Q. Write a program to evaluate the following arithmetic statement by using Three, Two, One and Zero address instruction

(a) A - B + (C+D) / E + F / G + (J*K)

(b) (A * B+C) + D / E + F * G + I * J

Ans. (a) A - B + (C+D) / E + F / G + (J * K)

Using Three-Address instructions :

$$\frac{A-B+(C+D)}{E+F/G+(J*K)}$$

> Note: Always use minimum number of instructions and minimun number of distinct registers

ADD	R1, C, D	R1 ← M[C] + M[D]
SUB	R2, A, B	R2 ← M[A] - M[B]
ADD	R1, R2, R1	R1 ← R2 + R1
MUL	R2, J, K	R2 ← M[J] * M[K]
ADD	R2, G, R2	R2 ← M[G] + R2
ADD	R3, E, F	R3 ← M[E] + M[F]
DIV	R3, R3, R2	R3 ← R1 / R2
DIV	X, R1, R3	X ← R1 / R3

Using Two-Address instructions

MOV	R1, C	R1 ← M[C]
ADD	R1, D	R1 ← R1 + M[D]
MOV	R2, A	R2 ← M[A]
SUB	R2, B	R2 ← R2 - M[B]
ADD	R1, R2	R1 ← R1 + R2
MOV	R2, J	R2 ← M[J]

MUL	R2, K	R2 ← R2 * M[K]
ADD	R2, G	R2 ← R2 + M[G]
MOV	R3, E	R3 ← M[E]
ADD	R3, F	R3 ← R3 + M[F]
DIV	R3, R2	R3 ← R3 / R2
DIV	R1, R3	R1 ← R1 / R3
MOV	X, R1	X ← R1

Using One-Address instructions.

{AC - Accumulator}

LOAD	J	AC ← M[J]
MUL	K	AC ← AC * M[K]
STORE	T	M[T] ← J * K
LOAD	G	AC ← M[G]
ADD	T	AC ← AC + M[T]
STORE	T	M[T] ← AC
LOAD	E	AC ← M[E]
ADD	F	AC ← AC + M[F]
DIV	T	AC ← AC / M[T]
STORE	T	M[T] ← AC
LOAD	C	AC ← M[C]
ADD	D	AC ← AC + M[D]
STORE	G	M[G] ← AC
LOAD	A	AC ← M[A]
SUB	B	AC ← AC - M[B]
ADD	G	AC ← AC + M[G]
DIV	T	AC ← AC / M[T]
STORE	X	M[X] ← AC

Zero-Address instructions :

For Zero-Address instructions, First write the given arithmetic expression into postfix notation.

Here the arithmetic ~~notation is~~ expression is

$$\frac{A-B+(C+D)}{E+F/G+(J*K)}$$

whose Postfix notation is

$$AB-CD++EFG/+JK*+/$$

PUSH	A	TOS ← A
PUSH	B	TOS ← B
SUB		TOS ← A-B
PUSH	C	TOS ← C
PUSH	D	TOS ← D
ADD		TOS ← C+D
ADD		TOS ← A-B+(C+D)
PUSH	E	TOS ← E
PUSH	F	TOS ← F
PUSH	G	TOS ← G
DIV		TOS ← F/G
ADD		TOS ← E+F/G
PUSH	J	TOS ← J
PUSH	K	TOS ← K
MUL		TOS ← J*K
ADD		TOS ← E+F/G+(J*K)
DIV		TOS ← $\frac{A-B+(C+D)}{E+F/G+(J*K)}$
POP	X	M[X] ← TOS

Q. Multiply -9 and -13 using Booth's algorithm

Ans.

Here, Multiplicand (M) = -9 = 10111 (2's complement)

Multiplier (Q) = -13 = 10011 (2's complement)

Note: M = 10111

-M = 01001 → 2's complement

Tracin Table

n	A	Q	V_0	operation
5	00000	10011	0	Initial
				A ← A-M ⇒ A+(-M) = 00000 + 01001 = 01001
	01001	10011	0	
	00100	11001	1	ashr (arithmetic shift right)
4	00010	01100	1	ashr
3	11001	01100	1	A ← A+M ⇒ 00010 + 10111 = 11001
	11100	10110	0	ashr
2	11110	01011	0	ashr
1	00111	01011	0	A ← A-M ⇒ A+(-M) = 11110 + 01001 = 00111
	00011	10101	1	ashr

→ Final output = 0001110101

= +117 Ans.