**Start:**

org 100h

jmp start ; jump over data declaration

msg: db "1-Add",0dh,0ah,"2-Multiply",0dh,0ah,"3-Subtract",0dh,0ah,"4-Divide",0dh,0ah,"5-Exit", 0Dh,0Ah, '$‘

msg2: db 0dh,0ah,"Enter First No: $"

msg3: db 0dh,0ah,"Enter Second No: $”

msg4: db 0dh,0ah,"Choice Error $"

msg5: db 0dh,0ah,"Result: $“

msg6: db 0dh,0ah,'Thank you for using the simple calculator! Press any key to exit...', 0Dh,0Ah, '$‘

**Start:**

**main\_loop:**

mov ah,9

mov dx, offset msg ; Display the first message from which the user can choose the operation using int 21h

int 21h

mov ah,0

int 16h ; Use int 16h to read a key press, to know the operation chosen

cmp al,31h ; Compare to '1' for addition

je Addition

cmp al,32h ; Compare to '2' for multiplication

je Multiply

cmp al,33h ; Compare to '3' for subtraction

je Subtract

cmp al,34h ; Compare to '4' for division

je Divide

cmp al,35h ; Compare to '5' for exit

je Exit

mov ah,09h

mov dx, offset msg4 ; Display choice error message

int 21h

call next\_line

jmp main\_loop ; Repeat the main loop if invalid choice

**InputNo:**

mov ah,0

int 16h ; Read a key press

mov dx,0

mov bx,1

cmp al,0dh ; Check if Enter key was pressed

je FormNo ; If Enter key pressed, form the number

sub ax,30h ; Convert from ASCII to decimal

call ViewNo ; Display the key pressed

mov ah,0 ; Reset ah

push ax ; Push the digit onto the stack

inc cx ; Increment digit counter

jmp InputNo ; Repeat for next digit

**FormNo:**

pop ax ; Take the last input from the stack

push dx

mul bx ; Multiply ax by bx

pop dx ; Remove from stack

add dx,ax ; Add dx and ax

mov ax,bx ; Move bx to ax

mov bx,10

push dx ; Push dx to stack before multiplying

mul bx ; Multiply bx by 10

pop dx ; Pop dx after multiplying

mov bx,ax ; Move result from ax to bx

dec cx ; Decrement digit counter

cmp cx,0 ; Check if counter is 0

jne FormNo ; If not, repeat

ret ; Return if all digits processed

**View:**

mov ax,dx

mov dx,0

div cx

call ViewNo

mov bx,dx

mov dx,0

mov ax,cx

mov cx,10

div cx

mov dx,bx

mov cx,ax

cmp ax,0

jne View

ret

**ViewNo:**

push ax ; Save ax

push dx ; Save dx

mov dx,ax ; Move value to dx

add dl,30h ; Convert to ASCII

mov ah,2

int 21h

pop dx

pop ax

ret

**Addition:**

mov ah,9 ; Handle addition operation

mov dx, offset msg2 ; Display message "Enter first number"

int 21h

mov cx,0 ; Reset counter

call InputNo ; Get the first number

push dx

mov ah,9

mov dx, offset msg3 ; Display message "Enter second number"

int 21h

mov cx,0

call InputNo ; Get the second number

pop bx

add dx,bx ; Add the two numbers

push dx

mov ah,9

mov dx, offset msg5 ; Display result

int 21h

mov cx,10000

pop dx

call View

call next\_line

jmp main\_loop

**Subtract:**

mov ah,9

mov dx, offset msg2

int 21h

mov cx,0

call InputNo

push dx

mov ah,9

mov dx, offset msg3

int 21h

mov cx,0

call InputNo

pop bx

sub bx,dx

mov dx,bx

push dx

mov ah,9

mov dx, offset msg5

int 21h

mov cx,10000

pop dx

call View

call next\_line

jmp main\_loop

**Multiply:**

mov ah,9

mov dx, offset msg2

int 21h

mov cx,0

call InputNo

push dx

mov ah,9

mov dx, offset msg3

int 21h

mov cx,0

call InputNo

pop bx

mov ax,dx

mul bx

mov dx,ax

push dx

mov ah,9

mov dx, offset msg5

int 21h

mov cx,10000

pop dx

call View

call next\_line

jmp main\_loop

**Divide:**

mov ah,9

mov dx, offset msg2

int 21h

mov cx,0

call InputNo

push dx

mov ah,9

mov dx, offset msg3

int 21h

mov cx,0

call InputNo

pop bx

mov ax,bx

mov cx,dx

mov dx,0

mov bx,0

div cx

mov bx,dx

mov dx,ax

push bx

push dx

mov ah,9

mov dx, offset msg5

int 21h

mov cx,10000

pop dx

call View

pop bx

call next\_line

cmp bx,0

je main\_loop

jmp main\_loop

**next\_line:**

mov ah, 2

mov dl, 0Dh ; Carriage return

int 21h

mov dl, 0Ah ; Line feed

int 21h

ret

**Exit:**

mov dx,offset msg6

mov ah, 9

int 21h

mov ah, 0

int 16h

ret