1. **Write an assembly language program to perform multiplication of 8-bit data.**

**Code**

.model small

.stack 100h

.data

num1 db 6h

num2 db 2h

result dw 0

msg db 'Your final Product is: $'

.code

main proc

mov ax, @data

mov ds, ax

mov al, num1

mov bl, num2

xor cx, cx

xor dx, dx

multiply:

test bl, 1

jz skip\_add

add cx, ax

skip\_add:

shl ax, 1

shr bl, 1

inc dx

cmp dx, 8

jl multiply

mov result, cx

lea dx, msg

mov ah, 09h

int 21h

mov ax, result

call DisplayResult

mov ah, 4Ch

int 21h

main endp

DisplayResult proc

mov bx, 10

xor cx, cx

convert\_digit:

xor dx, dx

div bx

add dl, '0'

push dx

inc cx

test ax, ax

jnz convert\_digit

display\_digit:

pop dx

mov ah, 02h

int 21h

loop display\_digit

ret

DisplayResult endp

end main  
  
**OUTPUT**  
A screenshot of a computer

Description automatically generated

1. **Write a program in assembly language to perform multiplication of 16-bit data.**

**CODE**

.model small

.stack 100h

.data

num1 dw 1234h

num2 dw 5678h

result dw 0

result\_high dw 0

msg db 'The product is: $'

.code

main proc

mov ax, @data

mov ds, ax

mov ax, num1

mov cx, num2

mul cx

mov result, ax

mov result\_high, dx

lea dx, msg

mov ah, 09h

int 21h

mov ax, result

call DisplayNumber

mov ax, result\_high

call DisplayNumber

mov ah, 4Ch

int 21h

main endp

DisplayNumber proc

mov bx, 10

xor cx, cx

convert\_digit:

xor dx, dx

div bx

add dl, '0'

push dx

inc cx

test ax, ax

jnz convert\_digit

display\_digit:

pop dx

mov ah, 02h

int 21h

loop display\_digit

ret

DisplayNumber endp

end main

**OUTPUT**

A screenshot of a computer

Description automatically generated