**Experiment Name**

Write a program to that takes a string as input and prints all the lowercase characters of that string in the output.

**Theory**

As assembly language instructions are so basic, Input/output is much harder in assembly language than high level languages. DOS functions are used to take I/O. In assembly language there are operation field, operand field, variables and they work with different registers. There are CMP, JG, JL, JE and JMP instructions available in assembly language for Comparing to operand, Jump to a specific line if first variable is greater than the second, Jump to a specific line if first variable is lesser than the second, Jump to a specific line if first variable is equal to the second and to jump to a line if none of the conditions become true respectively. Here, FOR: was used as for loop to handle the Array. For input and output, MOV AH,1 and MOV AH,2 are used respectively in assembly language.

**Code:**

.MODEL SMALL

.STACK 100H

.DATA

PROMPT DB 'Type a string: ','$'

ARR DB 10 DUB (?)

A DB ?

.CODE

MAIN PROC

MOV AX, @DATA

MOV DS, AX

MOV AH,9

LEA DX,PROMPT

INT 21H

XOR BX, BX

MOV CX, 6

FOR:

MOV AH, 1

INT 21H

MOV ARR[BX], AL

INC BX

LOOP FOR

MOV AH,2 ;for new line after ther input

MOV DL,0DH

INT 21H

MOV DL,0AH

INT 21H

XOR BX, BX

MOV CX, 6

PRINT:

MOV CL, ARR[BX] ;point to the current index

CMP CL,61h

JL END\_IF

CMP CL,00H

JE EXIT

CMP CL,7ah

JG END\_IF

MOV AH, 2 ;output

MOV DL, CL

INT 21H

END\_IF:

INC BX

;move pointer to the next element

LOOP PRINT ;loop until done

EXIT:

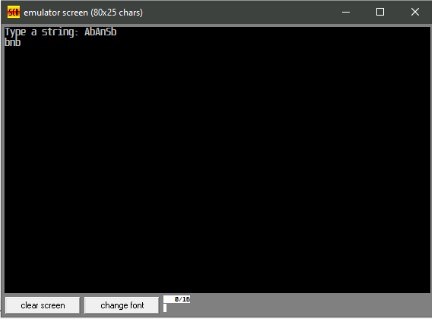
MOV AH,4CH ;this is for DOS exiting

INT 21H

MAIN ENDP

END MAIN

**Output:**



**Discussion:**

In this above code, firstly DATA Segment was used to show a prompt message and to declare an array to store the string. Then in the main program that data segment was initialized. After that, a FOR LOOP was used to store the characters into the array. Then CMP was used to compare the character whether it is lowercase or not. Then finally PRINT LOOP was used to show the output.

**Experiment Name**

Write a program that takes a hexadecimal number and in the output, it will print whether that hexadecimal number is Even or Odd.

**Theory**

As assembly language instructions are so basic, Input/output is much harder in assembly language than high level languages. DOS functions are used to take I/O. In assembly language there are operation field, operand field, variables and they work with different registers. There are CMP, JG, JL, JE and JMP instructions available in assembly language for Comparing to operand, Jump to a specific line if first variable is greater than the second, Jump to a specific line if first variable is lesser than the second, Jump to a specific line if first variable is equal to the second and to jump to a line if none of the conditions become true respectively. For input and output, MOV AH,1 and MOV AH,2 are used respectively in assembly language.

**Code:**

.MODEL SMALL

.STACK 100H

.DATA

PROMPT1 DB 'Type a 4 digit Hexadecimal number: ','$'

PROMPT2 DB 'This is an Even number $'

PROMPT3 DB 'This is an Odd number $'

A DB ?

B DB ?

C DB ?

D DB ?

.CODE

MAIN PROC

MOV AX, @DATA

MOV DS, AX

MOV AH,9

LEA DX,PROMPT1

INT 21H

MOV AH,1

INT 21H

MOV A,AL

MOV AH,1

INT 21H

MOV B,AL

MOV AH,1

INT 21H

MOV C,AL

MOV AH,1

INT 21H

MOV D,AL

MOV AH,2

MOV DL,0DH

INT 21H

MOV DL,0AH

INT 21H

EX6:

CMP D,30H

JE E

JMP EX7

EX7:

CMP D,31H

JE O

JMP EX8

EX8:

CMP D,32H

JE E

JMP EX9

EX9:

CMP D,33H

JE O

JMP EX10

EX10:

CMP D,34H

JE E

JMP EX11

EX11:

CMP D,35H

JE O

JMP EX12

EX12:

CMP D,36H

JE E

JMP EX13

EX13:

CMP D,37H

JE O

JMP EX14

EX14:

CMP D,38H

JE E

JMP EX15

EX15:

CMP D,39H

JE O

JMP EX16

EX16:

CMP D,41H

JE E

JMP EX17

EX17:

CMP D,42H

JE O

JMP EX18

EX18:

CMP D,43H

JE E

JMP EX19

EX19:

CMP D,44H

JE O

JMP EX20

EX20:

CMP D,45H

JE E

JMP EX21

EX21:

CMP D,46H

JE O

JMP EX6

E:

MOV AH,2

MOV DL,0DH

INT 21H

MOV DL,0AH

INT 21H

MOV AH,9

LEA DX,PROMPT2

INT 21H

JMP EXIT

O:

MOV AH,2

MOV DL,0DH

INT 21H

MOV DL,0AH

INT 21H

MOV AH,9

LEA DX,PROMPT3

INT 21H

JMP EXIT

EXIT:

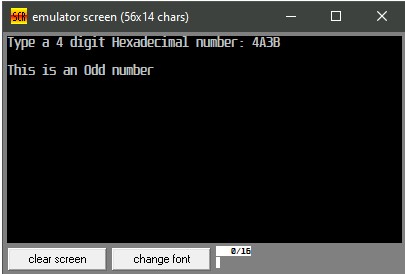
MOV AH,4CH ;this is for DOS exiting

INT 21H

MAIN ENDP

END MAIN

**Output:**



**Discussion:**

In this above code, firstly DATA Segment was used to show a prompt message and to declare 4 variables to store the digits of the hexadecimal number. Then in the main program that data segment was initialized. After that, the digits of the hexadecimal number were taken as input from the used and then the last digit of the number was checked by using several CMP operation which means comparing the last digit with 0 to 9, it was possible to identify a number as an even or odd number. After figuring out the number, a prompt message was printed regarding the number was even or odd.