**Computer Organization & Assembly Language**

**Lab 13**

**Tasks:**

1. Use keyboard interfacing to find out if any key is pressed or not.

.model small

.stack 100h

.data

.code

mov ax,@data

mov ds,ax

l1:

mov ah,01

int 16h

cmp al,13

jz again

again:

mov ah,01

int 16h

cmp al,13

mov dl,'Yes Key is not entered'

mov ah,2

int 21h

je exit

loop l1

exit:

mov dl,'Yes Key is entered'

mov ah,2

int 21h

mov ax,4ch

int 21h

end

1. Use keyboard interfacing to find out which key is pressed.

.model small

.stack 100h

.data

.code

mov ax,@data

mov ds,ax

mov ah,0

int 16h

mov dl,al

mov ah,2

int 21h

mov ax,4ch

int 21h

end

1. Use keyboard interfacing interrupt to and draw pixels as you press the up, down, left, and right arrow keys.

.model small

.stack 100h

.data

.code

mov ax,@data

mov ds,ax

l1:

mov ah,01

int 16h

MOV AH,1 ;; input

INT 21H

cmp al,48h ;up

jz exit

je point

cmp al,4Bh ;left

jz exit

je point

cmp al,4Dh ;right

jz exit

je point

cmp al,50h ;down

jz exit

je point

;;;;;point right

point:

mov bx, 153

mov cx,3h

mov dx,23

push cx

mov cx,bx

mov ah,0ch

mov al,14

int 10h

pop cx

exit:

mov dl,'No Key is entered'

mov ah,2

int 21h

mov ax,4ch

int 21h

end

1. On key press, only the tiles having first letters of your name should turn red. All other tiles should stay same.
2. .MODEL SMALL
3. .STACK 100H
4. .DATA
5. .CODE
6. mov ah,00h
7. mov al,13h
8. int 10h
9. ;;;;;;Squaree1
10. ;;;;;leftt
11. mov bx, 30
12. mov cx,10h
13. mov dx,20
14. l1:
15. push cx
16. mov cx,bx
17. mov ah,0ch
18. mov al,14
19. int 10h
20. inc dx
21. pop cx
22. loop l1
23. ;;;;;;;;;;;rightt
24. mov bx, 50
25. mov cx,10h
26. mov dx,20
27. l2:
28. push cx
29. mov cx,bx
30. mov ah,0ch
31. mov al,14
32. int 10h
33. inc dx
34. pop cx
35. loop l2
36. ;;;;;;;;;uppp
37. mov bx, 30
38. mov cx,15h
39. mov dx,20
40. l3:
41. push cx
42. mov cx,bx
43. mov ah,0ch
44. mov al,14
45. int 10h
46. inc bx
47. pop cx
48. loop l3
49. ;;;;;;;;;;down
50. mov bx, 30
51. mov cx,15h
52. mov dx,36
53. l4:
54. push cx
55. mov cx,bx
56. mov ah,0ch
57. mov al,14
58. int 10h
59. inc bx
60. pop cx
61. loop l4
62. ;;;;;;Squaree2
63. ;;;;;leftt
64. mov bx, 50
65. mov cx,10h
66. mov dx,20
67. l5:
68. push cx
69. mov cx,bx
70. mov ah,0ch
71. mov al,14
72. int 10h
73. inc dx
74. pop cx
75. loop l5
76. ;;;;;;;;;;;rightt
77. mov bx, 70
78. mov cx,10h
79. mov dx,20
80. l6:
81. push cx
82. mov cx,bx
83. mov ah,0ch
84. mov al,14
85. int 10h
86. inc dx
87. pop cx
88. loop l6
89. ;;;;;;;;;uppp
90. mov bx, 50
91. mov cx,15h
92. mov dx,20
93. l7:
94. push cx
95. mov cx,bx
96. mov ah,0ch
97. mov al,14
98. int 10h
99. inc bx
100. pop cx
101. loop l7
102. ;;;;;;;;;;down
103. mov bx, 50
104. mov cx,15h
105. mov dx,36
106. l8:
107. push cx
108. mov cx,bx
109. mov ah,0ch
110. mov al,14
111. int 10h
112. inc bx
113. pop cx
114. loop l8
115. ;;;;;;Squaree3
116. ;;;;;leftt
117. mov bx, 70
118. mov cx,10h
119. mov dx,20
120. l9:
121. push cx
122. mov cx,bx
123. mov ah,0ch
124. mov al,14
125. int 10h
126. inc dx
127. pop cx
128. loop l9
129. ;;;;;;;;;;;rightt
130. mov bx, 90
131. mov cx,10h
132. mov dx,20
133. l10:
134. push cx
135. mov cx,bx
136. mov ah,0ch
137. mov al,14
138. int 10h
139. inc dx
140. pop cx
141. loop l10
142. ;;;;;;;;;uppp
143. mov bx, 70
144. mov cx,15h
145. mov dx,20
146. l11:
147. push cx
148. mov cx,bx
149. mov ah,0ch
150. mov al,14
151. int 10h
152. inc bx
153. pop cx
154. loop l11
155. ;;;;;;;;;;down
156. mov bx, 70
157. mov cx,15h
158. mov dx,36
159. l12:
160. push cx
161. mov cx,bx
162. mov ah,0ch
163. mov al,14
164. int 10h
165. inc bx
166. pop cx
167. loop l12
168. ;;;;;;Squaree4
169. ;;;;;leftt
170. mov bx, 90
171. mov cx,10h
172. mov dx,20
173. l13:
174. push cx
175. mov cx,bx
176. mov ah,0ch
177. mov al,14
178. int 10h
179. inc dx
180. pop cx
181. loop l13
182. ;;;;;;;;;;;rightt
183. mov bx, 110
184. mov cx,10h
185. mov dx,20
186. l14:
187. push cx
188. mov cx,bx
189. mov ah,0ch
190. mov al,14
191. int 10h
192. inc dx
193. pop cx
194. loop l14
195. ;;;;;;;;;uppp
196. mov bx, 90
197. mov cx,15h
198. mov dx,20
199. l15:
200. push cx
201. mov cx,bx
202. mov ah,0ch
203. mov al,14
204. int 10h
205. inc bx
206. pop cx
207. loop l15
208. ;;;;;;;;;;down
209. mov bx, 90
210. mov cx,15h
211. mov dx,36
212. l16:
213. push cx
214. mov cx,bx
215. mov ah,0ch
216. mov al,14
217. int 10h
218. inc bx
219. pop cx
220. loop l16
221. mov ah,02h
222. mov dh,3
223. mov dl,5
224. int 10h
225. mov ah,09h
226. mov al,'Z'
227. mov bl,12
228. mov cx,1
229. int 10h
230. mov ah,02h
231. mov dh,3
232. mov dl,7
233. int 10h
234. mov ah,09h
235. mov al,'R'
236. mov bl,12
237. mov cx,1
238. int 10h
239. mov ah,02h
240. mov dh,3
241. mov dl,9
242. int 10h
243. mov ah,09h
244. mov al,'M'
245. mov bl,12
246. mov cx,1
247. int 10h
248. mov ah,02h
249. mov dh,3
250. mov dl,12
251. int 10h
252. mov ah,09h
253. mov al,'A'
254. mov bl,2
255. mov cx,1
256. int 10h
257. mov ah,4ch
258. int 21h
259. end

