



E. A. Nro. 1 Rehacer– Primera Evaluación de Aprendizaje - (Rehacer).

Debe resolver la siguiente problemática eligiendo la mejor estrategia.

Punto 1.- Para modificar el desempeño de un procesador de texto le han hecho un requerimiento, dado que *buscar y reemplazar* todas las ocurrencias de una cadena en un archivo que se está editando resulta inaceptablemente lento.

Se requiere una función que, en una cadena de caracteres (primer argumento), reemplace todas las ocurrencias de una sub cadena (tercer argumento), por otra sub cadena (cuarto argumento). Las sub cadenas del tercer y cuarto argumento pueden tener distinto tamaño (cantidad de caracteres). Como no se debe exceder el espacio de almacenamiento de la cadena, en su segundo argumento se pone la cantidad de posiciones (tamaño) del array. La función debe devolver la cantidad de reemplazos. Si no se encuentra la cadena a reemplazar debe devolver 0 (cero). Si hubiera que hacer un reemplazo y no se puede hacer por falta de espacio, deberá devolver -1 (menos uno) en lugar de devolver cero. Si se pudo hacer un reemplazo, pero por falta de espacio no se pueden hacer más, debe devolver -2 (menos dos). Si se pudieron hacer dos reemplazos y por falta de espacio no se pueden hacer más, debe devolver -3 (menos tres) y así sucesivamente.

- No debe hacer uso de funciones de biblioteca.
- No puede generar arrays auxiliares.
- La cadena que se, recibe se modifica in situ (sobre sí misma).
- Debe realizar todas las pruebas que considere caso de contorno (o borde -las que pueden hacer fallar su función-), y considerarlas.



UNLaM

Dto. Ingeniería e Investigaciones Tecnológicas

Punto 2.- Se requiere que haga una función que cargue un array bidimensional de enteros (matriz cuadrada con la misma cantidad de filas y columnas -esa cantidad debe ser impar-). Debe generar el llamado "cuadrado mágico" en que la suma de los elementos de cada fila y la de los de sus columnas además de la suma de la diagonal principal y de la secundaria dan lo mismo. Al efecto le mostramos algunos ejemplos:

<table><tr><td>8</td><td>1</td><td>6</td><td>15</td></tr><tr><td>3</td><td>5</td><td>7</td><td>15</td></tr><tr><td>4</td><td>9</td><td>2</td><td>15</td></tr><tr><td>15</td><td>15</td><td>15</td><td>15</td></tr></table>					8	1	6	15	3	5	7	15	4	9	2	15	15	15	15	15	<table><tr><td>47</td><td>58</td><td>69</td><td>80</td><td>1</td><td>12</td><td>23</td><td>34</td><td>45</td><td>369</td></tr><tr><td>57</td><td>68</td><td>79</td><td>9</td><td>11</td><td>22</td><td>33</td><td>44</td><td>46</td><td>369</td></tr><tr><td>67</td><td>78</td><td>8</td><td>10</td><td>21</td><td>32</td><td>43</td><td>54</td><td>56</td><td>369</td></tr><tr><td>77</td><td>7</td><td>18</td><td>20</td><td>31</td><td>42</td><td>53</td><td>55</td><td>66</td><td>369</td></tr><tr><td>6</td><td>17</td><td>19</td><td>30</td><td>41</td><td>52</td><td>63</td><td>65</td><td>76</td><td>369</td></tr><tr><td>16</td><td>27</td><td>29</td><td>40</td><td>51</td><td>62</td><td>64</td><td>75</td><td>5</td><td>369</td></tr><tr><td>26</td><td>28</td><td>39</td><td>50</td><td>61</td><td>72</td><td>74</td><td>4</td><td>15</td><td>369</td></tr><tr><td>36</td><td>38</td><td>49</td><td>60</td><td>71</td><td>73</td><td>3</td><td>14</td><td>25</td><td>369</td></tr><tr><td>37</td><td>48</td><td>59</td><td>70</td><td>81</td><td>2</td><td>13</td><td>24</td><td>35</td><td>369</td></tr><tr><td>369</td><td>369</td><td>369</td><td>369</td><td>369</td><td>369</td><td>369</td><td>369</td><td>369</td><td>369</td></tr></table>																	47	58	69	80	1	12	23	34	45	369	57	68	79	9	11	22	33	44	46	369	67	78	8	10	21	32	43	54	56	369	77	7	18	20	31	42	53	55	66	369	6	17	19	30	41	52	63	65	76	369	16	27	29	40	51	62	64	75	5	369	26	28	39	50	61	72	74	4	15	369	36	38	49	60	71	73	3	14	25	369	37	48	59	70	81	2	13	24	35	369	369	369	369	369	369	369	369	369	369	369																																																																																																																																																																																																													
8	1	6	15																																																																																																																																																																																																																																																																																																																																																			
3	5	7	15																																																																																																																																																																																																																																																																																																																																																			
4	9	2	15																																																																																																																																																																																																																																																																																																																																																			
15	15	15	15																																																																																																																																																																																																																																																																																																																																																			
47	58	69	80	1	12	23	34	45	369																																																																																																																																																																																																																																																																																																																																													
57	68	79	9	11	22	33	44	46	369																																																																																																																																																																																																																																																																																																																																													
67	78	8	10	21	32	43	54	56	369																																																																																																																																																																																																																																																																																																																																													
77	7	18	20	31	42	53	55	66	369																																																																																																																																																																																																																																																																																																																																													
6	17	19	30	41	52	63	65	76	369																																																																																																																																																																																																																																																																																																																																													
16	27	29	40	51	62	64	75	5	369																																																																																																																																																																																																																																																																																																																																													
26	28	39	50	61	72	74	4	15	369																																																																																																																																																																																																																																																																																																																																													
36	38	49	60	71	73	3	14	25	369																																																																																																																																																																																																																																																																																																																																													
37	48	59	70	81	2	13	24	35	369																																																																																																																																																																																																																																																																																																																																													
369	369	369	369	369	369	369	369	369	369																																																																																																																																																																																																																																																																																																																																													
matriz de 3 x 3					Matriz de 9 x 9																																																																																																																																																																																																																																																																																																																																																	
<table><tr><td>155</td><td>174</td><td>193</td><td>212</td><td>231</td><td>250</td><td>269</td><td>288</td><td>1</td><td>20</td><td>39</td><td>58</td><td>77</td><td>96</td><td>115</td><td>134</td><td>153</td><td>2465</td></tr><tr><td>173</td><td>192</td><td>211</td><td>230</td><td>249</td><td>268</td><td>287</td><td>17</td><td>19</td><td>38</td><td>57</td><td>76</td><td>95</td><td>114</td><td>133</td><td>152</td><td>154</td><td>2465</td></tr><tr><td>191</td><td>210</td><td>229</td><td>248</td><td>267</td><td>286</td><td>16</td><td>18</td><td>37</td><td>56</td><td>75</td><td>94</td><td>113</td><td>132</td><td>151</td><td>170</td><td>172</td><td>2465</td></tr><tr><td>209</td><td>228</td><td>247</td><td>266</td><td>285</td><td>15</td><td>34</td><td>36</td><td>55</td><td>74</td><td>93</td><td>112</td><td>131</td><td>150</td><td>169</td><td>171</td><td>190</td><td>2465</td></tr><tr><td>227</td><td>246</td><td>265</td><td>284</td><td>14</td><td>33</td><td>35</td><td>54</td><td>73</td><td>92</td><td>111</td><td>130</td><td>149</td><td>168</td><td>187</td><td>189</td><td>208</td><td>2465</td></tr><tr><td>245</td><td>264</td><td>283</td><td>13</td><td>32</td><td>51</td><td>53</td><td>72</td><td>91</td><td>110</td><td>129</td><td>148</td><td>167</td><td>186</td><td>188</td><td>207</td><td>226</td><td>2465</td></tr><tr><td>263</td><td>282</td><td>12</td><td>31</td><td>50</td><td>52</td><td>71</td><td>90</td><td>109</td><td>128</td><td>147</td><td>166</td><td>185</td><td>204</td><td>206</td><td>225</td><td>244</td><td>2465</td></tr><tr><td>281</td><td>11</td><td>30</td><td>49</td><td>68</td><td>70</td><td>89</td><td>108</td><td>127</td><td>146</td><td>165</td><td>184</td><td>203</td><td>205</td><td>224</td><td>243</td><td>262</td><td>2465</td></tr><tr><td>10</td><td>29</td><td>48</td><td>67</td><td>69</td><td>88</td><td>107</td><td>126</td><td>145</td><td>164</td><td>183</td><td>202</td><td>221</td><td>223</td><td>242</td><td>261</td><td>280</td><td>2465</td></tr><tr><td>28</td><td>47</td><td>66</td><td>85</td><td>87</td><td>106</td><td>125</td><td>144</td><td>163</td><td>182</td><td>201</td><td>220</td><td>222</td><td>241</td><td>260</td><td>279</td><td>9</td><td>2465</td></tr><tr><td>46</td><td>65</td><td>84</td><td>86</td><td>105</td><td>124</td><td>143</td><td>162</td><td>181</td><td>200</td><td>219</td><td>238</td><td>240</td><td>259</td><td>278</td><td>8</td><td>27</td><td>2465</td></tr><tr><td>64</td><td>83</td><td>102</td><td>104</td><td>123</td><td>142</td><td>161</td><td>180</td><td>199</td><td>218</td><td>237</td><td>239</td><td>258</td><td>277</td><td>7</td><td>26</td><td>45</td><td>2465</td></tr><tr><td>82</td><td>101</td><td>103</td><td>122</td><td>141</td><td>160</td><td>179</td><td>198</td><td>217</td><td>236</td><td>255</td><td>257</td><td>276</td><td>6</td><td>25</td><td>44</td><td>63</td><td>2465</td></tr><tr><td>100</td><td>119</td><td>121</td><td>140</td><td>159</td><td>178</td><td>197</td><td>216</td><td>235</td><td>254</td><td>256</td><td>275</td><td>5</td><td>24</td><td>43</td><td>62</td><td>81</td><td>2465</td></tr><tr><td>118</td><td>120</td><td>139</td><td>158</td><td>177</td><td>196</td><td>215</td><td>234</td><td>253</td><td>272</td><td>274</td><td>4</td><td>23</td><td>42</td><td>61</td><td>80</td><td>99</td><td>2465</td></tr><tr><td>136</td><td>138</td><td>157</td><td>176</td><td>195</td><td>214</td><td>233</td><td>252</td><td>271</td><td>273</td><td>3</td><td>22</td><td>41</td><td>60</td><td>79</td><td>98</td><td>117</td><td>2465</td></tr><tr><td>137</td><td>156</td><td>175</td><td>194</td><td>213</td><td>232</td><td>251</td><td>270</td><td>289</td><td>2</td><td>21</td><td>40</td><td>59</td><td>78</td><td>97</td><td>116</td><td>135</td><td>2465</td></tr><tr><td>2465</td><td>2465</td><td>2465</td><td>2465</td><td>2465</td><td>2465</td><td>2465</td><td>2465</td><td>2465</td><td>2465</td><td>2465</td><td>2465</td><td>2465</td><td>2465</td><td>2465</td><td>2465</td><td>2465</td><td>2465</td></tr></table>																			155	174	193	212	231	250	269	288	1	20	39	58	77	96	115	134	153	2465	173	192	211	230	249	268	287	17	19	38	57	76	95	114	133	152	154	2465	191	210	229	248	267	286	16	18	37	56	75	94	113	132	151	170	172	2465	209	228	247	266	285	15	34	36	55	74	93	112	131	150	169	171	190	2465	227	246	265	284	14	33	35	54	73	92	111	130	149	168	187	189	208	2465	245	264	283	13	32	51	53	72	91	110	129	148	167	186	188	207	226	2465	263	282	12	31	50	52	71	90	109	128	147	166	185	204	206	225	244	2465	281	11	30	49	68	70	89	108	127	146	165	184	203	205	224	243	262	2465	10	29	48	67	69	88	107	126	145	164	183	202	221	223	242	261	280	2465	28	47	66	85	87	106	125	144	163	182	201	220	222	241	260	279	9	2465	46	65	84	86	105	124	143	162	181	200	219	238	240	259	278	8	27	2465	64	83	102	104	123	142	161	180	199	218	237	239	258	277	7	26	45	2465	82	101	103	122	141	160	179	198	217	236	255	257	276	6	25	44	63	2465	100	119	121	140	159	178	197	216	235	254	256	275	5	24	43	62	81	2465	118	120	139	158	177	196	215	234	253	272	274	4	23	42	61	80	99	2465	136	138	157	176	195	214	233	252	271	273	3	22	41	60	79	98	117	2465	137	156	175	194	213	232	251	270	289	2	21	40	59	78	97	116	135	2465	2465	2465	2465	2465	2465	2465	2465	2465	2465	2465	2465	2465	2465	2465	2465	2465	2465	2465
155	174	193	212	231	250	269	288	1	20	39	58	77	96	115	134	153	2465																																																																																																																																																																																																																																																																																																																																					
173	192	211	230	249	268	287	17	19	38	57	76	95	114	133	152	154	2465																																																																																																																																																																																																																																																																																																																																					
191	210	229	248	267	286	16	18	37	56	75	94	113	132	151	170	172	2465																																																																																																																																																																																																																																																																																																																																					
209	228	247	266	285	15	34	36	55	74	93	112	131	150	169	171	190	2465																																																																																																																																																																																																																																																																																																																																					
227	246	265	284	14	33	35	54	73	92	111	130	149	168	187	189	208	2465																																																																																																																																																																																																																																																																																																																																					
245	264	283	13	32	51	53	72	91	110	129	148	167	186	188	207	226	2465																																																																																																																																																																																																																																																																																																																																					
263	282	12	31	50	52	71	90	109	128	147	166	185	204	206	225	244	2465																																																																																																																																																																																																																																																																																																																																					
281	11	30	49	68	70	89	108	127	146	165	184	203	205	224	243	262	2465																																																																																																																																																																																																																																																																																																																																					
10	29	48	67	69	88	107	126	145	164	183	202	221	223	242	261	280	2465																																																																																																																																																																																																																																																																																																																																					
28	47	66	85	87	106	125	144	163	182	201	220	222	241	260	279	9	2465																																																																																																																																																																																																																																																																																																																																					
46	65	84	86	105	124	143	162	181	200	219	238	240	259	278	8	27	2465																																																																																																																																																																																																																																																																																																																																					
64	83	102	104	123	142	161	180	199	218	237	239	258	277	7	26	45	2465																																																																																																																																																																																																																																																																																																																																					
82	101	103	122	141	160	179	198	217	236	255	257	276	6	25	44	63	2465																																																																																																																																																																																																																																																																																																																																					
100	119	121	140	159	178	197	216	235	254	256	275	5	24	43	62	81	2465																																																																																																																																																																																																																																																																																																																																					
118	120	139	158	177	196	215	234	253	272	274	4	23	42	61	80	99	2465																																																																																																																																																																																																																																																																																																																																					
136	138	157	176	195	214	233	252	271	273	3	22	41	60	79	98	117	2465																																																																																																																																																																																																																																																																																																																																					
137	156	175	194	213	232	251	270	289	2	21	40	59	78	97	116	135	2465																																																																																																																																																																																																																																																																																																																																					
2465	2465	2465	2465	2465	2465	2465	2465	2465	2465	2465	2465	2465	2465	2465	2465	2465	2465																																																																																																																																																																																																																																																																																																																																					
Matriz de 17 x 17																																																																																																																																																																																																																																																																																																																																																						
En las imágenes anteriores, dentro de las celdas, se muestran los números que forman el cuadrado mágico. A la derecha de cada fila, dejando un espacio extra, la suma de los elementos de cada fila. Debajo de cada columna, dejando un renglón vacío (en blanco), la suma de los elementos de cada columna. Y abajo a la derecha, dejando un espacio en blanco extra, la suma de la diagonal principal y de la diagonal secundaria.																																																																																																																																																																																																																																																																																																																																																						

En las imágenes anteriores, dentro de las celdas, se muestran los números que forman el cuadrado mágico. A la derecha de cada fila, dejando un espacio extra, la suma de los elementos de cada fila. Debajo de cada columna, dejando un renglón vacío (en blanco), la suma de los elementos de cada columna. Y abajo a la derecha, dejando un espacio en blanco extra, la suma de la diagonal principal y de la diagonal secundaria.

Se define al "*cuadrado mágico*" como una "*matriz esférica*", es decir que la fila anterior de la primera es la última, la fila siguiente de la última es la primera, la columna anterior de la primera es la última, y la columna siguiente de la última es la primera. Para *cargarla*, se procede a hacerlo con los números enteros sucesivos a partir de 1 (uno) como se indica más abajo.



- Debe escribir una función que cargue la matriz.
- Debe escribir otra función que la muestre.

Para generarla, entre todos los modos posibles, debe comenzar con 1 en la mitad de la primera fila. Los enteros sucesivos se colocan en la fila anterior columna siguiente salvo que esté ocupada, en cuyo caso se coloca en la misma columna de la fila siguiente.

Para mostrarla y poder comprobar que está bien cargada, debe mostrar los elementos de cada fila y dejando un espacio en blanco extra la suma de estos. Una vez mostradas todas las filas, debe mostrar (dejando un renglón en blanco) la suma de las columnas. Una vez mostradas debe mostrar la suma de la diagonal principal y la de la diagonal secundaria. El proyecto que se le entrega puede mostrar, sin mayores problemas por la cantidad de dígitos de los elementos y de las sumas, matrices de hasta 27 x 27.

RECOMENDACIÓN:

Analice el problema, entiéndalo, y busque una solución para que el algoritmo sea óptimo.

- es decir NO DEBE inicializar la matriz con un valor particular (por ejemplo, cero) que indique que esa posición no está ocupada.
- NO DEBE modificar una variable para luego, dada una condición modificarla nuevamente.
- NO DEBE emplear arrays auxiliares.

Aproveche el proyecto y genere matrices de orden 3, 5, 7, 9 y 11 y analice cómo resolver lo antes pedido. Recuerde que se pide una solución óptima.



Punto 3.- Los pedidos de clientes se almacenan a lo largo del día en un archivo binario (**pedido**) ⁽¹⁾. Si de alguno de los ítems del pedido no tuviera el precio unitario pactado para el producto, el pedido completo se almacenará en el archivo de pedidos pendientes (**pendie**) ^{(1) y (2)} con la misma estructura de información y orden. Se requiere generar dos listados, para el encargado del depósito, con los pedidos que no queden como pendientes. En uno (**mayor.txt**) ⁽³⁾ irán los pedidos por montos mayores a **un cierto importe**, y en el otro (**menor.txt**) ⁽³⁾ irán los demás. En los archivos para el encargado de depósito se separa cada pedido del siguiente con tres líneas de texto en blanco (vacías). Haga uso de un TDA PILA para almacenar temporariamente los ítems de cada pedido.

Aclaraciones:

⁽¹⁾ El archivo de pedidos de los clientes (**pedido**) y el de pedidos pendientes (**pendie**) tienen una estructura de registro (ver el proyecto) compuesta de código de cliente, código de producto, cantidad pedida y precio unitario pactado que queda en el archivo al momento de cargar el pedido. El archivo de pedidos se debe leer secuencialmente (cada registro se lee por única vez). El pedido de un cliente puede ser de uno o varios productos, con lo que habrá tantos registros consecutivos como productos pida el cliente (uno por cada producto).

⁽²⁾ Los registros del archivo de salida pedidos pendientes (**pendie**) deben estar en el mismo orden en que estaban en el archivo de pedidos.

⁽³⁾ En los archivos para el encargado de depósito (**mayor.txt** y **menor.txt**) se debe almacenar el código de cliente en una sola línea de texto junto con el importe, (ver ejemplo):



UNLaM

Dto. Ingeniería e Investigaciones Tecnológicas

```
*menor.txt - Notepad
File Edit Format View Help
Pedido del cliente xxxxxxxxx por dddd.dd

CodProduc. Cant.  PrecUnit  SubTotal
Producto-1   ddd   dddd.dd dddddd.dd
KXP11-1/a    dd    dd.dd   ddd.dd

Pedido del cliente yyyyyyyyyy por dddddd.dd

Ln 13, Col 1  100%  Windows (CRLF)  UTF-8
```

Note que

- las claves de productos y de clientes pueden tener distinta longitud (hasta diez caracteres).
 - para los importes y los subtotales se supone que no alcanzan a \$100.000.000.
- (3) En los archivos para el encargado de depósito (**mayor.txt** y **menor.txt**) se deben grabar los ítem del pedido en el mismo orden en que estaban en el archivo de pedidos (**pedido**).

Consideraciones Generales

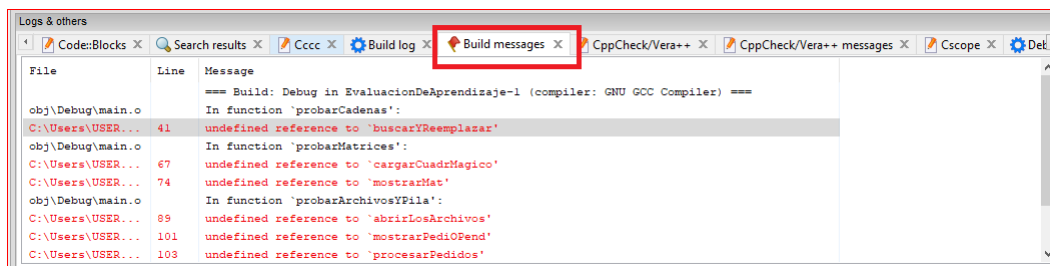
Descargue el proyecto provisto, genere una *carpeta* en su computadora y en ella descomprímalo.

Verá que se generan dos *subcarpetas*, una para plataforma (o compilador) de 32 bits y otro para plataforma de 64 bits.

En la carpeta que corresponde a su IDE (**EAR1-Proyecto64** o **EAR1-Proyecto32**) abra el proyecto que corresponde a su plataforma (habitualmente doble click en el archivo **EA-1.cbp** o en su defecto una vez abierto el IDE, "arrastrándolo" en el mismo).

Proceda a compilarlo.

Si en el cuadro de diálogo inferior le aparecen errores:



. . . ¡¡¡es porque se equivocó de proyecto!!! Vaya al otro, compile y ejecute:



UNLaM

Dto. Ingeniería e Investigaciones Tecnológicas

```

1
2 #include "main.h"
3
4 int main()
5 {
6     FILE *fpPant = fopen("pantalla.txt", "wt");
7
8     if(fpPant == NULL) // si no puede abrir el archivo
9         fpPant = stdout; // mostrará por consola
10
11     probarCadenas(fpPant);
12
13     probarMatrices(fpPant);
14
15 }

```

Build: Debug in EvaluacionDeAprendizaje-1 (compiler: GNU GCC Compiler)

```

gcc.exe -Wall -g -c C:\Users\USER\Desktop\EAR-1\EAR1-Proyecto64\funciones1.c -o obj\Debug\funciones1.o
gcc.exe -Wall -g -c C:\Users\USER\Desktop\EAR-1\EAR1-Proyecto64\funciones2.c -o obj\Debug\funciones2.o
gcc.exe -Wall -g -c C:\Users\USER\Desktop\EAR-1\EAR1-Proyecto64\funciones3.c -o obj\Debug\funciones3.o
gcc.exe -Wall -g -c C:\Users\USER\Desktop\EAR-1\EAR1-Proyecto64\main.c -o obj\Debug\main.o
gcc.exe -o bin\Debug\EvaluacionDeAprendizaje-1.exe obj\Debug\funciones1.o obj\Debug\funciones2.o obj\Debug\funciones3.o obj\Debug\main.o
Output file is bin\Debug\EvaluacionDeAprendizaje-1.exe with size 69.79 KB
Process terminated with status 0 (0 minute(s), 0 second(s))
0 error(s), 0 warning(s) (0 minute(s), 0 second(s))

```

El proyecto deberá compilar sin errores con las opciones de compilación marcadas.

Ejecútelo

```

// Ini - Parte 1 //
En la cadena
"int funcion(int seIntegra, int noIntegra);":
Se reemplaza la sub cadena "int" por la subcadena "unsigned int".
Se hicieron 5 reemplazos.
La cadena queda modificada:
"unsigned int funcion(unsigned int seunsigned integra, unsigned int nounsigned integra);".
// FIN - Parte 1 //

// Ini - Parte 2 //
El cuadrado magico es:
155 174 193 212 231 250 269 288 1 20 39 58 77 96 115 134 153 2465
173 192 211 230 249 268 287 17 19 38 57 76 95 114 133 152 154 2465
191 210 229 248 267 286 16 18 37 56 75 94 113 132 151 170 172 2465
209 228 247 266 285 15 34 36 55 74 93 112 131 150 169 171 190 2465
227 246 265 284 14 33 35 54 73 92 111 130 149 168 187 189 208 2465
245 264 283 13 32 51 53 72 91 110 129 148 167 186 188 207 226 2465
263 282 12 31 50 52 71 90 109 128 147 166 185 204 206 225 244 2465
281 11 30 49 68 70 89 108 127 146 165 184 203 205 224 243 262 2465
10 29 48 67 69 88 107 126 145 164 183 202 221 223 242 261 280 2465
28 47 66 85 87 106 125 144 163 182 201 220 222 241 260 279 9 2465
46 65 84 86 105 124 143 162 181 200 219 238 240 259 278 8 27 2465
64 83 102 104 123 142 161 180 199 218 237 239 258 277 7 26 45 2465
82 101 103 122 141 160 179 198 217 236 255 257 276 6 25 44 63 2465
100 119 121 140 159 178 197 216 235 254 256 275 5 24 43 62 81 2465
118 120 139 158 177 196 215 234 253 272 274 4 23 42 61 80 99 2465

```

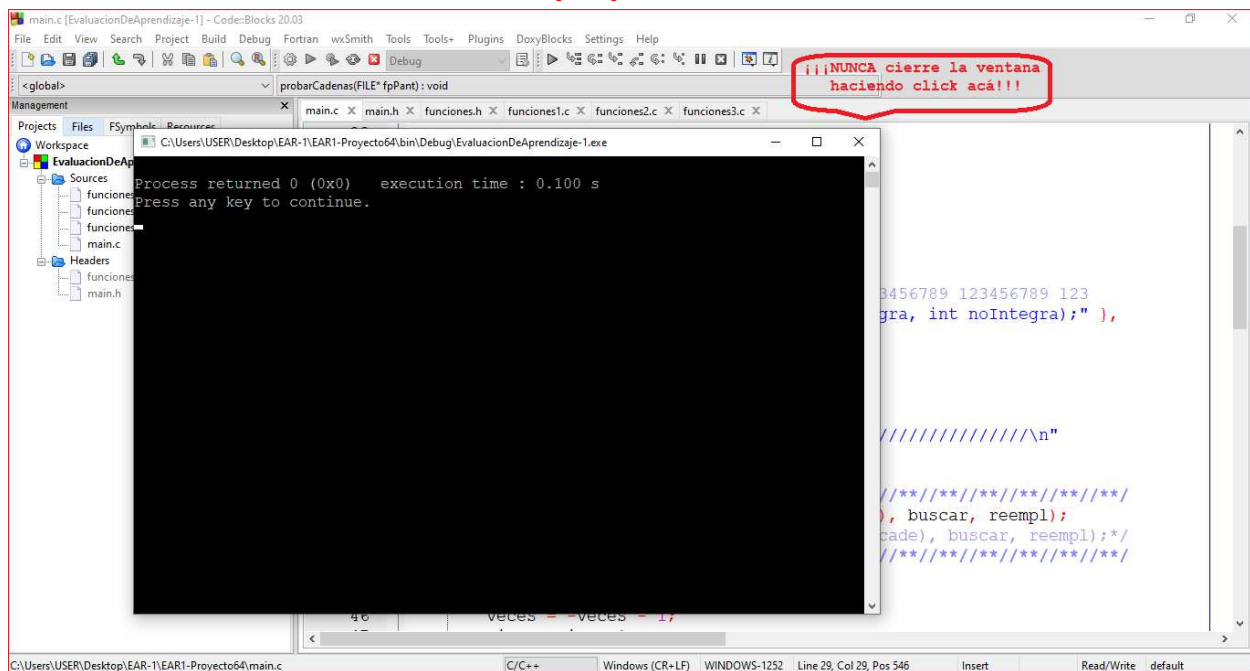



Verá que la ventana de ejecución (consola dos) queda tapada por un archivo de texto (`pantalla.txt` abierto por el "*Block de Notas*" (o "*Notepad*"). Acomode el ancho del mismo para poder verlo como en la figura anterior.

Una vez que haya visto la salida del programa (archivo `pantalla.txt`) puede proceder a cerrarlo, minimizarlo o guardarlo (por ejemplo, con el nombre `pantallaESPERABLE.txt`). Esta es la salida por pantalla que debe lograr.

Cierre o minimice el "*Block de Notas*" (o "*Notepad*").

!!!RECUERDE que la ventana de ejecución (consola DOS) NO LA DEBE CERRAR de otro modo que pulsando una tecla!!!

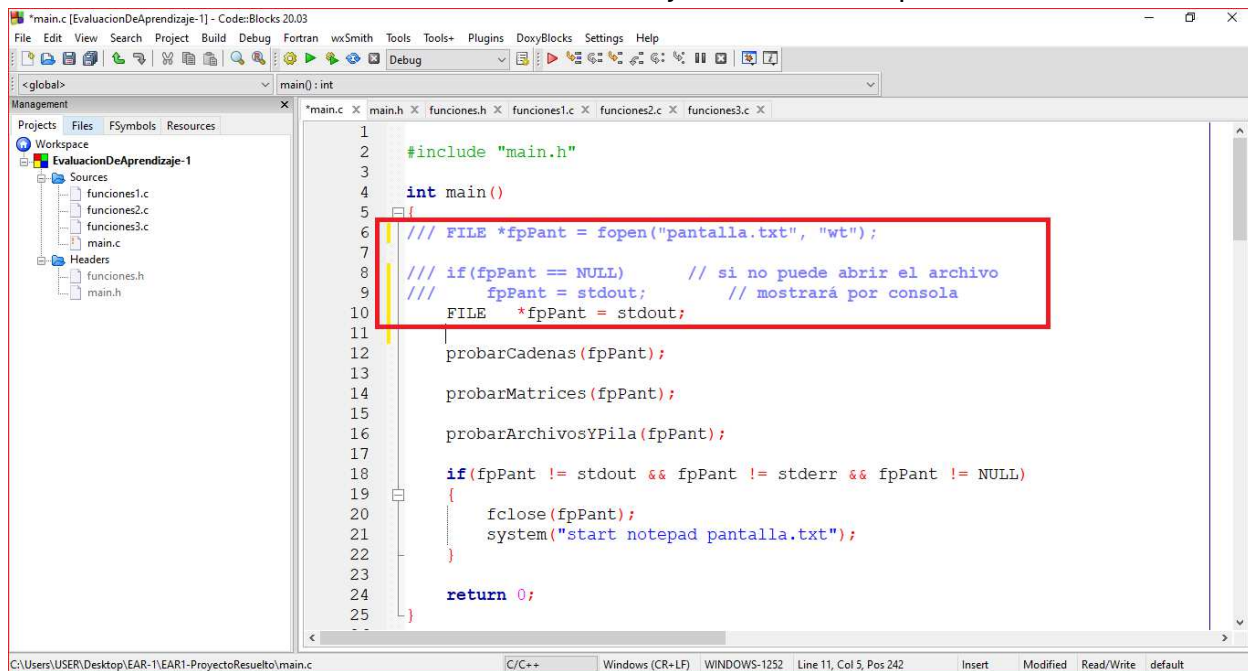




UNLaM

Dto. Ingeniería e Investigaciones Tecnológicas

Si le molesta que se abra el archivo "**pantalla.txt**" cada vez que ejecuta el proyecto durante el desarrollo de las soluciones a los ejercicios, modifique la función "**main**" . . .



```
1
2  #include "main.h"
3
4  int main()
5  {
6      /// FILE *fpPant = fopen("pantalla.txt", "wt");
7
8      /// if(fpPant == NULL)    // si no puede abrir el archivo
9      ///     fpPant = stdout;  // mostrará por consola
10     FILE *fpPant = stdout;
11
12     probarCadenas(fpPant);
13
14     probarMatrices(fpPant);
15
16     probarArchivosYPila(fpPant);
17
18     if(fpPant != stdout && fpPant != stderr && fpPant != NULL)
19     {
20         fclose(fpPant);
21         system("start notepad pantalla.txt");
22     }
23
24     return 0;
25 }
```

. . . del modo que se indica en la figura.



UNLaM

Dto. Ingeniería e Investigaciones Tecnológicas

NOTE que en **funciones.h** no debe hacer ningún cambio. Los prototipos de las funciones y primitivas ya están disponibles.

```

88  /**/
89
90  int buscarYReemplazar_MIO(char *cade, int tam,
91                          const char *busc, const char *reem);
92
93  /**/
94  /**/ PUNTO 2
95
96  int cargarCuadrMagico_MIO(int m[][TAM_MAT], int filas, int colum);
97
98  int mostrarMat_MIO(int m[][TAM_MAT], int filas, int colum, FILE *fp);
99
100  /**/
101  /**/ PUNTO 3
102
103  int procesarPedidos_MIO(FILE *fpPedi, FILE *fpPend, FILE *fpMayo, FILE *fpMeno,
104                        FILE *fpPant);
105
106  void crearPila_MIO(tPila *p);
107
108  int pilaLlena_MIO(const tPila *p, unsigned cantBytes);
109
110  int ponerEnPila_MIO(tPila *p, const void *d, unsigned cantBytes);
111
112  int verTope_MIO(const tPila *p, void *d, unsigned cantBytes);
  
```

Lo único que podría modificar (momentáneamente) es la etiqueta "**TAM_MAT**" para verificar y ver cómo se comporta el *cuadrado mágico*.

```

1  #ifndef FUNCIONES_H_
2  #define FUNCIONES_H_
3
4  #include <stdio.h>
5
6
7  /**/
8  /**/ PUNTO 1
9
10 int buscarYReemplazar(char *cade, int tamCade,
11                     const char *busc, const char *reem);
12
13 /**/
14 /**/ PUNTO 2
15 /**/ con hasta un máximo de 27 tendrá una salida legible por pantalla / archivo
16 /**/ TAM_MAT debe ser IMPAR y mayor o igual a 3 (tres)
17
18 #define TAM_MAT 17
19
20 int mostrarMat(int m[][TAM_MAT], int filas, int colum, FILE *fp);
  
```



UNLaM

Dto. Ingeniería e Investigaciones Tecnológicas

Otras cosas que puede modificar en **main.c**, son las cadenas a buscar y a reemplazar además de la cadena en la que se busca y se reemplaza, del ejercicio de cadenas . . .

```

27 void probarCadenas(FILE *fpPant)
28 {
29     char buscar[] = { "int" },
30                cade[99] = { "int funcion(int seIntegra, int noIntegra);" },
31                reempl[] = { "unsigned int" };
32     int veces = 0,
33         sinEspacio = 0;
34
35     fprintf(fpPant,
36            "////////// Ini - Parte 1 //////////\n"
37            "En la cadena \n\"%s\":\n",
38            cade);
39
40     /* **** */
41     veces = buscarYReemplazar(cade, sizeof(cade), buscar, reempl);
42     /* veces = buscarYReemplazar_MIO(cade, sizeof(cade), buscar, reempl); */
43     /* **** */
44     if(veces < 0)

```

En la figura anterior, en la parte superior, están recuadradas en rojo dónde cambiar por sus propios lotes de prueba para el ejercicio de cadenas.

En la parte inferior está recuadrado también en rojo el lugar dónde invocar a su función. Note que del mismo modo que muestra por pantalla utilizando la función **printf**, puede mostrar por pantalla (o grabar en archivo de texto) con la función **fprintf**, utilizando el puntero a file **fpPant**, como se ve en la siguiente figura . . .

```

35     fprintf(fpPant,
36            "////////// Ini - Parte 1 //////////\n"
37            "En la cadena \n\"%s\":\n",
38            cade);
39
40     /* **** */
41     veces = buscarYReemplazar(cade, sizeof(cade), buscar, reempl);
42     /* veces = buscarYReemplazar_MIO(cade, sizeof(cade), buscar, reempl); */
43     /* **** */
44     if(veces < 0)
45     {
46         veces = -veces - 1;
47         sinEspacio = 1;
48     }
49     fprintf(fpPant,
50            "Se reemplaza la sub cadena \"%s\" por la subcadena \"%s\".\n"
51            "Se hicieron %d reemplazos.\n"
52            "La cadena queda modificada:\n\"%s\".\n"
53            "////////// FIN - Parte 1 //////////\n\n",
54            buscar, reempl,
55            sinEspacio ? " solamente" : "",
56            veces,
57            sinEspacio ? " porque falta espacio en la cadena" : "",
58            cade);
59 }

```

