NAME : KESHAV GAUR
ROLL NO. : AC-1223

ADDITION :

- **1 digit**

```
.model small
.stack 100H
.data
msg db 10,13,"Enter the first no.:: $"
msg1 db 10,13,"Enter the second no.:: $"
msg2 db 10,13,"The Resultant sum is :: $"

.code
.startup

 MOV AH,09
 MOV DX,OFFSET msg
 INT 21H

 MOV AH, 01
 INT 21H
 SUB AL,30H
 MOV BL, AL

 MOV AH,09
 MOV DX,OFFSET msg1
 INT 21H

 MOV AH, 01
 INT 21H
 SUB AL,30H
 ADD BL,AL

 MOV AH,09
 MOV DX,OFFSET msg2
 INT 21H

 MOV DL,BL
 CMP DL, 09
 JG L6
```

```
 ADD DL,30H
 JMP L7

 L6: ADD DL, 37H

 L7: MOV AH,02
 INT 21H

 MOV AH, 4CH
 INT 21H
.exit
end
```

- 2 digit

```
.model small
.stack 100H

.data
 data1 db 00H
 msg db 10,13,"Enter the first no.:: $"
 msg1 db 10,13,"Enter the second no.:: $"
 msg2 db 10,13,"The Resultant sum is :: $"

.code
.startup

 MOV BL, 00

 MOV AH,09
 MOV DX,OFFSET msg
 INT 21H

 MOV CX, 2
 AGAIN: MOV AH, 01
        INT 21H
        CMP AL, 'A'
        JGE P1
        SUB AL,30H
        JMP P4
```

```
            P1: SUB AL, 37H
            P4: SHL BL, 4
            ADD BL, AL
LOOP AGAIN

MOV data1, BL

MOV AH,09
MOV DX,OFFSET msg1
INT 21H

MOV CX, 2
AGAIN2: MOV AH, 01
      INT 21H

      CMP AL, 'A'
      JGE P2
      SUB AL,30H
      JMP P3
      P2: SUB AL, 37H
      P3: SHL BL, 4
      ADD BL,AL
LOOP AGAIN2

ADD BL,data1

MOV AH,09
MOV DX,OFFSET msg2
INT 21H

MOV DL, 00
MOV CX, 2
AGAIN3: ROL BL, 4
      MOV DL,BL
       AND DL, 0FH
       CMP DL, 09
       JG L6
       ADD DL,30H
      JMP L7
```

```
        L6: ADD DL, 37H
        L7: MOV AH,02
        INT 21H
 LOOP AGAIN3

 MOV AH, 4CH
 INT 21H
.exit
End
```

- **4 digit**

```
.model small
.stack 100H

.data
  data1 dw 00
  msg db 10,13,"Enter the first no.:: $"
  msg1 db 10,13,"Enter the second no.:: $"
  msg2 db 10,13,"The Resultant sum is :: $"

.code
.startup
  MOV BX, 00

  MOV AH,09
  MOV DX,OFFSET msg
  INT 21H

  MOV CX, 4
  AGAIN: MOV AH, 01
        INT 21H
        CMP AL, 'A'
        JGE P1
        SUB AL,30H
        JMP P4
        P1: SUB AL, 37H
        P4: SHL BX, 4
        MOV AH,00
```

```
        ADD BX, AX
LOOP AGAIN

MOV data1, BX

MOV AH,09
MOV DX,OFFSET msg1
INT 21H

MOV CX, 4
AGAIN2: MOV AH, 01
        INT 21H

        CMP AL, 'A'
        JGE P2
        SUB AL,30H
        JMP P3
        P2: SUB AL, 37H
        P3: SHL BX, 4
        MOV AH,00
        ADD BX,AX
LOOP AGAIN2

ADD BX, data1

MOV AH,09
MOV DX,OFFSET msg2
INT 21H

MOV DX, 00
MOV CX, 4
AGAIN3: ROL BX, 4
            MOV DL,BL
            AND DL, 0FH
            CMP DL, 09
              JG L6
            ADD DL,30H
            JMP L7
```

```
        L6: ADD DL, 37H

        L7: MOV AH,02
        INT 21H
  LOOP AGAIN3

  MOV AH, 4CH
  INT 21H
.exit
end
```

- **8 digit**

```
.model small
.386
.stack 100H     ;stack size
.data
  data1 dd 00
  msg db 10,13,"Enter the first number::$"     ;db:define byte
  msg1 db 10,13,"Enter the second number::$"
  msg2 db 10,13,"The Resultant sum is::$"

.code
.startup

  MOV ebx,00  ;TURNED TO ZERO FROM REWRITING

  MOV AH,09
  MOV DX,OFFSET msg
  INT 21H

  MOV CX,8    ;TAKING 8 digit INPUT
  AGAIN:MOV AH,01  ;LOOP
        INT 21H
        CMP  AL, 'A'    ;IF A<=AL
        JGE P1        ;if TRUE
        SUB AL,30H    ;if false
        JMP P4

        P1:SUB AL,37H
```

```
        P4:SHL ebx,4    ;shift left

        MOV AH,00
        ADD ebx,eax
LOOP AGAIN

MOV data1,ebx

MOV AH,09
MOV DX,OFFSET msg1
INT 21H

MOV CX,8    ;TAKING 8 digit INPUT
AGAIN2:MOV AH,01  ;LOOP
      INT 21H
      CMP  AL,'A'   ;IF A<=AL
      JGE P2       ;IF TRUE
      SUB AL,30H  ;IF FALSE
      JMP P3

      P2:SUB AL,37H
      P3:SHL ebx,4    ;shift left
      MOV AH,00
      ADD ebx,eax
LOOP AGAIN2

ADD ebx,data1

MOV  AH,09
MOV DX,OFFSET msg2
INT 21H

MOV DX,00
MOV CX,8 ;PRINT 8-DIGIT
AGAIN3: ROL ebx,4
       MOV DL,BL
       AND DL,0FH
       CMP DL,09
       JGE L6
```

```
        ADD DL,30H
        JMP L7

        L6:ADD DL,37H

        L7:MOV AH,02
        INT 21H
  LOOP AGAIN3

.exit
end
```

SUBTRACTION

- **1 digit**

```
.model small
.stack 100H
.data
msg db 10,13,"Enter the first no.:: $"
msg1 db 10,13,"Enter the second no.:: $"
msg2 db 10,13,"The Resultant sum is :: $"

.code
.startup

  MOV AH,09
  MOV DX,OFFSET msg
  INT 21H

  MOV AH, 01
  INT 21H
  SUB AL,30H
  MOV BL, AL

  MOV AH,09
  MOV DX,OFFSET msg1
  INT 21H

  MOV AH, 01
  INT 21H
```

```
    SUB AL,30H
    SUB BL,AL

    MOV AH,09
    MOV DX,OFFSET msg2
    INT 21H

    MOV DL,BL
    CMP DL, 09
    JG L6
    ADD DL,30H
    JMP L7

    L6: ADD DL, 37H

    L7: MOV AH,02
    INT 21H

    MOV AH, 4CH
    INT 21H
   .exit
  end
```

- **2 digit**

```
.model small
.stack 100H

.data
 data1 db 00H
 msg db 10,13,"Enter the first no.:: $"
 msg1 db 10,13,"Enter the second no.:: $"
 msg2 db 10,13,"The Resultant sum is :: $"

.code
.startup

 MOV BL, 00

 MOV AH,09
```

```
MOV DX,OFFSET msg
INT 21H

MOV CX, 2
AGAIN: MOV AH, 01
INT 21H
CMP AL, 'A'
JGE P1
SUB AL,30H
JMP P4
P1: SUB AL, 37H
P4: SHL BL, 4
ADD BL, AL
LOOP AGAIN

MOV data1, BL

MOV AH,09
MOV DX,OFFSET msg1
INT 21H

MOV CX, 2
AGAIN2: MOV AH, 01
    INT 21H

    CMP AL, 'A'
    JGE P2
    SUB AL,30H
    JMP P3
    P2: SUB AL, 37H
    P3: SHL BL, 4
    ADD BL,AL
LOOP AGAIN2

SUB BL,data1

MOV AH,09
MOV DX,OFFSET msg2
INT 21H
```

```
MOV DL, 00
MOV CX, 2
AGAIN3: ROL BL, 4
        MOV DL,BL
        AND DL, 0FH
        CMP DL, 09
        JG L6
        ADD DL,30H
        JMP L7

        L6: ADD DL, 37H
        L7: MOV AH,02
        INT 21H
LOOP AGAIN3

MOV AH, 4CH
INT 21H
.exit
   end
```

- **4 digit**

```
   .model small
   .stack 100H

   .data
     data1 dw 00
     temp dw 00
     msg db 10,13,"Enter the first no.:: $"
     msg1 db 10,13,"Enter the second no.:: $"
     msg2 db 10,13,"The Resultant sum is :: $"

   .code
   .startup

     MOV BX, 00

     MOV AH,09
     MOV DX,OFFSET msg
```

```
INT 21H

MOV CX, 4
AGAIN: MOV AH, 01
      INT 21H
      CMP AL, 'A'
      JGE P1
      SUB AL,30H
      JMP P4
      P1: SUB AL, 37H
      P4: SHL BX, 4
      MOV AH,00
      ADD BX, AX
LOOP AGAIN

MOV data1, BX

MOV AH,09
MOV DX,OFFSET msg1
INT 21H

MOV CX, 4
AGAIN2: MOV AH, 01
      INT 21H

      CMP AL, 'A'
      JGE P2
      SUB AL,30H
      JMP P3
      P2: SUB AL, 37H
      P3: SHL BX, 4
      MOV AH,00
      ADD BX,AX
LOOP AGAIN2

MOV temp,BX
MOV BX,data1
SUB BX,temp
```

```
  MOV AH,09
  MOV DX,OFFSET msg2
  INT 21H

  MOV DX, 00
  MOV CX, 4
  AGAIN3: ROL BX, 4
          MOV DL,BL
          AND DL, 0FH
          CMP DL, 09
          JG L6
          ADD DL,30H
          JMP L7

          L6: ADD DL, 37H

          L7: MOV AH,02
          INT 21H
  LOOP AGAIN3

  MOV AH, 4CH
  INT 21H
.exit
End
```

- **8 digit**

```
.model small
.386
.stack 100H     ;stack size
.data
  data1 dd 00
  temp dd 00
  msg db 10,13,"Enter the first number::$"     ;db:define byte
  msg1 db 10,13,"Enter the second number::$"
  msg2 db 10,13,"The Resultant sum is::$"

.code
.startup
```

```
MOV ebx,00  ;TURNED TO ZERO FROM REWRITING

MOV AH,09
MOV DX,OFFSET msg
INT 21H

MOV CX,8    ;TAKING 8 digit INPUT
AGAIN:MOV AH,01  ;LOOP
        INT 21H
        CMP  AL, 'A'    ;IF A<=AL
        JGE P1        ;if TRUE
        SUB AL,30H    ;if false
        JMP P4

        P1:SUB AL,37H
        P4:SHL ebx,4    ;shift left

        MOV AH,00
        ADD ebx,eax
LOOP AGAIN

MOV data1,ebx

MOV AH,09
MOV DX,OFFSET msg1
INT 21H

MOV CX,8    ;TAKING 8 digit INPUT
AGAIN2:MOV AH,01  ;LOOP
      INT 21H
      CMP  AL,'A'   ;IF A<=AL
      JGE P2       ;IF TRUE
      SUB AL,30H  ;IF FALSE
      JMP P3

      P2:SUB AL,37H
      P3:SHL ebx,4    ;shift left
      MOV AH,00
      ADD ebx,eax
```

```
  LOOP AGAIN2

  MOV temp,ebx
  MOV ebx,data1

  SUB ebx,temp

  MOV  AH,09
  MOV DX,OFFSET msg2
  INT 21H

  MOV DX,00
  MOV CX,8 ;PRINT 8-DIGIT
  AGAIN3: ROL ebx,4
        MOV DL,BL
        AND DL,0FH
        CMP DL,09
        JGE L6
        ADD DL,30H
        JMP L7

        L6:ADD DL,37H

        L7:MOV AH,02
        INT 21H
  LOOP AGAIN3

.exit
end
```