

ax register (accumulator register)

ah, al \Rightarrow input receive

counter bx \Rightarrow bh, bl

cx \Rightarrow ch, cl

dx \Rightarrow dh, dl

c/b register use & lower portion use too

output to the receive

Output newline & Print error

input & the first two are output to the code, first two

mov ah, 2 \Rightarrow character output first next line first

mov dl, 10 \Rightarrow dl & the output after the newline

int 21h \Rightarrow ascii value 10. function call

1 \Rightarrow input

1 \Rightarrow output to cause carriage return in the print screen

so carriage return must be 10. Carriage return is 13

mov ah, 2

mov dl, 10

int 21h

mov dl, 13

int 21h

\Rightarrow 10 13

5.10.10

2.10.10

1.10.10

5.10.10

2.10.10

1.10.10

5.10.10

2.10.10

1.10.10

Variable declaration,

- model
- stack
- data

small
looh

data segment & data segment & data segment
Variable declare karo hai

a db 5 ⇒ a store variable & 5 value,
db ⇒ define byte for small value (1 byte)
dw ⇒ define word for big value (2 byte)

b db ?

- Code

main proc

mov ax, 0

mov ds, ax

mov ah, 1

int 21h

mov b, al

mov ah, 2

mov dl, a

int 21h

mov ah, 2

mov dl, b

int 21h

exit

mov ah, 4Ch

int 21h

end main

ax register & data segment address.

data segment & code segment & connection.

a & value store
ds register & address of data segment active karo hai.

b input karo.

⇒ output & ASCII value print karo.

add ~~code~~ (5+48) ~~code~~ 5 Print ~~code~~ 53(ASCII)=5.

```
mov ah, 2
mov dl, a
add dl, 48 ⇒ 48 মোডা করে দিতে।
int 21h
```

String print ⇒ String কিসে রাখার জন্য
dollar sign use করতে হবে।

lea ⇒ load effective address.

- model small
- stack loah
- data

a: db 'Bangla Baba \$'

- Code

main Proc

```
mov ax, @data
mov ds, ax
```

> data segment to code segment
initialize করুন।

```
mov ah, 9
lea dx, a ⇒ 48 মোডা করে
int 21h
```

exit:

```
mov ah, 4ch
int 21h
main endp
```

end main

$\frac{0.99 \text{ mol } \text{Ba}(\text{OH})_2}{(2+8)} = 0.26$

The number

I can also take 3 inputs and print them.

Exchange value

$\text{exchange } b_k, b_n \Rightarrow \text{exchange to exchange two values}$

main PROC book = 991

mov ah, 1

int 21h

mov bl, al

mov ah, 1

21h

may
bh, al

Box 49, 2a

mod ab, 2

7a, b

inf 2lh

mov ah, 2

Now q, p

int 21 h

mov ah, 2

mov dl, bh

inf 21h

Beep sound: print = 07 => ASCII value.

↳ dl is start of string print out extra beep sound zt0 |

Add two numbers

```
mov ah,1
int 21h
mov bl,al

mov ah,1
int 21h
mov bh,al
```

add bl,bh => that means $bl = bl + bh \geq 20$ number add error
bl is start of string | bl print error
addition or print error

```
mov ah,2
mov dl,bl
int 21h.
```

it will show ASCII value.

But if I want to print the decimal value than I need to subtract 48

```
mov ah,2
mov dl,bl
sub dl,48
int 21h
```

Addition of three numbers

```
add bl, bh
add bl, cl
```

```
bl = bl + bh
bl = bl + cl
```

Modifying out bbl

Subtraction of two numbers

main proc

```
mov ah, 1
```

```
int 21h
```

```
mov bl, al
```

```
mov ah, 1
```

```
int 21h
```

```
mov bh, al
```

```
sub bl, bh ; bl = bl - bh
```

```
add bl, 48
```

```
mov ah, 2
```

```
mov dl, bl
```

```
int 21h
```

Subtraction of 2
Sub to result

to 48 add zero

Addition of 2 add to
to result to store of
Sub zero

5 10 100
10 10 100
10 10 100
10 10 100

Lower to upper case

↳ Subtract 32.

upper to lower \Rightarrow add 32.

Convert Hexadecimal to decimal

A=10, B=11, C=12, D=13, E=14, F=15
जहाँ 1 by default Print करेगा A को 0 को बाद 65 को 108 को मिलेगा
10 बाद 10 Print करेगा।

mov ah, 1
int 21h
mov bl, al
sub bl, 17 \Rightarrow 17 बिना करेगा।

mov ah, 2
mov dl, 10
int 21h
mov dl, 13
int 21h

mov ah, 2
mov dl, 10
int 21h \Rightarrow 1 by default Print
 \Rightarrow 10 ASCII Value.

mov ah, 2
mov dl, bl
int 21h

Multiply With Static Initialization

mov al, 3 → al નો મૂલ્ય 3

mov bl, 2 → bl નો મૂલ્ય 2

mul bl → $al = al * bl$

mov ah, 2

mov dl, al

add dl, 48

int 21h

→ to print decimal value.