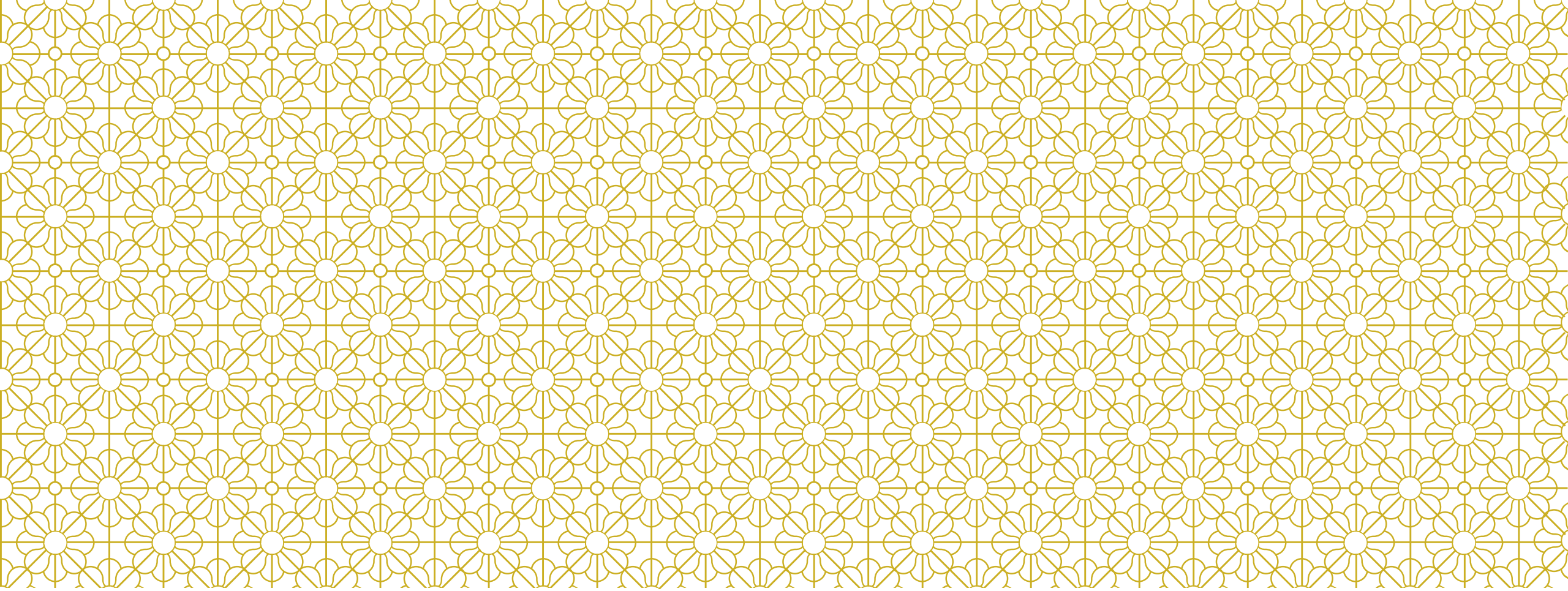


LOOP,LABEL,COUNTER REGISTER,INC

Loop, Label, Counter Register, Inc.

Loop "Series of conditions that is repeated until a terminating condition is reached."

```
Mov dx, 'a'
Mov ah, 2
int 21h
```

Basic instructions for prnig single character. To repeat it again and again?

. Give name of series of instuction, In beginning, called it label.

LabelName: ⟶ Give name Series of instruction In beginning, we called it label

```
Mov dx, 'a'
Mov ah, 2
int 21h
```

} Instruction.

Loop LabelName ⟶ After Series of instruction where it is called give its name after write Loop.

- LabelName should be any name.
- To call it how many times, it will be decided by Counter Register.

Counter Register

- General Purpose Register.
- Count program
- Main purpose is to be used for a loop.

- The value placed in counter register, (a constant value) it will run according to it.

- [illegible]

- Before writing the loop we send value to Counter Register

Mov cx, 10 (runs 10 times)

- Value reduced from $Cx = 10$ to $Cx = 0$ loop stop. (Work on decremant by 1 till 0).

$Cx = 10$
$Cx = 9$
⋮
$Cx = 0$.

- **Label Syntax**

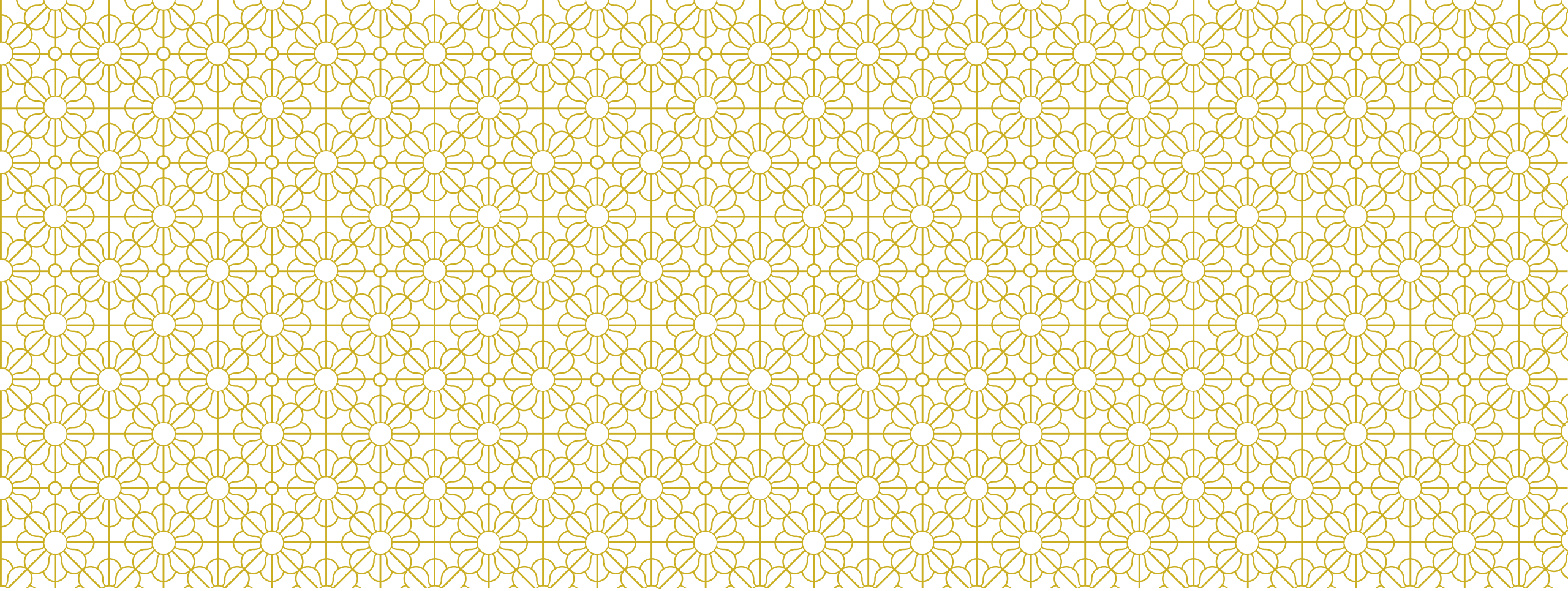
✓ Test:
✓ Test1:
✓ T1:
✗ 1Test:
✗ Reserved word:

Label Rules

1) A label can be placed at the beginning of a statement, because the label is assigned the current value of line.

2) Label Name must not be a reserved word.
   e.g. Mov, Add, DB, DW, etc.

3) Colon : must be used with label while initializing, but not while calling.



LAB

Program to print 0 to 9.


```
.dosseg
.model small
.stack 100h.
.data
.code.
main proc
```

Mov cx, 10 ⟶ first Counter is given for 10 times (0 to 9).

(Then give label name),

```
L1:
    Mov dx,48
    Mov ah,2
    int 21h

Loop L1
```

In this case 0 print again and again. dx again h again get 48. So we placed it before loop Value go only one time.

So we change it.

```
Mov cx,10
mov dx,48
L1:
   mov ah,2
   int 21h
     Add dx,1
Loop L1
```

In this also zero print 1 time, In loop after print write [Add dx,1], 48 add 1, so on loop runs.

- We used,

```
Add dx, 1
```

OR

```
Inc dx
```

→ Increment by 1

```
.dosseg
.model small
.stack 100h
.data
.code
main proc
mov cx, 10
mov dx, 48
```

```
L1:
mov ah, 2
int 21h
Add dx, 1
Loop L1
mov ah, 4ch
int 21h
main endp
end main
```

(2) Program to print Capital letters from A to Z using loop.

- tell counter register how many times the loop run, (A to Z) 26 characters,
  Mov cx, 26.
- Give loop name.

```
Main proc
Mov cx, 26      ⟶ loop run 26 times.
mov dx, 65      ⟶ 65 ASCII for A.
L1:
  mov ah, 2
  int 21h
```

```
inc dx      or   add dx, 1
Loop L1
mov ah, 4ch
int 21h
main endp
end main
```

(3) Program to print Small letters from a to z using loop?

AND TRUTH TABLE

Input		Output
A	B	Y = A.B
0	0	0
0	1	0
1	0	0
1	1	1

AND OPERATION

```
.model small
.stack 100h
.data
.code
main proc
MOV BL, 101B
AND BL, 110B
ADD BL, 48
MOV DL, BL
MOV AH, 2
INT 21H
MOV AH, 4CH
INT 21H
MAIN ENDP
END MAIN
```

```
101
110
---
100
```

AND operation

Result in
Binary 100
Decimal 4

OR TRUTH TABLE

Input		Output
A	B	Y = A+B
0	0	0
0	1	1
1	0	1
1	1	1

OR Operation

```
.model small
.stack 100h
.data
.code
main proc
MOV BL,101B
OR BL,110B
ADD BL,48
MOV DL,BL
MOV AH,2
INT 21H
MOV AH,4CH
```

```
 101
 110
-----
 111
-----
```

Result in
Binary 111
Decimal 7

XOR TRUTH TABLE

Input		Output
A	B	Y = A⊕B
0	0	0
0	1	1
1	0	1
1	1	0

XOR OPERATION

```
.model small
.stack 100h
.data
.code
main proc
MOV BL, 101B
XOR BL, 110B
ADD BL, 48
MOV DL, BL
MOV AH, 2
INT 21H
MOV AH, 4CH
INT 21H
```

```
  101
  110
-----
  011
-----
```

Result in
Binary 011
Decimal 3