Code: -

```
data Segment
    msg db 0dh, 0ah, "Enter a 16-bit number: $"
    result db 0dh, 0ah, "The Result is: $"
    newl db 0dh, 0ah, " $"
    menu db 0dh, 0ah, "1: add", 0dh, 0ah, "2: sub", 0dh, 0ah, "3: mul", 0dh, 0ah, "4: div", 0dh, 0ah, "$"
data ends

code Segment
assume cs:code, ds:data
    Start:
        mov ax, data
        mov ds, ax

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

        call AcceptNum
        mov bh, bl
        call AcceptNum
        mov cx, bx

        mov dx, offset msg
        mov ah, 09h
        int 21h
```

```
    call AcceptNum
    mov bh, bl
    call AcceptNum

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

    mov ah, 01h
    int 21h

    cmp al, '1'
    jz Addition
    cmp al, '2'
    jz Subtraction
    cmp al, '3'
    jz Multiplication
    cmp al, '4'
    jz Division

Addition:
    add cx, bx
    jmp EndSwitch

Subtraction:
    sub cx, bx
```

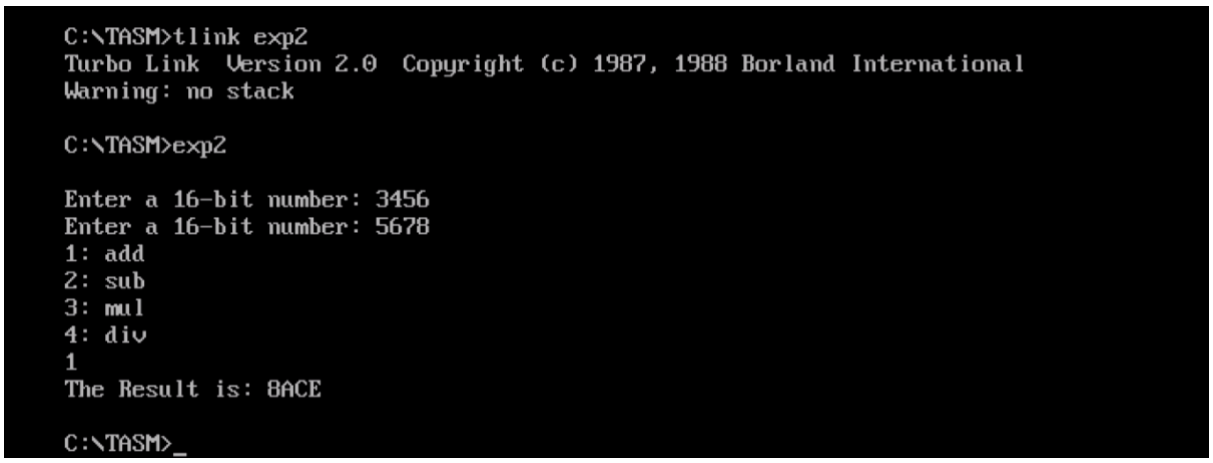
```
    jmp EndSwitch

Multiplication:
    mov ax, cx
    mul bl
    mov cx, ax
    jmp EndSwitch

Division:
    mov ah, 0
    mov al, cl
    div bl
    mov ch, 0
    mov cl, al
    jmp EndSwitch

EndSwitch:

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

    mov bl, ch
    call DispNum

    mov bl, cl
    call DispNum
```

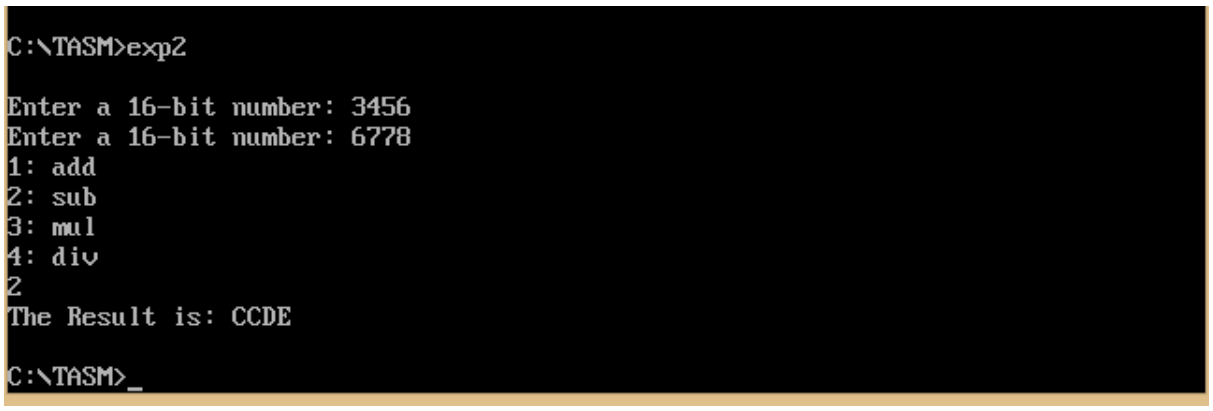
```
    mov dx, offset newl
    mov ah, 09h
    int 21h

    mov ah, 4ch
    int 21h

AcceptNum proc
    mov ah, 01h
    int 21h

    call HexAccept

    mov bl, al
    rol bl, 4

    mov ah, 01h
    int 21h

    call HexAccept
    add bl, al
    ret
endp

DispNum proc
    mov al, bl
```

```
    and al, 0f0h
    ror al, 4

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

    mov al, bl
    and al, 0fh

    mov dl, al
    call HexDisp
    mov ah, 02h
    int 21h
endp

HexAccept proc    ; Compare to 41 if it is less than A then we need to sub only 30
  ; If it is greater than or equal to 41 then we also need to sub 07
    cmp al, 41h
    jc norm
    sub al, 07h
    norm: sub al, 30h
    ret
endp

HexDisp proc     ; Compare to 0a if it is less than A then we need to add only 30
```

```
    ; If it is greater than or equal to 0a then we also need to add 07
      cmp dl, 0ah
      jc nothex
      add dl, 07h
      nothex: add dl, 30h
      ret
   endp
   end Start
Code ends
```

OUTPUT: -





```
C:\TASM>exp2

Enter a 16-bit number: 3478
Enter a 16-bit number: 6790
1: add
2: sub
3: mul
4: div
3
The Result is: 4380

C:\TASM>
```

```
C:\TASM>exp2

Enter a 16-bit number: 6768
Enter a 16-bit number: 9324
1: add
2: sub
3: mul
4: div
4
The Result is: 0002

C:\TASM>_
```