

;Name : Omar mohsen emam

; Id : 2017015

org 100h

jmp start

; messages block

msg0: db 0dh,0ah,"Calculator \$"

msg: db 0dh,0ah,"Press 'A' For Addition",0dh,0ah,"Press 'S' For Subtraction",0dh,0ah,"Press 'M' For Multiplication",0dh,0ah,"Press 'D' For Division",0dh,0ah,"Press 'E' For Exit",0dh,0ah,"Press 'R' For Return to Main

Menu",0dh,0ah,"*****",0dh,0ah,"*****"
*****",0dh,0ah,"Enter Your Choice : ",0dh,0ah,"\$"

msg2: db 0dh,0ah," Enter First Number : ",0dh,0ah," \$"

msg3: db 0dh,0ah," Enter Second Number : ",0dh,0ah," \$"

msg4: db 0dh,0ah," Wrong Choice Press any key.... ",0dh,0ah," \$"

msg5: db 0dh,0ah," Answer : ",0dh,0ah," \$"

msg6: db 0dh,0ah,"Thank You, Press 'R' to return to main menu",0dh,0ah,"OR",0dh,0ah,"Press any key to exit.....",0dh,0ah," \$"

msg7: db 0dh,0ah,"Note - Reminder Will not be Showed \$"

msg8: db 0dh,0ah,"For Addition \$"

msg9: db 0dh,0ah,"For Subtraction \$"

msg10: db 0dh,0ah,"For Multiplication \$"

msg11: db 0dh,0ah,"For division \$"

msg12: db 0dh,0ah,"Thank You \$"

;Main function

start: mov ah,9

mov dx, offset msg0

```
int 21h
mov ah,9
mov dx, offset msg
int 21h
mov ah,0
int 16h
call choice
;addition
cmp al,065
je Addition ;jump if equal (jump flag equal 1)
cmp al,097
je Addition
;multiply
cmp al,109
je Multiply
cmp al,077
je Multiply
;Subtract
cmp al,115
je Subtract
cmp al,083
je Subtract
;divide
cmp al,068
je Divide
cmp al,100
je Divide
;return
cmp al,114
```

```
je start
cmp al,082
je start
;exit
cmp al,101
je exit
cmp al,069
je exit
;error
mov ah,9
mov dx, offset msg4
int 21h
mov ah,0
int 16h
jmp start
```

Addition: mov ah,9

```
mov dx, offset msg8
int 21h
mov ah,9
mov dx, offset msg2
int 21h
mov cx,0
call input
push dx
mov ah,9
mov dx, offset msg3
int 21h
mov cx,0
```

```
call input
pop bx
add dx,bx
push dx
mov ah,9
mov dx,offset msg5
int 21h
pop dx
mov cx,10000 ;maximum number this calc can calculate
call Viewno
jmp exit
```

```
exit:  mov ah,9
       mov dx, offset msg12
       int 21h
       ret
```

```
Viewno: mov ax,dx          ;show numbers on screen
        mov dx,0
        div cx
        call view
        mov bx,dx ;(value of reminder)
        mov dx,0
        mov ax,cx
        mov cx,10
        div cx
        mov dx,bx
        mov cx,ax
        cmp ax,0
```

```
jne viewno
mov dx,offset msg6
    mov ah,9
    int 21h
    mov ah,0
    int 16h
cmp al,114
je start
cmp al,082
je start
ret
```

```
input: mov ah,0
    int 16h
    mov dx,0 ;reg to add values in each iteration
    mov bx,1 ;initial value [ex 2 = 2*1+0]
    cmp al,0dh ;this line to make sure user finished typing his no in multi digits(compare al to ascii value
of enter)
    je Form
    sub ax,30h ;change input for ascii to decimal
    call view
    mov ah,0
    push ax ;forming a loop so input screen doesnt end untill user finish input number
    inc cx
    jmp input
```

```
Form: pop ax
```

```

push dx
mul bx
pop dx
add dx,ax
mov ax,bx
mov bx,10
push dx
mul bx
pop dx
mov bx,ax
dec cx
cmp cx,0
jne Form    ; jumpflag =0  (not equal)
ret

```

view: push ax ; move ax, dx to stack

```
push dx
```

```
mov dx,ax    ;move input nu to dx to be viewed on screen
```

add dl,30h ;convert nu back to ascii to be showed in screen (on ax there still the input number we will do arithmetics)

```
mov ah,2    ; when viewing a text msg we move ah to 9 but when viewing a num w move to 2
```

```
int 21h
```

```
pop dx
```

```
pop ax    ; return dx,ax [first number to push must be last number to pop
```

```
ret
```

Multiply: mov ah,9

```
mov dx, offset msg10
```

```
int 21h
```

```
mov ah,9
mov dx, offset msg2
int 21h
mov cx,0
call input
push dx
mov ah,9
mov dx, offset msg3
int 21h
mov cx,0
call input
pop bx
mov ax,dx ;move first number to ax since mul instruction always expect to store value in ax
mul bx ;multiply value of bx to ax
mov dx,ax ; return Result of multiplication to dx
push dx
mov ah,9
mov dx,offset msg5
int 21h
pop dx
mov cx,10000 ;maximum number this calc can calculate
call Viewno
jmp exit
```

Subtract: mov ah,9

```
mov dx, offset msg9
int 21h
mov ah,9
mov dx, offset msg2
```

```
int 21h
mov cx,0
call input
push dx
mov ah,9
mov dx, offset msg3
int 21h
mov cx,0
call input
pop bx
sub bx,dx ;cange parameters here so second number from first number
mov dx,bx ;move back result to be shown
push dx
mov ah,9
mov dx,offset msg5
int 21h
pop dx
mov cx,10000 ;maximum number this calc can calculate
call Viewno
jmp exit
```

Divide: mov ah,9

```
mov dx, offset msg11
int 21h
mov ah,9
mov dx, offset msg7
int 21h
mov ah,9
mov dx, offset msg2
```



```
int 21h
mov cx,0
call input
push dx
mov ah,9
mov dx, offset msg3
```

```
int 21h
mov cx,0
call input
pop bx
```

mov ax,bx ; coffecent will be stored in ax while bx will store the reminder Although reminder will not be shown in result

```
mov cx,dx
mov dx,0
div cx
mov dx,ax
push dx
mov ah,9
mov dx,offset msg5
int 21h
pop dx
mov cx,10000 ;maximum number this calc can calculate
call Viewno
jmp exit
```

choice: push ax

```
push dx
mov dx,ax
mov ah,2
```

int 21h

pop dx

pop ax

ret

ret