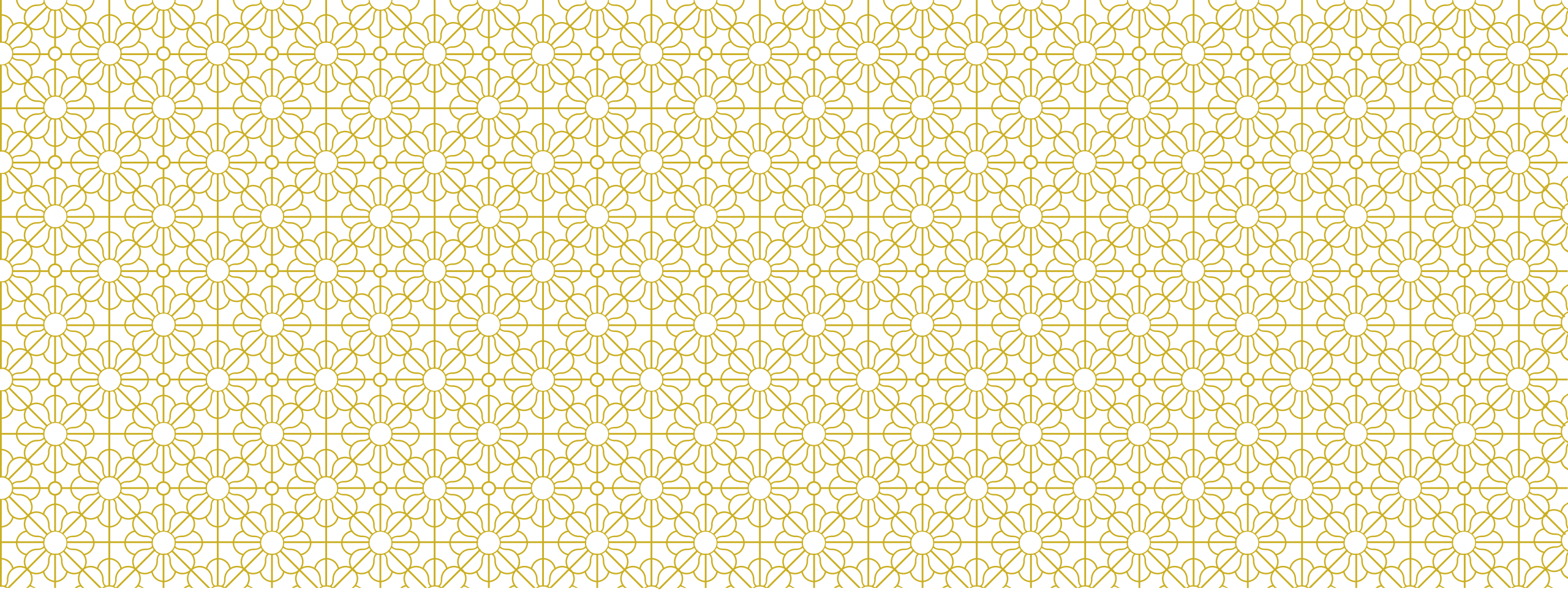
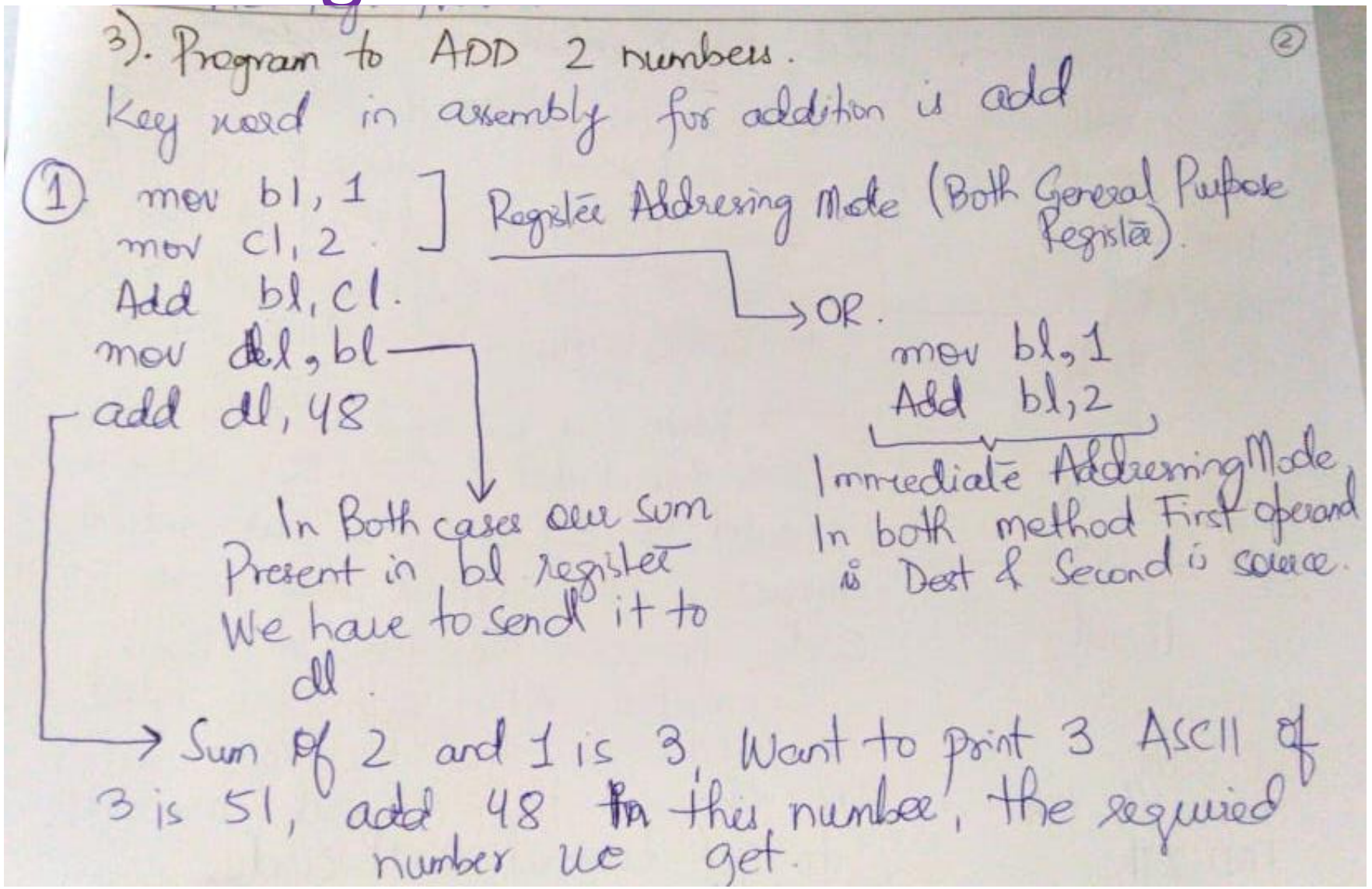


LAB 2

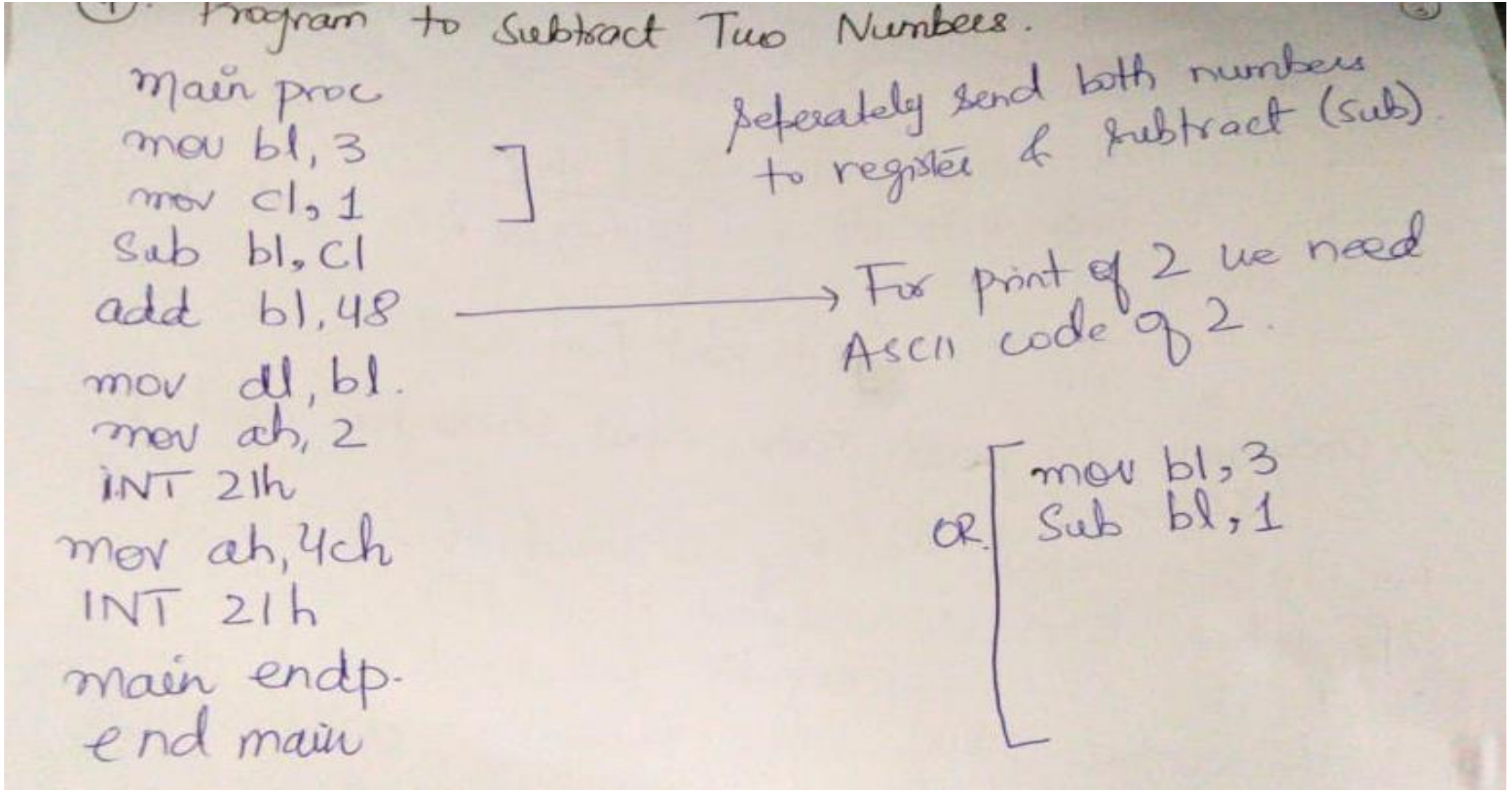
BSSE

1. Program to ADD two Numbers

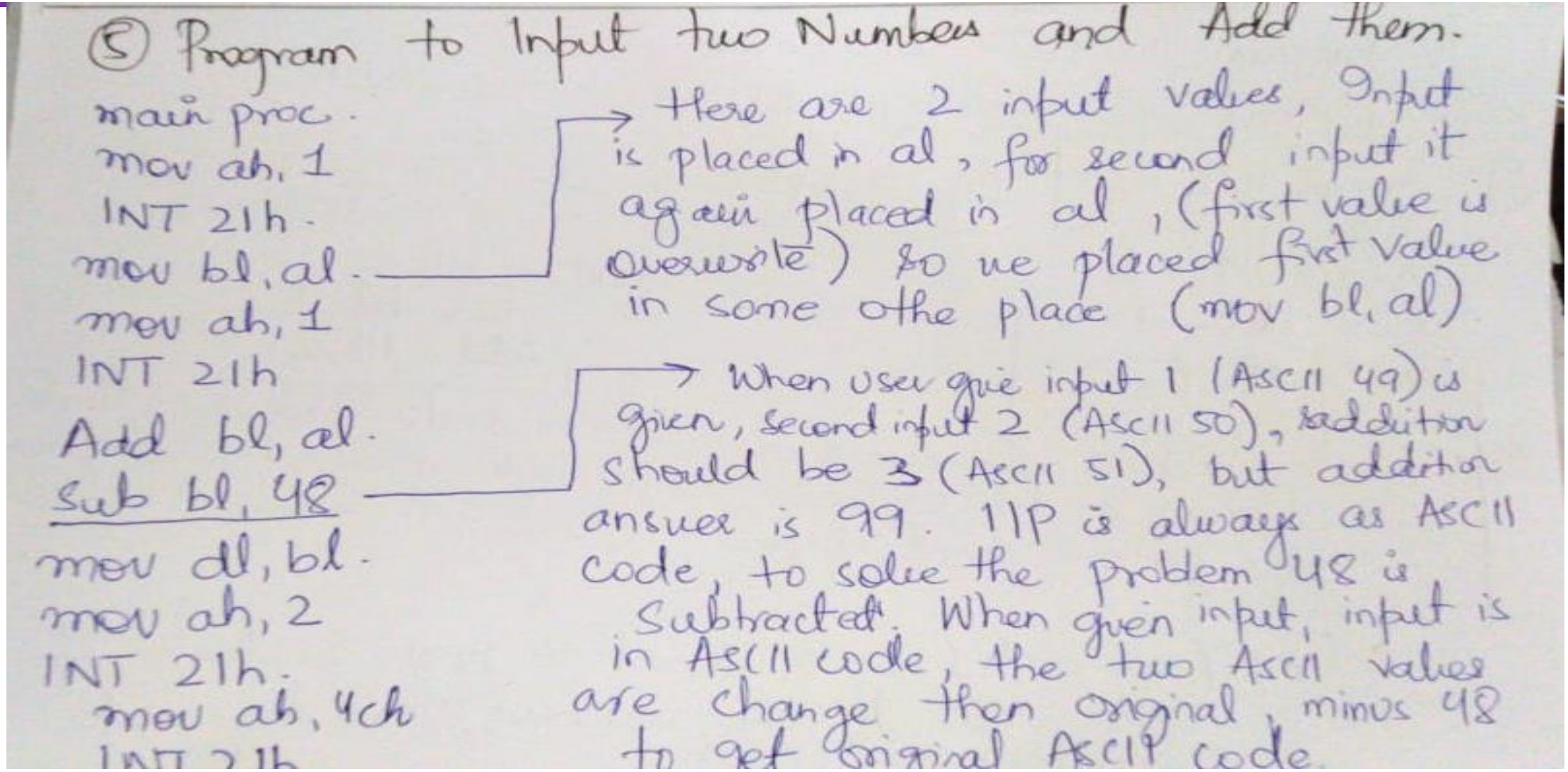



```
.model small
.stack 100h
.data
.code

main proc

mov bl,2
mov dl,1
add dl,bl
add dl,48
mov ah,2
INT 21h


mov ah,4ch
INT 21h

main endp
```

2. Program to Subtract Two Numbers



3. Program to input two numbers and Add them

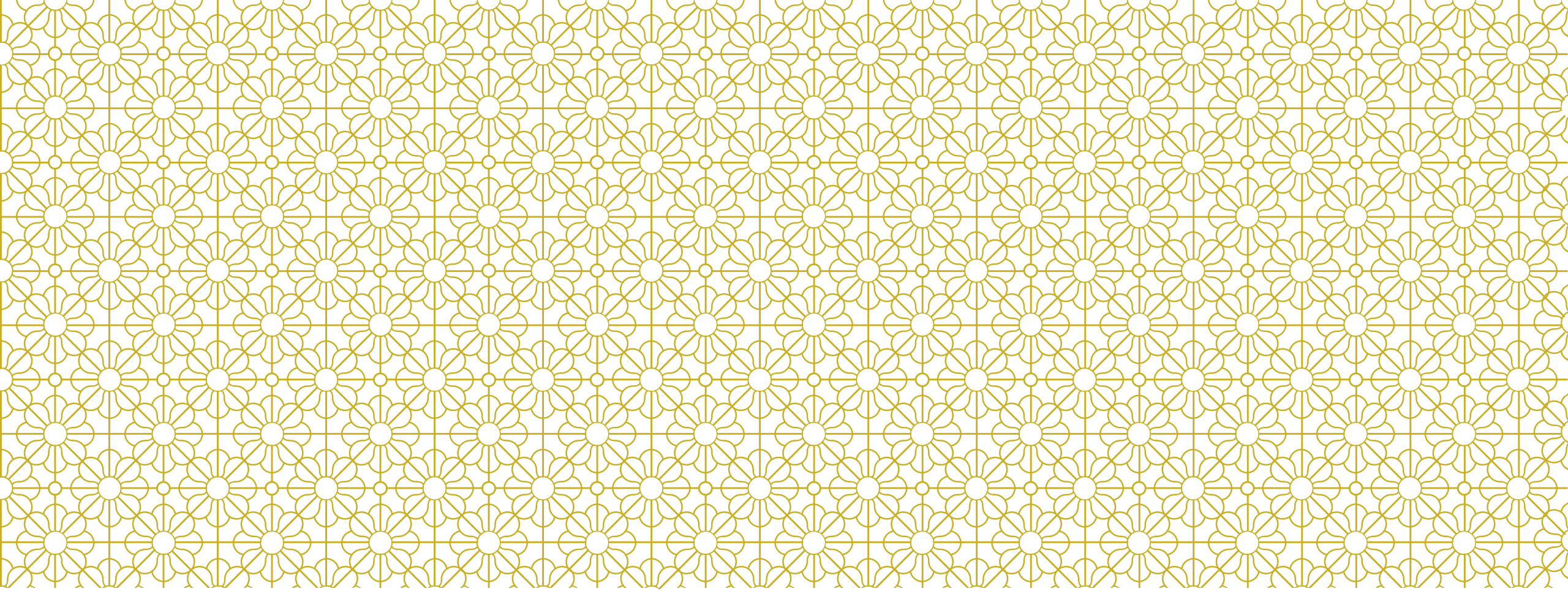


⑤ Program to Input two Numbers and Add them.

```
main proc.
mov ah, 1
INT 21h.
mov bl, al.
mov ah, 1
INT 21h
Add bl, al.
Sub bl, 48
mov dl, bl.
mov ah, 2
INT 21h.
  mov ah, 4ch
```

Here are 2 input values, 9nput is placed in al, for second input it again placed in al, (first value is overwrite) so we placed first value in some othe place (mov bl, al)

When user give input 1 (ASCII 49) is given, second input 2 (ASCII 50), addition should be 3 (ASCII 51), but addition answer is 99. I/P is always as ASCII code, to solve the problem 48 is Subtracted. When given input, input is in ASCII code, the two ASCII values are change then original, minus 48 to get original ASCII code.



FLAG REGISTER

Flag Register

- It's a flag register, It has several bits, every bit has different function and names.
- Basically a flag register, which has different bit names

Why flag Register :-

1) We know about what controls the operations of CPU ? like int 21h ⟶ how CPU handle this
2). What handles the status of operations.

```
  10101010
+ 01010101
----------
```

Bit move to register & add performed, what about carry while addition, where carry is moved.

How flag Reg do this work in add, sub, div.

- What bit handles the status of operation.

3) Conditional jump.

```
mov dl, 12
mov al, 10
```

If i want comparison b/w dl and al (If dl is greater then al) then it go up.

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

- Use of conditional jump with the help of flag register.

4) Which number is lessee or greater ?

- Flag Register is a register that contain the current state of the processor.

In CPU flag register 16 bits (0_15).

Useful bits is 9.

Any bit of register like 64 or 128 useful is 9.

1) Carry flag :- (CF) : Addition of 2 bits, last final carry out is Handled by this Register.

last carry out → 1 1 0 1 1
+ 1 0 1 1
1 0 1 0

Carry flag : CF
1 : When thee is last cary out
0 : When there is not last carry out.

2) Parity flag :- Important flag. (PF)

- Tells about integrity of data.
- Result of Add, Sub any operation, when this result is stored in register, go through some medium (wire/bus). When it reaches, the data is correct?, who tells about validity & verified it.

An additional extra bit is add in result to tell about its validity.

On 8086 arch Parity Flag : PF

1 : When there is even no. of ones.
0 : When there is not even no. of ones.

0 (00000010) → odd. (no. of bits).

3). Auxillary Flag (AF) In addition, the carry (not last carry) every 3rd bit carry.

Every 3rd bit carry

```
 1 0 1 0 1 0 1 0
+1 1 0 1 0 1 0 0
```

Every 3rd bit carry.

Every 3rd bit carry controlled by Auxillary flag.

1: When 3rd bit carry exists
0: When 3rd bit carry doesn't exist.

4) Zero flag: (ZF)

Perform subtraction of two numbers like.

```
  00000001  <-- 1
- 00000001  <-- 1
----------
  00000000  <-- 0
```

Result of operation is zero. handled by ZF

1: When result is zero.
0: When result is not zero.

(5) Sign Flag (SF)

Subtract of two numbers like result is in minus,

```
  00000011   3
- 00000111   7
----------
  00000110  -4
```

Result is in minus, but not visible in binary, this minus is handled by Sign flag.

SF : 1 ⟶ When result is negative
0 ⟶ When result is positive. (handles sign in result).

⑥ Trap flag. Error trap. (Bugs trap.

When prog run it show errors, how it is show, How CPU show you the error, Shown by TF.

. System used this for Debugging.

1 : When single step mode (debugging) is required/needed
0 : When single step mode (debugging) is not needed.

Used by system by default at Backend.

⑦ Interupt Flag (IF) Interupt handled by IF
tells CPU that interupt is called.

IF : 1 ⟶ when interupt is called
0 ⟶ when interupt is not called.

⑧ Direction flag (DF): 'hello$' Any string, string print from h to 0, normally, If you want to print in reverse order, direction handled by DF.

DF : 1 : String automatically decrements the address.
0 : string does not automatically decrement the address.

'hello$'
⟵

⑨ Overflow flag(OF) If size is register is less then result, handled by OF.

Add dx, ax.

ax |||||||||||||||| max range 65536
dx |||||||||||||||| size 65536

Addition of ax and dx, become out of range.

1: When result is too big to fit in the Destination
0: When ~~result~~ there is not too big to fit in the destination.

Status flag: To handle the result of an operation.
CF, PF, AF, ZF, SF

Control flag: To control operation of CPU
TF, DF, IF