

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA



Ingeniería en Software y Tecnologías Emergentes
Programación Estructurada

Actividad 7. Parte 1

ALUMNO: Fernando Haro Calvo

MATRICULA: 372106

GRUPO: 932

PROFESOR: Pedro Núñez Yépiz

24 de septiembre del 2023

Funciones Auxiliares:

```
119  //*****
120  // Convierte la cadena a mayusculas.
121  void mayus(char cadena[])
122  {
123      // Variables Locales
124      int i;
125
126      // Programa
127      for (i = 0; cadena[i] != '\0'; i++)
128      {
129          char caracter = cadena[i];
130
131          if (caracter >= 'a' && caracter <= 'z')
132          {
133              caracter -= 32; // Convertir a mayúscula (ASCII)
134          }
135
136          cadena[i] = caracter;
137      }
138  }
139  //*****
140  // Imprime la cadena normalmente.
141  int largoCadena(char cadena[])
142  {
143      int i;
144
145      for (i = 0; cadena[i] != '\0'; i++)
146      ;
147
148      return i - 1;
149  }
150  //*****
```

Ejercicio 1:

```
150  //*****
151  // Imprime la cadena normalmente.
152  // HCF_ACT7_01_01_932
153  void printNormal(char cadena[])
154  {
155      // Variables Locales
156      int i;
157
158      // Programa
159      for (i = 0; cadena[i] != '\0'; i++)
160      {
161          printf("%c", cadena[i]);
162      }
163  }
```

ENSENADA

Presione una tecla para continuar . . . █

Ejercicio 2:

```
164  //*****
165  // Imprime la cadena al revés.
166  // HCF_ACT7_01_02_932
167  void printReversa(char cadena[])
168  {
169      // Variables Locales
170      int i, largo;
171
172      // Programa
173      largo = largoCadena(cadena);
174
175      for (i = largo; i >= 0; i--)
176      {
177          printf("%c", cadena[i]);
178      }
179  }
```

ADANESNE

Presione una tecla para continuar . . .

Ejercicio 3:

```
180  //*****
181  // Imprime la cadena normal verticalmente.
182  // HCF_ACT7_01_03_932
183  void printVertical(char cadena[])
184  {
185      // Variables Locales
186      int i;
187
188      // Programa
189      for (i = 0; cadena[i] != '\0'; i++)
190      {
191          printf("%c\n", cadena[i]);
192      }
193  }
```

E
N
S
E
N
A
D
A

Presione una tecla para continuar . . . █

Ejercicio 4:

```
194  //*****
195  // Imprime la cadena al revés y verticalmente.
196  // HCF_ACT7_01_04_932
197  void printVerticalReversa(char cadena[])
198  {
199      // Variables Locales
200      int i, largo;
201
202      // Programa
203      largo = largoCadena(cadena);
204
205      for (i = largo; i >= 0; i--)
206      {
207          printf("%c\n", cadena[i]);
208      }
209  }
```

A
D
A
N
E
S
N
E

Presione una tecla para continuar . . .

Ejercicio 5:

```
210 //*****
211 // Imprime la cadena, en cada ciclo borrando un carácter a la derecha.
212 // HCF_ACT7_01_05_932
213 void printBorrarDer(char cadena[])
214 {
215     // Variables Locales
216     int i, j, largo;
217
218     // Programa
219     largo = largoCadena(cadena);
220
221     for (i = largo + 1; i >= 0; i--)
222     {
223         for (j = 0; j < i; j++)
224         {
225             printf("%c", cadena[j]);
226         }
227         printf("\n");
228     }
229 }
```

```
ENSENADA
ENSENAD
ENSENA
ENSEN
ENSE
ENS
EN
E
```

Presione una tecla para continuar . . .

Ejercicio 6:

```
230  //*****
231  // Imprime la cadena al revés, en cada ciclo borrando un carácter a la derecha.
232  // HCF_ACT7_01_06_932
233  void printReversaBorrarDer(char cadena[])
234  {
235      // Variables Locales
236      int i, j, largo;
237
238      // Programa
239      largo = largoCadena(cadena);
240
241      for (i = 0; i <= largo; i++)
242      {
243          for (j = largo; j >= i; j--)
244          {
245              printf("%c", cadena[j]);
246          }
247          printf("\n");
248      }
249  }
```

```
ADANESNE
ADANESN
ADANES
ADANE
ADAN
ADA
AD
A
```

Presione una tecla para continuar . . .

Ejercicio 7:

```
250  //*****
251  // Imprime la cadena, en cada ciclo borrando un carácter a la izquierda.
252  // HCF_ACT7_01_07_932
253  void printBorrarIzq(char cadena[])
254  {
255      // Variables Locales
256      int i, j, contador, largo;
257
258      // Programa
259      largo = largoCadena(cadena);
260
261      for (i = largo; i >= 0; i--)
262      {
263          for (j = largo - i; j <= largo; j++)
264          {
265              printf("%c", cadena[j]);
266          }
267          printf("\n");
268      }
269  }
```

```
ENSENADA
NSENADA
SENADA
ENADA
NADA
ADA
DA
A
```

Presione una tecla para continuar . . .

Ejercicio 8:

```
270 //*****
271 // Imprime la cadena al revés, en cada ciclo borrando un carácter a la izquierda.
272 // HCF_ACT7_01_08_932
273 void printReversaBorrarIzq(char cadena[])
274 {
275     // Variables Locales
276     int i, j, largo;
277
278     // Programa
279     largo = largoCadena(cadena);
280
281     for (i = largo; i >= 0; i--)
282     {
283         for (j = i; j >= 0; j--)
284         {
285             printf("%c", cadena[j]);
286         }
287         printf("\n");
288     }
289 }
```

```
ADANESNE
DANESNE
ANESNE
NESNE
ESNE
SNE
NE
E
```

Presione una tecla para continuar . . .

Ejercicio 9:

```
290  //*****
291  // Imprime solo las letras consonantes de la cadena.
292  // HCF_ACT7_01_09_932
293  void printConsonantes(char cadena[])
294  {
295      // Variables Locales
296      int i;
297
298      // Programa
299      for (i = 0; cadena[i] != '\0'; i++)
300      {
301          if (cadena[i] != 'A' && cadena[i] != 'E' && cadena[i] != 'I' && cadena[i] != 'O' && cadena[i] != 'U')
302          {
303              printf("%c", cadena[i]);
304          }
305      }
306  }
```

NSND

Presione una tecla para continuar . . .

Ejercicio 10:

```
307  //*****
308  // Imprime solo las letras vocales de la cadena.
309  // HCF_ACT7_01_10_932
310  void printVocales(char cadena[])
311  {
312      // Variables Locales
313      int i;
314
315      // Programa
316      for (i = 0; cadena[i] != '\0'; i++)
317      {
318          if (cadena[i] == 'A' || cadena[i] == 'E' || cadena[i] == 'I' || cadena[i] == 'O' || cadena[i] == 'U')
319          {
320              printf("%c", cadena[i]);
321          }
322      }
323  }
```

EEAA

Presione una tecla para continuar . . .