|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Carrera:** | **Desarrollo de Software** | | | | **Academia:** | **Sistemas Digitales** | **Plantel:** | **Colomos** |
| **Materia:** | **Arquitectura y Organización de Computadoras** | | | | **Clave:** | **18MPBDS0514** | **Revisión:** | **A** |
| **Unidad:** | **3.** **Programación de un sistema basado en microprocesador en un lenguaje de bajo nivel.** | | | | **Tema:** | **La programación.** | | |
| **No. de Práctica:** | **10** | **Nombre de la práctica:** | | **Comparación de letras utilizando la pantalla LCD** | | | | |
| **Profesor:** | **Antonio Lozano González** | | | | | | | |
| **Alumno:** | **Emmanuel Buenrostro Briseño** | | | | | | **Registro:** | **22300891** |
| **Alumno:** |  | | | | | | **Registro:** |  |
| **Semestre:** | **5** | **Grupo:** | **I1** | | **Período:** | **Ago-Dic 2024** | **Fecha:** | **10/12/24** |

**1. Objetivo.** **Utilizar operaciones aritméticas, para encontrar la forma de saber cuál letra  
es mayor, menor o igual a otra e ir aprendiendo a utilizar el carry como  
elemento diferenciador.**

**2. Material, Equipo y/o Herramientas**

|  |  |
| --- | --- |
| **Cantidad** | **Material, Equipo y/o Herramientas** |
| **1** | **Microcontrolador** |
| **1** | **Memoria** |
| **1** | **Latch** |
| **1** | **Buffer** |
| **1** | **Minidip** |
| **8** | **Leds** |

**3. Desarrollo de la práctica**

**a) Condiciones de la práctica:**

Utilizando su sistema mínimo, deberá introducir por el dip switch, una frase que le sera proporcionada por su maestro, entonces por medio de su programación, deberá acomodar de mayor a menor cada letra si su ultimo dígito del registro es par y de menor a mayor cada letra si su ultimo dígito del registro es impar. Los datos introducidos irán en la parte superior de la pantalla y los ordenados en la parte inferior.

**b) Algoritmo o Diagrama de Flujo.**

Lo que hacemos es leer los valores ascii de las letras de 8 bits (leemos de 4 en 4 y los juntamos con un orl y un swap), entonces ahora vamos a aplicar bubble sort, para eso vamos a hacer un “recorrido” de los 16 valores y haciendo una comparación entre dos consecutivos y haciendo los swaps necesarios para que quede el arreglo en orden.

Ese recorrido lo hacemos 16 veces para asegurarnos que el arreglo quede debidamente ordenado y con eso terminamos imprimiendo todos los valores en el orden en que nos dejo el bubble sort.

**c) Código LST.**

comparacion PAGE 1

0000 C2B0 1 pant: CLR P3.0

0002 C2B1 2 CLR P3.1

0004 759038 3 mov P1,#38H

0007 12021C 4 lcall de5ms

000A 759038 5 mov P1,#38H

000D 12021C 6 lcall de5ms

0010 759038 7 mov P1,#38H

0013 12021C 8 lcall de5ms

0016 759038 9 mov P1,#38H

0019 12021C 10 lcall de5ms

001C 75900C 11 mov P1,#0CH

001F 12021C 12 lcall de5ms

0022 759001 13 mov P1,#01H

0025 12021C 14 lcall de5ms

0028 759006 15 mov P1,#06H

002B 12021C 16 lcall de5ms

002E 759080 17 mov P1,#80H

0031 782F 18 inicio: mov R0,#2FH

0033 793F 19 e mov R1,#3FH

0035 120068 20 lcall readc

0038 120068 21 lcall readc

003B 120068 22 lcall readc

003E 120068 23 lcall readc

0041 120068 24 lcall readc

0044 120068 25 lcall readc

0047 120068 26 lcall readc

004A 120068 27 lcall readc

004D 120068 28 lcall readc

0050 120068 29 lcall readc

0053 120068 30 lcall readc

0056 120068 31 lcall readc

0059 120068 32 lcall readc

005C 120068 33 lcall readc

005F 120068 34 lcall readc

0062 120068 35 lcall readc

0065 020089 36 ljmp ord

37

0068 12022C 38 readc: lcall time

006B E5B0 39 mov A,P3

006D 54F0 40 anl A,#0F0H

006F F570 41 mov 70H,A

0071 12022C 42 lcall time

0074 E5B0 43 mov A,P3

0076 54F0 44 anl A,#0F0H

0078 C4 45 swap A

0079 4570 46 orl A,70H

007B 08 47 INC R0

007C 09 48 INC R1

007D F7 49 mov @R1,A

007E F6 50 mov @R0,A

007F D2B0 51 setb P3.0

0081 F590 52 mov P1,A

0083 12021C 53 lcall de5ms

0086 C2B0 54 clr P3.0

0088 22 55 ret

0089 12013A 56 ord: lcall bubble

008C 12013A 57 lcall bubble

008F 12013A 58 lcall bubble

0092 12013A 59 lcall bubble

0095 12013A 60 lcall bubble

0098 12013A 61 lcall bubble

009B 12013A 62 lcall bubble

009E 12013A 63 lcall bubble

00A1 12013A 64 lcall bubble

00A4 12013A 65 lcall bubble

00A7 12013A 66 lcall bubble

00AA 12013A 67 lcall bubble

00AD 12013A 68 lcall bubble

00B0 12013A 69 lcall bubble

00B3 12013A 70 lcall bubble

00B6 12013A 71 lcall bubble

72

00B9 7590C0 73 imp: mov P1,#0C0H

00BC 12021C 74 lcall de5ms

00BF D2B0 75 setb P3.0

00C1 854090 76 mov P1,40H

00C4 12021C 77 lcall de5ms

00C7 854190 78 mov P1,41H

00CA 12021C 79 lcall de5ms

00CD 854290 80 mov P1,42H

00D0 12021C 81 lcall de5ms

00D3 854390 82 mov P1,43H

00D6 12021C 83 lcall de5ms

00D9 854490 84 mov P1,44H

00DC 12021C 85 lcall de5ms

00DF 854590 86 mov P1,45H

00E2 12021C 87 lcall de5ms

00E5 854690 88 mov P1,46H

00E8 12021C 89 lcall de5ms

00EB 854790 90 mov P1,47H

00EE 12021C 91 lcall de5ms

00F1 854890 92 mov P1,48H

00F4 12021C 93 lcall de5ms

00F7 854990 94 mov P1,49H

00FA 12021C 95 lcall de5ms

00FD 854A90 96 mov P1,4AH

0100 12021C 97 lcall de5ms

0103 854B90 98 mov P1,4BH

0106 12021C 99 lcall de5ms

0109 854C90 100 mov P1,4CH

010C 12021C 101 lcall de5ms

010F 854D90 102 mov P1,4DH

0112 12021C 103 lcall de5ms

0115 854E90 104 mov P1,4EH

0118 12021C 105 lcall de5ms

011B 854F90 106 mov P1,4FH

011E 12021C 107 lcall de5ms

0121 C2B0 108 clr P3.0

0123 02022C 109 ljmp time

0126 E570 110 comp: mov A, 70H

0128 8571F0 111 mov B, 71H

012B C3 112 CLR C

012C C2D6 113 ClR AC

012E 95F0 114 subb A,B

0130 5001 115 JNC ncarry

0132 22 116 ret

0133 E571 117 ncarry: mov A,71H

0135 C570 118 xch A,70H

0137 C571 119 xch A,71H

0139 22 120 ret

121

122

013A 854070 123 bubble: mov 70H, 40H

013D 854171 124 mov 71H, 41H

0140 120126 125 lcall comp

0143 857040 126 mov 40H,70H

0146 857141 127 mov 41H,71H

0149 854170 128 mov 70H, 41H

014C 854271 129 mov 71H, 42H

014F 120126 130 lcall comp

0152 857041 131 mov 41H,70H

0155 857142 132 mov 42H,71H

0158 854270 133 mov 70H, 42H

015B 854371 134 mov 71H, 43H

015E 120126 135 lcall comp

0161 857042 136 mov 42H,70H

0164 857143 137 mov 43H,71H

0167 854370 138 mov 70H, 43H

016A 854471 139 mov 71H, 44H

016D 120126 140 lcall comp

0170 857043 141 mov 43H,70H

0173 857144 142 mov 44H,71H

0176 854470 143 mov 70H, 44H

0179 854571 144 mov 71H, 45H

017C 120126 145 lcall comp

017F 857044 146 mov 44H,70H

0182 857145 147 mov 45H,71H

0185 854570 148 mov 70H, 45H

0188 854671 149 mov 71H, 46H

018B 120126 150 lcall comp

018E 857045 151 mov 45H,70H

0191 857146 152 mov 46H,71H

0194 854670 153 mov 70H, 46H

0197 854771 154 mov 71H, 47H

019A 120126 155 lcall comp

019D 857046 156 mov 46H,70H

01A0 857147 157 mov 47H,71H

01A3 854770 158 mov 70H, 47H

01A6 854871 159 mov 71H, 48H

01A9 120126 160 lcall comp

01AC 857047 161 mov 47H,70H

01AF 857148 162 mov 48H,71H

01B2 854870 163 mov 70H, 48H

01B5 854971 164 mov 71H, 49H

01B8 120126 165 lcall comp

01BB 857048 166 mov 48H,70H

01BE 857149 167 mov 49H,71H

01C1 854970 168 mov 70H, 49H

01C4 854A71 169 mov 71H, 4AH

01C7 120126 170 lcall comp

01CA 857049 171 mov 49H,70H

01CD 85714A 172 mov 4AH,71H

01D0 854A70 173 mov 70H, 4AH

01D3 854B71 174 mov 71H, 4BH

01D6 120126 175 lcall comp

01D9 85704A 176 mov 4AH,70H

01DC 85714B 177 mov 4BH,71H

01DF 854B70 178 mov 70H, 4BH

01E2 854C71 179 mov 71H, 4CH

01E5 120126 180 lcall comp

01E8 85704B 181 mov 4BH,70H

01EB 85714C 182 mov 4CH,71H

01EE 854C70 183 mov 70H, 4CH

01F1 854D71 184 mov 71H, 4DH

01F4 120126 185 lcall comp

01F7 85704C 186 mov 4CH,70H

01FA 85714D 187 mov 4DH,71H

01FD 854D70 188 mov 70H, 4DH

0200 854E71 189 mov 71H, 4EH

0203 120126 190 lcall comp

0206 85704D 191 mov 4DH,70H

0209 85714E 192 mov 4EH,71H

020C 854E70 193 mov 70H, 4EH

020F 854F71 194 mov 71H, 4FH

0212 120126 195 lcall comp

0215 85704E 196 mov 4EH,70H

0218 85714F 197 mov 4FH,71H

021B 22 198 ret

199

200

201

021C D2B1 202 de5ms: SETB P3.1

021E 7F0A 203 mov R7, #0AH

0220 7EFA 204 aca: mov R6,#0FAH

0222 DEFE 205 aqui2: DJNZ R6, aqui2

0224 DFFA 206 DJNZ R7, aca

0226 C2B1 207 CLR P3.1

0228 22 208 RET

209

210

0229 020229 211 time2: ljmp time2

022C 7F40 212 time: mov R7,#40H

022E 7EFA 213 paca: mov R6,#0FAH

0230 7DFA 214 aca2: MOV R5,#0FAH

0232 DDFE 215 aqui3: DJNZ R5,aqui3

0234 DEFA 216 djnz R6,aca2

0236 DFF6 217 djnz R7,paca

0238 22 218 ret

\*\*\*\*WARNING:Missing`END'directive

ASSEMBLY COMPLETE, NO ERRORS FOUND, 1 WARNING

**5. Observaciones y Conclusiones**

|  |
| --- |
| **Conclusiones** |
| **Definitivamente es el bubble sort mas dificil que e programado, aunque se me hizo bastante divertido** |
| **El uso de los apuntadores e ir recorriendo en las direcciones para recorrer los 16 valores e ir haciendo** |
| **Los swaps, en los 16 recorrido que hacemos en total.** |
|  |
|  |