Green University of Bangladesh

Department of Computer Science and Engineering(CSE)

Faculty of Sciences and Engineering

Semester: (Spring, Year: 2021), B.Sc. in CSE (Day)

LAB REPORT NO : 04

Course Title: Microprocessors and Microcontrollers Lab

Course Code: CSE 304 **Section:** PC-DD

Lab Experiment Name: Introduction of understanding the use of array.

Student Details

Name	Id
Md. Romzan Alom	201902144

Lab Date: 14.11. 2021

Submission Date: 28.11.2021

Course Teacher's Name: Rusmita Halim Chaity

[For Teachers use only: Don't Write Anything inside this box]

Lab Report Status	
Marks: ……	Signature:.....
Comments:.....	Date:.....

Title of the Lab Experiment: Introduction of understanding the use of array.

Objectives / Aim:

We learn about array from this experiment. We can take user input as elements of array index and we can perform various operations on this input and show the result as output.

Procedure / Analysis / Design:

Instruction MOV AX, BX is used moves data BX to AX and data is stored in AX.

Instruction DIV BX is used division. Here, AX is divided by BX and Quotient is stored in AL and Remainder is stored in AH.

Instruction ADD AX, BX is used addition. Here, add to BX and AX and this data is stored in AX.

Instruction SUB AX, BX is used subtraction. Here, BX and AX are subtracted and this data is stored in AX.

Instruction MOV AH, 1 and INT 21H are used for user input and it stored AX.

Instruction MOV DX, AX and MOV AH, 2 and INT 21H are used for show output. Here, DX store AX data and compiler understand DX data then it show DX data as output.

Instruction LEA DX, R and MOV AH, 9 and INT 21H are used for call variable and show output the variable data.

Instruction MOV AH,2 and MOV DL,0DH and INT 21H and MOV DL,0AH and INT 21H are used for newline.

Instruction MOV AH,2 and MOV DL,0DH and INT 21H and MOV DL,0AH and INT 21H are used for newline.

Instruction Array DB N (?) are used for initialize an array.

Instruction CMP AX, BX are used for compare AX and BX.

Instruction JZ and JNZ used for CMP condition.

Instruction RESULT_1: are created a level.

Instruction LOOP START are used for looping. It goes to START level and this level decrease with CX.

Instruction V DB 0AH, 0DH, “The result is: $" are used for newline and take a message. Here, 0AH, 0DH, is used newline.

Problem-01

Write an assembly language code to take natural number series as input and as output, show:

a. The summation of odd numbers:

b. The summation of even numbers:

Pseudo-code:

```
.MODEL SMALL

.STACK 100H

.DATA


n db ?

odd db ?

even db ?

R db ?
```

```
z db ?
A db n dup (?)
    R1 db "Enter the number of input: $"
    R2 db 0ah,0dh,"Enter the all element of index: $"
    R3 db 0ah,0dh,"The result of our pragram: $"
    R4 db 0ah,0dh,"$"
    R5 db 0ah,0dh,"The summation of odd: $"
    R6 db 0ah,0dh,"The summation of even: $"
  .CODE
            MAIN PROC
mov ax, @DATA
mov ds, ax
  mov odd,0
  mov even, 0
  mov bx,0
  lea dx,R1
  mov ah,9
  int 21H
  mov ah,1
  int 21H
  sub al,30H
  mov n,al
```

```
    lea dx,R4
    mov ah,9
    int 21H
    lea dx,R2
    mov ah,9
    int 21H
    xor cx,cx
    mov cl,n
    mov si,0
    Loop_1:
    mov ah,1
    int 21H
    sub al,30H
mov A[si],al
inc si
loop Loop_1
    lea dx,R4
    mov ah,9
    int 21H
    lea dx,R3
    mov ah,9
    int 21H
```

```
xor cx,cx
mov cl,n
mov z,2
mov si,0
Loop_2:
  mov ax,00
  mov al,A[si]
  div z
  mov di,si
  inc si
 cmp AH,00
 jz RESULT_1
 jnz RESULT_2
  RESULT_1:
    mov bl,even
    add bl,A[di]
    mov even,bl
  RESULT_2:
loop Loop_2
xor cx,cx
mov cl,n
mov si,0
```

```
Loop_3:
  mov AX,00
  mov AL,A[si]
  div z
  mov di,si
  inc si
 cmp AH,00
 jnz RESULT_3
 jz RESULT_4
  RESULT_3:
    mov bl,odd
    add bl,A[di]
    mov odd,bl
  RESULT_4:
loop Loop_3
  lea dx,R4
  mov ah,9
  int 21H
  lea dx,R5
  mov ah,9
  int 21H
  mov R,10
```

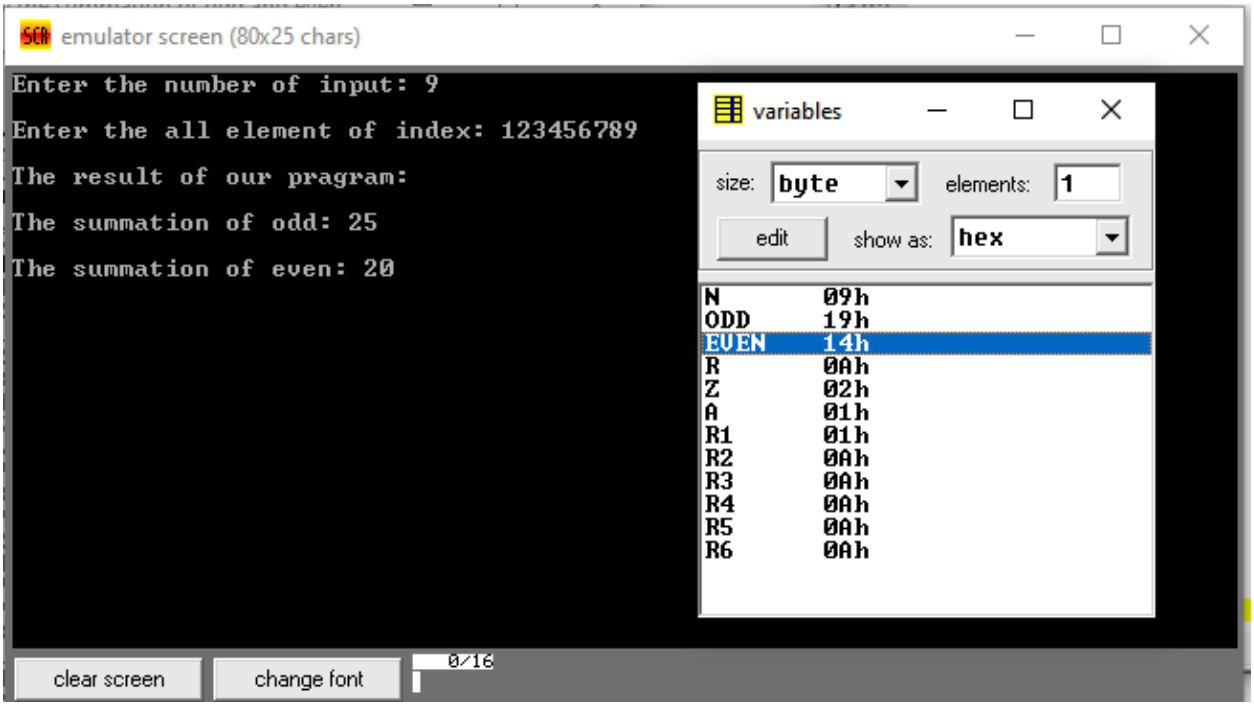
```
xor ax,ax
mov al,odd
div R
mov bh,ah
mov ah,2
mov dl,al
add dl,30H
int 21H
mov ah,2
mov dl,bh
add dl,30H
int 21H
lea dx,R4
mov ah,9
int 21H
lea dx,R6
mov ah,9
int 21H
xor ax,ax
mov al,even
div R
mov bh,ah
```

```
    mov ah,2
    mov dl,al
    add dl,30H
    int 21H
    mov ah,2
    mov dl,bh
    add dl,30H
    int 21H
mov ah, 4ch
int 21h
MAIN ENDP
END MAIN
```

Calculation:

We will try to find the summation of odd and even number from a serial number. For check even or odd we use a condition. When the number % 2 = 0, this number is even otherwise odd.

Test Result / Output:



The serial numbers are 1,2,3,4,5,6,7,8,9
The summation of odd is 25 and it should have been = (1+3+5+7+9) = 25.
The summation of even is 20 and it should have been = (2+4+6+8) = 20.
We see our output and calculation output are same. So, it is 100% right outputs.

Analysis and Discussion:

1. Due to Covid-19 situation, we can't do this experiment directly. So, it is completely based on software.

2. Since, it is done with Software. So it may have some Software and Mechanical errors.

3. To emulate those codes I am facing so many problems for hexadecimal number.

4. From this problem, we use an array. When we operate that array index, we facing some many problem.

5. We can use loop but loop decreases with CX register. That is really confusing to think when loop will stop.

6. To divide value of array index we face some problem.

7. From our compile show only char value of hexadecimal number. That why we face some problem to add array index.

8. Network problem. Cause of the ups and downs of the internet we could not attend the class properly. That why I saw the class recording but this video was very bad quality. This reason I have so many confusion.

SUMMARY:

From this problem, we can take so many decimal numbers from user as input and also take an array from user and print the summation or perform various operations on this input and show the result as output very easily which will help the user to change the program's output. Those are very important to complete this course.