“Heaven’s light is our guide”

**Rajshahi University of Engineering & Technology**

Department of

Computer Science & Engineering

**Course No: CSE 3110**

**Course Title: Microprocessors and Assembly Language Sessional**

**Lab Report (Lab 4)**

|  |  |
| --- | --- |
| Submitted By:  ***Saifur Rahman***  *Roll No:* ***1703018***  *Section: ’17-A*  *Class: 3rd year (Odd Semester)* | Submitted To:  ***Sadia Zaman Mishu***  *Assistant Professor,*  *Dept. of CSE,*  *RUET.* |

***Date of Experiment: 9th Mar, 2021***

***Date of Submission: 17th Mar, 2021***

Experiment no: 01

Name of the experiment:

**Write an assembly language program that will prompt the user to enter a hex digit character (“0”,......”9” or “A”,.... “F”), display it on the next line in decimal and check whether this decimal number is odd or not. If odd then the program will repeat and if even then the program will terminate. If the user enters an illegal character, prompt the user to enter another character.**

Objectives:

* To take input a hex character
* To run a loop with condition
* To check whether the input is even or odd
* To check whether a input character is valid or not

Code:

1. .MODEL **SMALL**
2. .**STACK** 100H
3. **.DATA**
5. **CR EQU 0DH**
6. LF **EQU** 0AH
8. MSG1 **DB** 0DH, 0AH, 'Enter the hex digit: $'
9. MSG2 **DB** 0DH, 0AH, 'In decimal it is: $'
10. **MSG3 DB ' is an odd number $'**
11. MSG4 **DB** ' is an even number $'
12. MSG5 **DB** ' is illegal character$'
13. MSG6 **DB** 0DH, 0AH, '$'

16. .**CODE**
17. MAIN PROC
19. *;initialize DS*
20. **MOV AX, @DATA *;get data segment***
21. **MOV** **DS**, **AX** *;initialize DS*
23. **START**:
24. *;print user prompt*
25. **LEA DX, MSG1 *;prompt first message***
26. **MOV** **AH**, 9 *;display string function*
27. **INT** 21H *;display message*
29. *;input and store characters*
30. **MOV AH, 1 *;read character function***
31. **INT** 21H *;read character*

34. **MOV** **CL**, **AL** *;store character's Ascii value*
36. **CMP** **CL**, 48D *;if input is less than 0*
37. **JL** ILLEGAL *;then illegal input*
39. **CMP** **CL**, 70D *;if input is greater than F*
40. **JG ILLEGAL *;then illegal input***
42. **CMP** **CL**, 57D *;compares if input is less than*
43. **JLE** DIGIT *;or equal to 9*
45. **SUB CL, 17D *;subtract 17 to match the ASCII value***
46. *;with 0 - 5 of decimal digit*
47. *;print*
48. **LEA** **DX**, MSG2 *;prompt second message*
49. **MOV** **AH**, 9 *;display string function*
50. **INT 21H *;display message***
52. **MOV** **DL**, 49D *;display 1 at the first*
53. **MOV** **AH**, 2 *;display character function*
54. **INT** 21H *;display character*
56. **MOV** **DL**, **CL** *;display the stored character*
57. **MOV** **AH**, 2 *;display character function*
58. **INT** 21H *;display character*
60. **LEA DX, MSG6 *;prompt new line***
61. **MOV** **AH**, 9 *;display string function*
62. **INT** 21H *;display message*
64. **MOV** **DL**, 49D *;display 1 at the first*
65. **MOV AH, 2 *;display character function***
66. **INT** 21H *;display character*
68. **MOV** **DL**, **CL** *;display the stored character*
69. **MOV** **AH**, 2 *;display character function*
70. **INT 21H**
72. **MOV** **BX**, 2 *;get 2 in bx*
73. **XOR** **DX**, **DX** *;to avoid overflow*
74. **DIV** **BX** *;divide ax with bx (2)*
76. **CMP** **DX**, 0H *;dx remainder*
77. *;compare remainder with 0*
78. **JE** EVEN *;if input is even, end loop*

81. **LEA** **DX**, MSG3 *;prompt third message*
82. **MOV** **AH**, 9 *;display string function*
83. **INT** 21H *;display message*
85. **JMP START *;exit***
87. DIGIT:
88. *;print*
89. **LEA** **DX**, MSG2 *;prompt second message*
90. **MOV AH, 9 *;display string function***
91. **INT** 21H *;display message*
93. **MOV** **DL**, **CL** *;display the stored character*
94. **MOV** **AH**, 2 *;display character function*
95. **INT 21H *;display character***
97. **LEA** **DX**, MSG6 *;prompt new line*
98. **MOV** **AH**, 9 *;display string function*
99. **INT** 21H *;display message*
101. **MOV** **DL**, **CL** *;display the stored character*
102. **MOV** **AH**, 2 *;display character function*
103. **INT** 21H

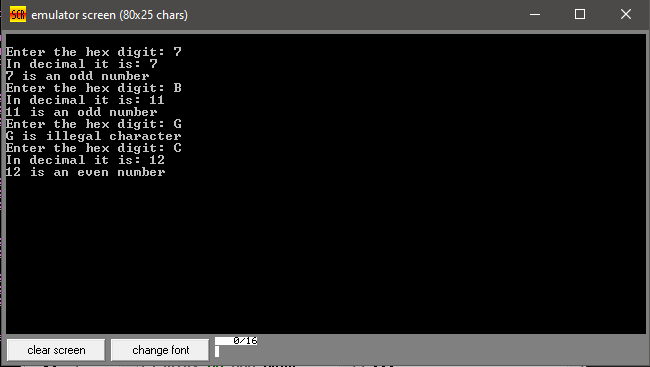
106. **XOR** **DX**, **DX** *;to avoid overflow*
107. **MOV** **BX**, 2 *;get 2 in bx*
108. **DIV** **BX** *;divide ax with bx (2)*
110. **CMP DX, 0H *;dx remainder***
111. *;compare remainder with 0*
112. **JE** EVEN *;if input is even, end loop*

115. **LEA DX, MSG3 *;prompt third message***
116. **MOV** **AH**, 9 *;display string function*
117. **INT** 21H *;display message*
118. **JMP** **START**

121. ILLEGAL:
122. **LEA** **DX**, MSG6 *;prompt new line*
123. **MOV** **AH**, 9 *;display string function*
124. **INT** 21H *;display message*
126. **MOV** **DL**, **CL**
127. **MOV** **AH**, 2 *;display character function*
128. **INT** 21H *;display character*
130. **LEA DX, MSG5 *;prompt fifth message***
131. **MOV** **AH**, 9 *;display string function*
132. **INT** 21H *;display message*

135. **JMP START *;else loop continues***
137. EVEN:
139. **LEA** **DX**, MSG4 *;prompt fourth message*
140. **MOV AH, 9 *;display string function***
141. **INT** 21H *;display message*
143. *;DOS exit*
144. **MOV** **AH**, 4CH
145. **INT 21H**
147. MAIN ENDP
148. END MAIN

Output:



Discussion:

First, it is checked that the input character is legal or not. Then it is checked that the input is from 0-9 or A-F hex digit. If input is 0 – 9 it is directly converted into decimal and checked whether even or not. And if it is from A-F then 17d is subtracted from it then it is printed and checked by same process whether it is even or odd. Then the loop continues until an even input is given.