° PROBLEM 3

coefficients → 2's COMPLEMENT values on 3 bits : $-2^2 \leq N \leq 2^2 - 1 \equiv [-4, +3]$

$\Delta \neq 0$

° Number of bits to represent the determinants:

WORSE CASE: $\Delta = [-4 * (-4) - 0] = 16$

To represent number 16, you need 6 bits:

5 bits are not enough because you can only represent a maximum value equal to $2^4 - 1 = 15$

(as positive value).

↳ Variables to store determinants:

DET	DB	0	; Δ
DET_X	DB	0	; Δx
DET_Y	DB	0	; Δy

° Number of bits to represent the results

WORSE CASE: $\Delta = 1$

$\Rightarrow$ x_int = Δx or y_int = Δy

Same analysis performed above $\Rightarrow$ 6 bits needed (Variable on 8 bits)

To compute the result you need a 16/8 division, so that the integer part will be represented on 8 bits.

° Compute determinants:

- 2 IMUL 8 × 8 ... but only the smallest 8 bits are taken
- 1 SUB 16 - 16 → (because of the analysis performed above)

; Input part : read coefficient's values and store them in the corresponding variables.
; Assuming that : the input is correct, coefficient ∈ [-4, +3], positive numbers are written WITHOUT "+".

; Reading a coefficient   (the code is always the same for all coefficients)

```
    MOV  AH, 01H        ; read one char from keyboard
    INT  21H
    CMP  AL, '-'
    JE   negativeNumber
    SUB  AL, '0'
    MOV  COEFF_A, AL
    JMP  exit1
    INT  21H
    SUB  AL, '0'
    NEG  AL
    MOV  COEFF_A, AL
```

; Compute the determinants (only Δ, for the other ones the code is the same)

```
    MOV AL, COEFF_A
    MOV CL, COEFF_E
    IMUL CL              ; AL = a*e          (use IMUL instead of MUL because of signed numbers)
    MOV DET, AL
    MOV AL, COEFF_B
    MOV CL, COEFF_D
    IMUL CL              ; AL = b*d
    SUB DET, AL          ; DET = Δ = a*e - b*d
```

```asm
; Compute the final value of X_INT and Y_INT    (actually, only if Δ ≠ 0)
; Compute X_INT   (Y_INT is computed in the same way)
  MOV  AL, DET_X
  CBW
  MOV  CL, DET
  IDIV CL
  MOV  X_INT, AL
         └── DB


; Feature A  ( Δ = 0 )
; if Δx ≠ 0 and Δy ≠ 0 ⇒ impossible   else if Δx = 0 and Δy = 0 ⇒ undetermined
  CMP  DET, 0
  JE   verifyType
  ; otherwise  (Δ ≠ 0), compute x_int and y_int as above
  CMP  DET_X, 0
  JE   undetermined
  ; otherwise  print "impossible"  and return to user menu
undetermined:
  ;print "undetermined", and return to user menu
```

```asm
; Feature B → compute overall squared error
; Assume to define SQ_ERR DW ?   and TMP DW ?

MOV AL, COEFF_A
MOV CL, X_INT
IMUL CL                 ; AX = a·X_INT
MOV SQ_ERR, AX
MOV AL, COEFF_B
MOV CH, Y_INT
IMUL CH                 ; AX = b·Y_INT
ADD SQ_ERR, AX          ; a·X_INT + b·Y_INT
MOV AL, COEFF_C
CBW
SUB  SQ_ERR, AX         ; a·X_INT + b·Y_INT - c
MOV  AX, SQ_ERR
PUSH CX
MOV  CX, SQ_ERR
IMUL CX
MOV SQ_ERR, AX
POP CX
MOV AL, COEFF_D
IMUL CL
MOV TMP, AX
MOV AL, COEFF_E
IMUL CH
ADD TMP, AX
MOV AL, COEFF_F
CBW
SUB TMP, AX
MOV AX, TMP
MOV CX, TMP
IMUL CX
ADD  SQ_ERR, AX
```