****

**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:** 05

**Course Title:** Microprocessor & Microcontroller Lab

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

**Lab Experiment Name:** Implement Array and String in Assembly Language Programming

**Student Details**

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

**Lab Date :** 26-10-2023

**Submission Date :** 29-11-2023

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

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

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

Implement Array and String in Assembly Language Programming

**2. OBJECTIVES/AIM**

* To understand the use of Array in Assembly Language
* To understand the use of String in Assembly Language
* To understand ‘dup’ to create duplicate array or string as well as array size initialize
* To understand of some Pointer register like SI, DI, BX, BP

**3. PROCEDURE**

**Problem: Sum of even and odd number from natural number series 1+2+3…+n**

Step 1: start

Step 2: initialize data segment arr of 10 size, I, temp to 0 and oddSum,evenSum to empty

Step 3: Include data segment into main procedure into ‘DS’ register

Step 4: take array size from user and copy to ‘bl’ and ‘cl’ register to set ‘cx’ value

Step 5: set pointer or index register to the ‘arr’ using ‘si’.

Step 6: continue loop. That will continue cx=cl to 0. **‘taken\_input’**

Step 7: Take input and store arry [si], al.

Increment ‘si’ pointer by 1 ‘inc si’. Print a space between each input

Continue **‘loop taken\_input’** and repeat **step-7**

Step 8: again point array using ‘si’ for loop to calculate even and odd number summation

Step 9: set ‘dl’ as 2 to divide each element of the array to check even or odd number

Step 10: take ‘arr\_traverse’ level to jumping conditionally again and again

Initially set ‘ah’ as 0.

Take element from array using [si] into ‘al’.copy ‘al’ to ‘temp’ variable for future

Now divide ‘al’ by ‘dl’ note: ah=remainder and al= quotient

Compare ‘ah’ with 0 or not. If 0 this is even number else odd number

Jump ‘JE’ even\_sum :**jump equal to 0**

Jump ‘JNE’ odd\_sum **jump not equal to 0**

After jumping we need to get back into ‘arr\_traverse’ loop. So take a level ‘back’

After backing, increment variable ‘i’ by 1 and ‘inc si’

Compare ‘i’ with ‘bl’. If equal called ‘print\_result’ level

If less then again called ‘arr\_traverse’ level

Step 11: continue **step-10**

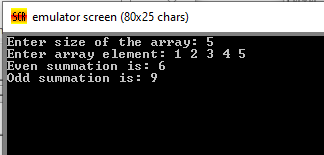
Step 12: print even and odd summation into console.

Step 13: end

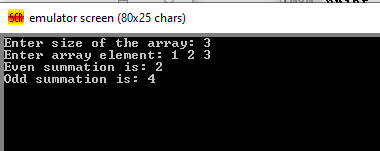
**4. IMPLEMENTATION**

**Problem: Sum of even and odd number from natural number series 1+2+3…+n**

|  |  |
| --- | --- |
| ;take array element from user and size as well  include 'emu8086.inc';emu8086 library  .stack 100h  .model small  .data  arr db 100 dup(?) ;100 size empty array  i db 0  temp db 0  oddSum db ?  evenSum db ?  newline db 10, 13, '$'    .code  main proc  ;import data segment  mov ax, @data  mov ds, ax    mov si, offset arr ;set pointer to the array    ;take array size from user  print 'Enter size of the array: '  mov ah, 1  int 21h  mov bl, al ;store input size into bl register  sub bl, 48    mov cx, 0  mov cl, bl ;cx=user input size    ;print a newline  mov ah, 9  lea dx, newline  int 21h    print 'Enter array element: '  take\_input:  mov ah, 1  int 21h  mov [si], al  inc si  ;pirnt a space between each input  mov ah, 2  mov dl, 32  int 21h  loop take\_input  **Note:** end of this block now read right side block | ;point array usnig source index register  mov si, offset arr  mov dl, 2 ;to divide each arr element    ;traverse & calculate odd,even sum  arr\_traverse:  mov ah, 0  mov al, [si]  ;temporary store current element  mov temp, al  ;after divide by 2 ah=remainder al=quotien  div dl  cmp ah, 0  JE even\_sum  JNE odd\_sum  back:  inc i  inc si  cmp i, bl ;cmp array size and I(incremented value)  JL arr\_traverse  JE print\_result    ;even number calculation  even\_sum:  mov al, temp  add al, evenSum  sub al, 48  mov evenSum, al  jmp back    ;odd number calculation  odd\_sum:  mov al, temp  add al, oddSum  sub al, 48  mov oddSum, al  jmp back  ;print the result  print\_result:  print 'Even summation is: '  mov ah, 2  mov dl, evenSum  add dl, 48  int 21h      print 'Odd summation is: '  mov ah, 2  mov dl, oddSum  add dl, 48  int 21h  main endp  end main |

**Output:**

*Figure-1: Summation of even and odd from* ***5th*** *natural number series using array*

**5. TEST RESULT / OUTPUT**

*Figure-2: Summation of even and odd from* ***3th*** *natural number series using array*

- The program computes the sum of even and odd of the natural numbers from 1 to 3.

- Result will be stored into ‘evenSum’ and ‘oddSum’ variable

- here I have taken 3th natural number where even number only 2 so evenSum is 2 ( figure-2)

This series contain only 1 and 3 even number so sum should be 4. Yes oddSum is 4 (figure-2)

- I have taken two input of natural number series of **5th** (figure-1) and **3th** (figure-2) position.

- both test cases I have got the right answer what actually should be.

**6. ANALYSIS AND DISCUSSION**

* The problem, involving the summation of even and odd number from a natural number series, natural number should be taken from input and store into array. Position of the series taken from user and effectively utilizes a loop to taken from user. And I have perfectly completed this task and my program run properly and give right answer for any natural number series. I have tested two position natural series 5th and 3th. So that I can show answer properly in console. I I could take natural number series more than 6 then it will print the special character since we have only 0 to 9 number for ascii value. I can show the answer in variable for any number of position natural series.
* I have run two input cases and I have got the right answer that is showed into figure-1 and figure-2. I have discussed in section ‘Test Result and Output’section

**7. SUMMARY**

I have completed given **Lab Report** task successfully. It takes much time from me to find out even and odd number from array. After spending I have succeeded to complete this problem.

By doing this lab report I have got some knowledge about array and string problem as well as I have got chance to recover loop concept. As well as pointer register uses cases. Overall I have got some new idea and new technique and new things from this lab problem. the solutions effectively demonstrate the logical flow and control required to tackle various computational challenges.