**Count +ve and –ve numbers in an array**

DATA SEGMENT

ARR DB -1, 2, -3, 4, -5, 6, -7, 8, -9, 10 ; Array of numbers

SIZE EQU $-ARR ; Size of the array

POS\_COUNT DB 0 ; Counter for positive numbers

NEG\_COUNT DB 0 ; Counter for negative numbers

DATA ENDS

CODE SEGMENT

ASSUME CS:CODE, DS:DATA

START:

MOV AX, DATA ; Load the data segment address into AX

MOV DS, AX ; Move the data segment address to DS

MOV CX, SIZE ; Load the size of the array into CX

MOV SI, 0 ; Initialize the source index to 0

MOV BL, 0 ; Initialize the positive counter to 0

MOV BH, 0 ; Initialize the negative counter to 0

LOOP\_START:

MOV AL, ARR[SI] ; Load a number from the array into AL

CMP AL, 0 ; Compare the number with 0

JG POS\_NUM ; Jump if greater (positive)

JL NEG\_NUM ; Jump if less (negative)

NEXT\_NUM:

INC SI ; Increment the source index

LOOP LOOP\_START ; Loop until all elements of the array are processed

POS\_NUM:

INC BL ; Increment the positive counter

JMP NEXT\_NUM ; Jump to process the next number

NEG\_NUM:

INC BH ; Increment the negative counter

JMP NEXT\_NUM ; Jump to process the next number

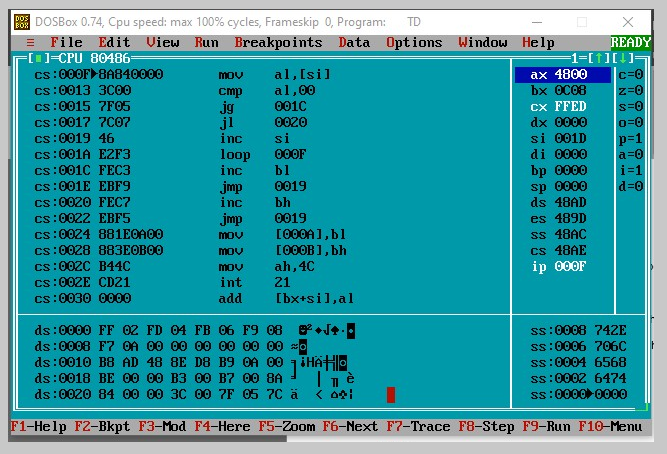
MOV POS\_COUNT, BL ; Store the final count of positive numbers in the counter

MOV NEG\_COUNT, BH ; Store the final count of negative numbers in the counter

MOV AH, 4CH ; Set the exit function number

INT 21H ; Invoke the DOS interrupt to terminate the program

CODE ENDS END START



**Add all +ve numbers in an array**

DATA SEGMENT

ARR DB -1, 2, -3, 4, -5, 6, -7, 8, -9, 10 ; Array of numbers

SIZE EQU $-ARR ; Size of the array

POS\_SUM DW 0 ; Variable to store the sum of positive numbers

NEG\_SUM DW 0 ; Variable to store the sum of negative numbers

DATA ENDS

CODE SEGMENT

ASSUME CS:CODE, DS:DATA

START:

MOV AX, DATA ; Load the data segment address into AX

MOV DS, AX ; Move the data segment address to DS

MOV CX, SIZE ; Load the size of the array into CX

MOV SI, 0 ; Initialize the source index to 0

MOV AX, 0 ; Initialize the sum of positive numbers to 0

MOV DX, 0 ; Initialize the sum of negative numbers to 0

LOOP\_START:

MOV AL, ARR[SI] ; Load a number from the array into AL

CMP AL, 0 ; Compare the number with 0

JG POS\_NUM ; Jump if greater (positive)

JL NEG\_NUM ; Jump if less (negative)

NEXT\_NUM:

INC SI ; Increment the source index

LOOP LOOP\_START ; Loop until all elements of the array are processed

POS\_NUM:

ADD AX, DX ; Add the number to the sum of positive numbers

JMP NEXT\_NUM ; Jump to process the next number

NEG\_NUM:

ADD DX, AX ; Add the number to the sum of negative numbers

JMP NEXT\_NUM ; Jump to process the next number

MOV POS\_SUM, AX ; Store the final sum of positive numbers in the POS\_SUM variable

MOV NEG\_SUM, DX ; Store the final sum of negative numbers in the NEG\_SUM variable

MOV AH, 4CH ; Set the exit function number

INT 21H ; Invoke the DOS interrupt to terminate the program

CODE ENDS END START