MASM PROGRAMS

15. Display HelloWorld

AIM

Implement a program in MASM to display HelloWorld

```
.model small
.stack 64
.data
msg db "HELLO WORLD$"
.code
mov ax,@data
mov ds,ax
mov dx,offset msg
mov ah,09h
int 21h
mov ah,4ch
int 21h
end
```

OUTPUT

HELLO WORLD

Theory

The **AH** register is frequently used in conjunction with the **DOS interrupt 21h** to specify different system service functions.

Value in AH	Function (when used with `int 21h`)	
`09h`	Display a string (as in your example, used to print text)	→Associated with register DX
`4Ch`	Terminate the program	
`01h`	Read a character from standard input (keyboard)	→Associated with register AL
`02h`	Display a single character on the screen	→Associated with register DL

If you set **AH = 09h** and call **int 21h**, it will display a string terminated by a $ symbol

If you set **AH = 4Ch** and call **int 21h**, it will terminate your program and return control to DOS

16. Display the next character of a given character

AIM

Implement a program in MASM to display the next character of a given character

PROGRAM

```
.model small
.stack 64
.data
msg1 db "Enter the character:$"
msg2 db 0ah, "The Next character:$"                    //0Ah representing a new line

.code
mov ax,@data          //Loads the starting address of the data segment into the AX register
mov ds,ax

mov dx,offset msg1
mov ah,09h
int 21h

mov ah,01h
int 21h

add al,01h

mov dx,offset msg2
mov ah,09h
int 21h

mov dl,al
mov ah,02h
int 21h

mov ah,4ch
int 21h
end
```

OUTPUT

```
Enter the character: a
The Next character: b
```

17. Add two numbers

AIM

Implement a program in MASM to add two numbers

PROGRAM

```
.model small
.stack 64
.data
msg1 db "Enter number1 $ "
msg2 db 0ah, "Enter number2 $ "
msg3 db 0ah, "Sum is $ "
.code
mov ax,@data
mov ds,ax

mov dx,offset msg1
mov ah,09
int 21h

mov ah,01h
int 21h
sub al,30h
mov bl,al

mov dx,offset msg2
mov ah,09
int 21h

mov ah,01h
int 21h
sub al,30h
add bl,al
add bl,30h

mov dx,offset msg3
mov ah,09
int 21h

mov ah,02h
mov dl,bl
int 21h

mov ah,4ch
int 21h

end
```

```
sub al, 30h
```

- **Explanation**: Subtracts **30h** (the ASCII value of '0') from the value in **AL** to convert the ASCII character (representing a number) to its numerical equivalent.
    - For example, if **AL** contains '5' (ASCII 53 or 0x35), subtracting 0x30 results in the number 5.

OUTPUT

```
Enter number1 1
Enter number2 3
Sum is 4
```

18. Display two digit numbers

AIM

Implement a program in MASM to display two digit numbers

PROGRAM

```
.model small
.stack 64
.data
msg1 db "Enter digit1 $"
msg2 db 0ah, "Enter digit2 $"
msg3 db 0ah, "Number is $"
.code
mov ax,@data
mov ds,ax

mov dx,offset msg1              //Prompting for the first digit
mov ah,09h
int 21h

mov ah,01h
int 21h
sub al,30h

mov bl,0ah                      //Multiply the first digit by 10
mul bl

mov bl,al

mov dx,offset msg2              //Prompting for the second digit
mov ah,09h
int 21h

mov ah,01h
int 21h
sub al,30h

add al,bl                       //Add the two digits, AL holds the final two-digit number

mov bl,0ah
mov ah,00h
div bl

add al,30h
add ah,30h
mov bl,ah

mov dx,offset msg3
```

//Divides the number in **AL** by **10** to separate the tens and ones digits.

- After the division, **AL** contains the **ones digit** (remainder), and **AH** contains the **tens digit** (quotient).

```
mov ah,09h
int 21h

mov ah,02h
mov dl,al
int 21h

mov al,bl

mov ah,02h
mov dl,al
int 21h

mov ah,4ch
int 21h

end
```

OUTPUT

```
Enter digit1 2
Enter digit2 3
Number is  23
```

19. Calculator

AIM

Implement a calculator in MASM

PROGRAM

```
.model small
.stack 64
dispm macro msg
mov dx,offset msg
mov ah,09h
int 21h
endm

dispn macro al
mov dl,al
mov ah,02h
int 21h
endm

readno macro num
mov ah,01h
int 21h
sub al,30h
mov num,al
endm

.data
msg1 db 0ah,"Calculator 1.Sum 2.Product 3.Subtraction 4.Quotient 5.Exit$"
msg2 db 0ah,"Enter the choice:$"
msg3 db 0ah,"Enter the number(1):$"
msg4 db 0ah,"Enter the number(2):$"
msg5 db 0ah,"The sum is:$"
msg6 db 0ah,"The product is:$"
msg7 db 0ah,"The difference is:$"
msg8 db 0ah,"The quotient is:$"
num1 db ?
num2 db ?
ch1 db ?

.code
start:mov ax,@data
mov ds,ax
dispm msg1
dispm msg2
readno ch1
cmp ch1,05
jz exit
```

Theory

Macros:

- Macros are reusable blocks of code that can be invoked with a simple name. They help reduce code duplication. In this program, 3 macros defined.

1. dispm macro msg

- This macro displays a message by loading the address of the message into the **DX** register, setting **AH** to `09h`, and calling interrupt `int 21h`.

2. dispn macro al

- This macro displays a single character stored in **AL** by moving it to **DL**, setting **AH** to `02h` , and calling interrupt `int 21h`.

3. readno macro num

- This macro reads a single character input from the user, converts it from ASCII to a numeric value (by subtracting `30h`), and stores it in the variable `num`.

```
dispm msg3
readno num1
dispm msg4
readno num2
cmp ch1,01
jz addn
cmp ch1,02
jz pron
cmp ch1,03
jz subn
cmp ch1,04
jz divn

exit:mov ah,4ch
int 21h

addn:dispm msg5
mov al,num1
add al,num2
add al,30h
dispn al
jmp start

pron:dispm msg6
mov al,num1
mul num2
add al,30h
dispn al
jmp start

subn:dispm msg7
mov al,num1
sub al,num2
add al,30h
dispn al
jmp start

divn:dispm msg8
mov ah,00h
mov al,num1
div num2
add al,30h
dispn al
jmp start

end
```

OUTPUT

```
Calculator 1.Sum 2.Product 3.Subtraction 4.Quotient 5.Exit"
```

```
Enter the choice: 1
Enter the number(1): 3
Enter the number(2): 4
    The sum is: 7
```