****

**Green University of Bangladesh**

**Department of Computer Science and Engineering (CSE)**

**Faculty of Sciences and Engineering**

**Semester: (Fall: Year 2023), B.Sc. in CSE (Day)**

**Lab Report No:** 02

**Course Title:** Microprocessor & Microcontroller Lab

**Course Code:** CSE 304 **Section:** 213D2

**Lab Experiment Name:** Implementation of conditional statement

using assembly language

**Student Details**

|  |  |  |
| --- | --- | --- |
| **Name** | | **ID** |
| **1.** | Md Masud Rana | 213902101 |

**Lab Date :** 12-10-2023

**Submission Date :** 19-10-2023

## **Course Teacher’s Name :** Sudip Ghoshal

|  |
| --- |
| **Lab Report Status**  **Marks: ………………………………… Signature:.....................**  **Comments:.............................................. Date:..............................** |

**1. TITLE OF THE LAB REPORT EXPERIMENT**

Implementation of conditional statement using assembly language

**2. OBJECTIVES/AIM**

* To understand basic conditional statements in assembly language
* To understand about conditional branching
* To understand about conditional and unconditional jumping
* To understand about level using jumping

**3. PROCEDURE**

**Problem-1: Check a character vowel or consonant**

**Step-1:** Define the necessary data strings for messaging.

**Step-2:** Read the input character from the user.

**Step-3:** Compare the input character with various vowels (both uppercase and lowercase) using conditional statements.

**Step-4:** If the character is a vowel, print "Vowel" and return.

**Step-5:** else the character is a consonant, print "Consonant" and return.

**Step-6:** Terminate the program using the DOS interrupt (int 21h, function 4Ch).

**Problem-2: Check whether an input is an Alphabet or digit or others**

**Step-1:** Declare the necessary data strings for messaging and a newline.

**Step-2:** Load the data segment and move it to the data register.

**Step-3:** Prompt the user to enter an input character.

**Step-4:** Check if the input character is a digit (ASCII range 30h to 39h).

**Step-5:** If it is not a digit, check if it is an uppercase letter (ASCII range 41h to 5Ah).

**Step-6:** If it is not an uppercase letter, check if it is a lowercase letter (ASCII range 61h to 7Ah).

**Step-7:** Print the appropriate message based on the category of the input character

**4. IMPLEMENTATION**

**Problem-1:** Check a character vowel or consonant

|  |  |
| --- | --- |
| ; check a character is vowel or consonant  org 100h  .model small  .stack 100h  .data  msg1 db "Enter Character : $"  vowel db "Vowel$"  const db "Consonant$"  newline db 0dh, 0ah, "$"  .code  main proc  mov ax,@data  mov ds,ax  mov ah, 9  lea dx, msg1  int 21h  mov ah,1h  int 21h    ; print a newline  mov ah, 9  lea dx, newline  int 21h    ; below is conditional statements for various conditon with level  cmp al,'A'  je vwl  cmp al,'a'  je vwl | cmp al,'E'  je vwl  cmp al,'e'  je vwl  cmp al,'I'  je vwl  cmp al,'i'  je vwl  cmp al,'O'  je vwl  cmp al,'o'  je vwl  cmp al,'U'  je vwl  cmp al,'u'  je vwl    mov ah,9  lea dx, const  int 21h  ret  vwl:  mov ah,9  lea dx, vowel  int 21h  ret  Quit:  mov ah,4ch  int 21h  main endp |

**Output:**

|  |
| --- |
|  |

*Figure-1: Output snapshot in console to print a character is vowel or consonant*

**Problem-2:** Whether an input is alphabet or digits or others

|  |  |
| --- | --- |
| org 100h  .model small  .stack 100h  .data  msg1 db 'Alphabet$'  msg2 db 'Digit$'  msg3 db 'others$'  msg4 db 'Enter Input:$'  newline db 0dh, 0ah, "$"  .code  main proc  MOV AX, @DATA  MOV DS, AX  mov ah, 9  lea dx, msg4  int 21h  ; Read character  mov ah, 01h  int 21h  mov bl, al  ; Print newline  mov ah, 9  lea dx, newline  int 21h  ; Check character category  cmp bl, 30h ; Check if it's a digit  JAE check\_alpha  others:  mov ax, @data  mov ds, ax  mov ah, 9  lea dx, msg3  int 21h  jmp end  check\_alpha:  ; Compare with ASCII for digit '9'  cmp bl, 39h  JBE print2 | ; Compare with ASCII for uppercase letter 'A'  cmp bl, 41h  JAE check\_lower  print2:  mov ax, @data  mov ds, ax  mov ah, 9  lea dx, msg2  int 21h  jmp end  check\_lower:  ; Compare with ASCII for uppercase letter 'Z'  cmp bl, 5Ah  JBE print3  small\_letter:  ; Compare with ASCII for lowercase letter 'a'  cmp bl, 61h  JAE small\_letter\_1  print3:  mov ax, @data  mov ds, ax  mov ah, 9  lea dx, msg1  int 21h  jmp end  small\_letter\_1:  ; Compare with ASCII for lowercase letter 'z'  cmp bl, 7Ah  JBE print4  print4:  mov ax, @data  mov ds, ax  mov ah, 9  lea dx, msg1  int 21h  end:  ret  main endp  endp |

**Output:**

|  |
| --- |
|  |

*Figure-2: Output of an input whether this input is an alphabet or digit or others*

**5. TEST RESULT / OUTPUT**

For the program checking whether a character is a vowel or not:

Input: 'a'

Output: " vowel."

Input: 'j'

Output: " consonant."

For the program determining whether the input is an alphabet or digit:

Input: '7'

Output: " digit."

Input: '('

Output: " ohters."

**6. ANALYSIS AND DISCUSSION**

**Vowel Checker Program:**

This program effectively checks whether the given character is a vowel by comparing it with the set of vowel characters ('a', 'e', 'i', 'o', 'u').

It provides accurate results for all inputs within the specified range.

The code execution is simple and efficient, involving basic conditional checks.

**Alphabet or Digit Program:**

The program accurately determines whether the input is an alphabet or a digit based on the ASCII values of the characters.

It handles both lower and uppercase letters effectively, distinguishing them from digits.

The logic is straightforward, involving ASCII range comparisons to categorize the input.

**7. SUMMARY**

The vowel-checking program reliably determines whether a given character is a vowel, providing the expected results for the specified inputs.

The alphabet-or-digit identification program effectively differentiates between alphabetic and numeric inputs, offering the correct classification based on the ASCII values of the characters.

Both programs demonstrate the efficient use of conditional checks to classify characters. They showcase the fundamental principles of conditional branching in assembly language, enabling accurate classification based on specific criteria.