAUSE");
244
245 void salida9(void)
246 {
247     char cadena[30], nuevacadena[30];
248     int i, j = 0;
249     system("CLS");
250     printf("Escribe una cadena: ");
251     fflush(stdin);
252     gets(cadena);
253     printf("SALIDA 9: \n");
254     for(i = 0; i < strlen(cadena); i++)
255     {
256         if(cadena[i] != 'a' && cadena[i] != 'e' && cadena[i] != 'i' && cadena[i] != 'o' && cadena[i] != 'u' && cadena[i] != 'A' && cadena[i] != 'E' && cadena[i] != 'I' && cadena[i] != 'O' &&
257         {
258             nuevacadena[j] = cadena[i];
259             j++;
260         }
261     }
262     nuevacadena[j] = '\0';
263     printf("La cadena sin vocales es: %s\n", nuevacadena);
264     printf("\n");
265     system("PAUSE");
266 }
267
268 void salida10(void)
269 {
270     char cadena[30];
271     int i, j = 0;
272     system("CLS");
273     printf("Escribe una cadena: ");
274     fflush(stdin);
275     gets(cadena);
276     tam = strlen(cadena);
277     printf("SALIDA 10: \n");
278     for (i = 0; i < tam; i++)
```

```
283 {
284     if (cadena[i] == 'a' || cadena[i] == 'e' || cadena[i] == 'i' || cadena[i] == 'o' || cadena[i] == 'u' || cadena[i] == 'A' || cadena[i] == 'E' || cadena[i] == 'I' || cadena[i] == 'O'
285     {
286         printf("%c", cadena[i]);
287     }
288 }
289 printf("\n");
290 system("PAUSE");
291
292
293
294
295
```



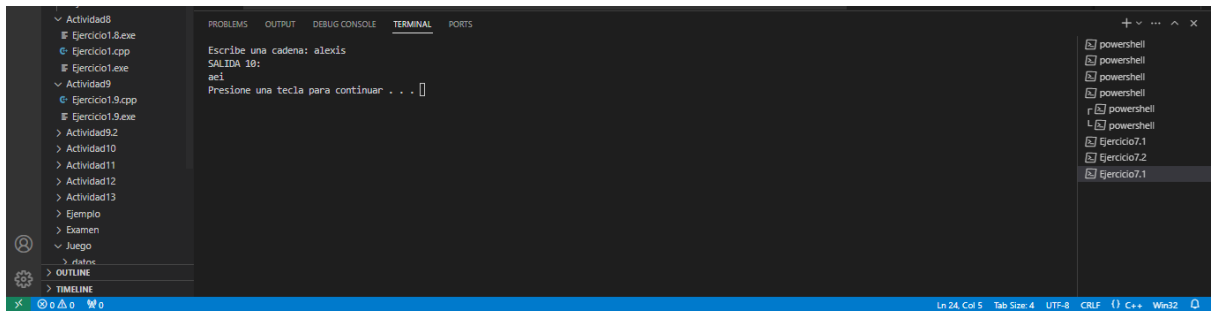
```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Ingresar una cadena: Alexis
SALIDA 5:
ALEXIS
ALEX
ALE
AL
A
Presione una tecla para continuar . . .
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Ingresar una cadena: alexis
SALIDA 6:
SIXELA
SIXEL
SIXE
SIX
SI
S
Presione una tecla para continuar . . .
```

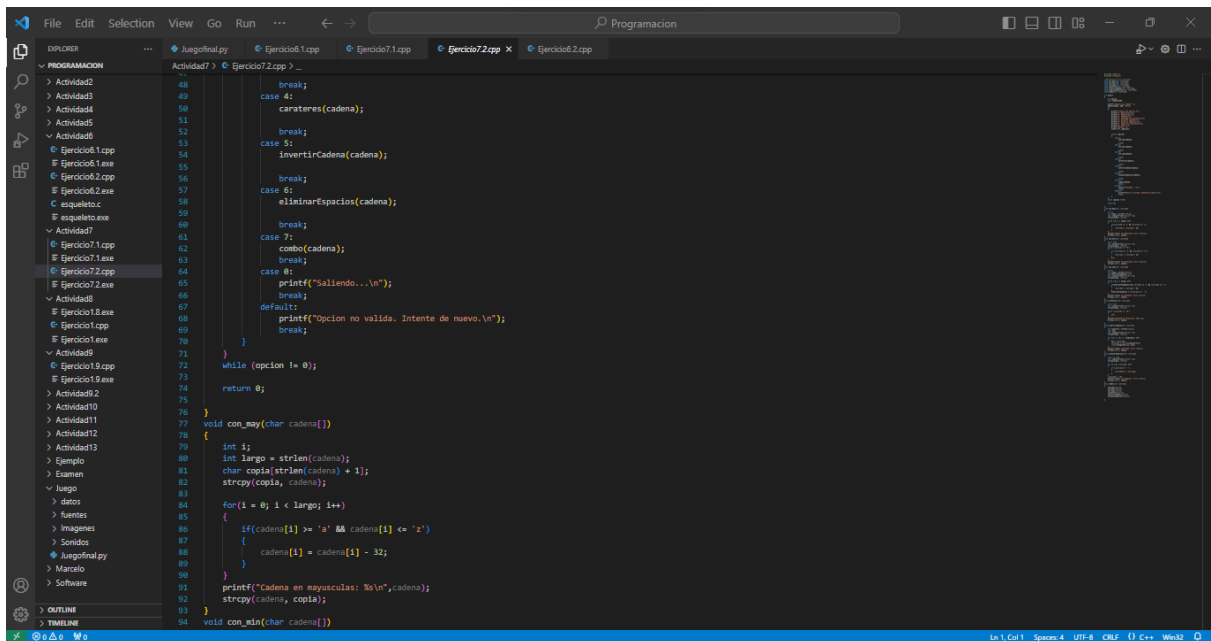
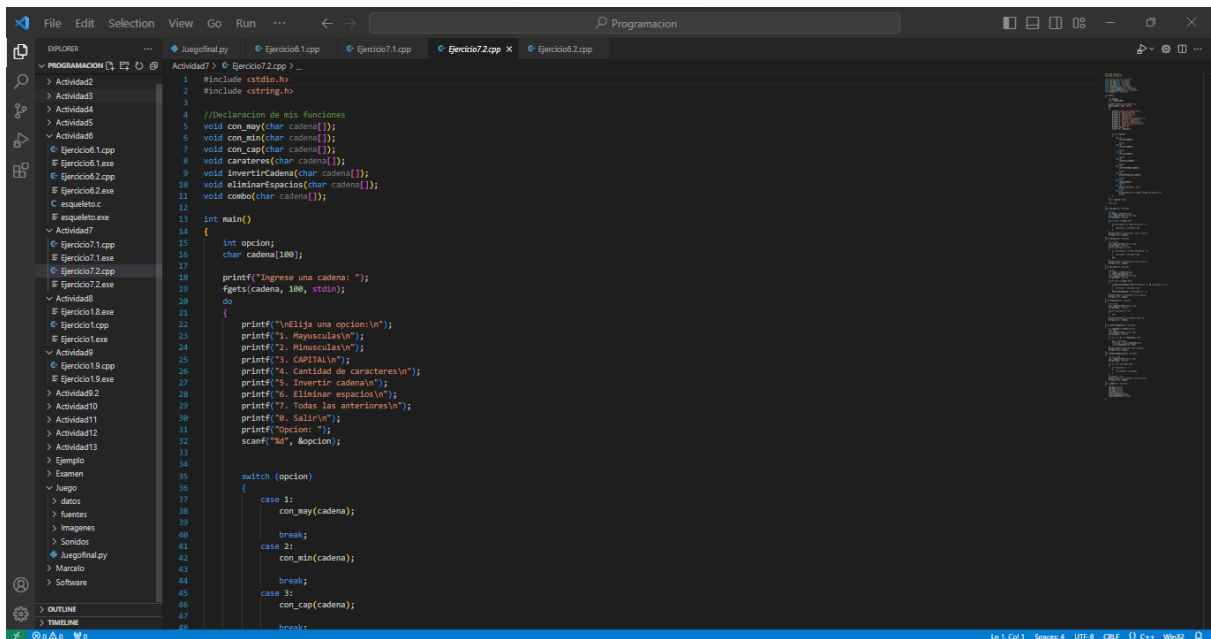
```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Ingresar una cadena: alexis
SALIDA 7:
ALEXIS
LEXIS
EXIS
XIS
IS
S
Presione una tecla para continuar . . .
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Ingresar una cadena: Alexis
SALIDA 8:
SIXELA
IXELA
XELA
ELA
LA
A
Presione una tecla para continuar . . .
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Escribir una cadena: alexis
SALIDA 9:
La cadena sin vocales es: lxs
Presione una tecla para continuar . . .
```



## Parte 2



```
105 int i = 0;
106 char copia[strlen(cadena) + 1];
107 strcpy(copia, cadena);
108 while (cadena[i] != '\0')
109 {
110     if (cadena[i] >= 'A' && cadena[i] <= 'Z')
111     {
112         cadena[i] = cadena[i] + 32;
113     }
114     i++;
115 }
116 printf("Cadena en minusculas: %s\n", cadena);
117 strcpy(cadena, copia);
118 void con_cap(char cadena[])
119 {
120     int i;
121     int largo = strlen(cadena);
122     int anteriorFraseEspacio = 1;
123     char copia[strlen(cadena) + 1];
124     strcpy(copia, cadena);
125     for (i = 0; i < largo; i++)
126     {
127         if (anteriorFraseEspacio && cadena[i] >= 'a' && cadena[i] <= 'z')
128         {
129             cadena[i] = cadena[i] - 32;
130         }
131         anteriorFraseEspacio = (cadena[i] == ' ');
132     }
133     printf("Cadena en capitales: %s\n", cadena);
134     strcpy(cadena, copia);
135 }
136 void carateres(char cadena[])
137 {
138     int i = 0;
139     char copia[strlen(cadena) + 1];
140     strcpy(copia, cadena);
141     while (cadena[i] != '\0')
142     {
143         i++;
144     }
145     printf("Cantidad de caracteres: %d\n", i);
146     strcpy(cadena, copia);
147 }
```

```
142 void invertirCadena(char cadena[])
143 {
144     int longitud = strlen(cadena);
145     char aux;
146     char copia[strlen(cadena) + 1];
147     strcpy(copia, cadena);
148     for (int i = 0; i < longitud/2; i++)
149     {
150         aux = cadena[i];
151         cadena[i] = cadena[longitud-1-i];
152         cadena[longitud-1-i] = aux;
153     }
154     printf("Cadena invertida: %s\n", cadena);
155     strcpy(cadena, copia);
156 }
157 void eliminarEspacios(char cadena[])
158 {
159     int i, j = 0;
160     char copia[strlen(cadena) + 1];
161     strcpy(copia, cadena);
162     for (i = 0; i < strlen(copia); i++)
163     {
164         if (copia[i] != ' ')
165         {
166             cadena[j++] = copia[i];
167         }
168     }
169     cadena[j] = '\0';
170     printf("Cadena sin espacios: %s\n", cadena);
171     strcpy(cadena, copia);
172 }
173 void combo(char cadena[])
174 {
175     con_may(cadena);
176     con_min(cadena);
177     con_cap(cadena);
178     carateres(cadena);
179     invertirCadena(cadena);
180     eliminarEspacios(cadena);
181 }
```

```
158 {
159     int i = 0;
160     char copia[strlen(cadena) + 1];
161     strcpy(copia, cadena);
162     for (int i = 0; i < longitud/2; i++)
163     {
164         aux = cadena[i];
165         cadena[i] = cadena[longitud-1-i];
166         cadena[longitud-1-i] = aux;
167     }
168     printf("Cadena invertida: %s\n", cadena);
169     strcpy(cadena, copia);
170 }
171 void eliminarEspacios(char cadena[])
172 {
173     int i, j = 0;
174     char copia[strlen(cadena) + 1];
175     strcpy(copia, cadena);
176     for (i = 0; i < strlen(copia); i++)
177     {
178         if (copia[i] != ' ')
179         {
180             cadena[j++] = copia[i];
181         }
182     }
183     cadena[j] = '\0';
184     printf("Cadena sin espacios: %s\n", cadena);
185     strcpy(cadena, copia);
186 }
187 void combo(char cadena[])
188 {
189     con_may(cadena);
190     con_min(cadena);
191     con_cap(cadena);
192     carateres(cadena);
193     invertirCadena(cadena);
194     eliminarEspacios(cadena);
195 }
```

```
158 {
159     int i = 0;
160     char copia[strlen(cadena) + 1];
161     strcpy(copia, cadena);
162     for (int i = 0; i < longitud/2; i++)
163     {
164         aux = cadena[i];
165         cadena[i] = cadena[longitud-1-i];
166         cadena[longitud-1-i] = aux;
167     }
168     printf("Cadena invertida: %s\n", cadena);
169     strcpy(cadena, copia);
170 }
171 void eliminarEspacios(char cadena[])
172 {
173     int i, j = 0;
174     char copia[strlen(cadena) + 1];
175     strcpy(copia, cadena);
176     for (i = 0; i < strlen(copia); i++)
177     {
178         if (copia[i] != ' ')
179         {
180             cadena[j++] = copia[i];
181         }
182     }
183     cadena[j] = '\0';
184     printf("Cadena sin espacios: %s\n", cadena);
185     strcpy(cadena, copia);
186 }
187 void combo(char cadena[])
188 {
189     con_may(cadena);
190     con_min(cadena);
191     con_cap(cadena);
192     carateres(cadena);
193     invertirCadena(cadena);
194     eliminarEspacios(cadena);
195 }
```



