**Count Odd 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

COUNT DB 0 ; Counter to store the count of odd 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 counter to 0

LOOP\_START:

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

AND AL, 1 ; Perform bitwise AND with 1 to check if it's odd

CMP AL, 1 ; Compare AL with 1 to check if it's odd

JNZ NOT\_ODD ; Jump if it's not odd

INC BL ; Increment the counter if it's odd

NOT\_ODD:

INC SI ; Increment the source index

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

MOV COUNT, BL ; Store the final count of odd 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

**Count Even 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

COUNT DB 0 ; Counter to store the count of even 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 counter to 0

LOOP\_START:

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

AND AL, 1 ; Perform bitwise AND with 1 to check if it's odd

JNZ ODD\_NUM ; Jump if it's odd

INC BL ; Increment the counter if it's even

ODD\_NUM:

INC SI ; Increment the source index

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

MOV COUNT, BL ; Store the final count of even 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 odd numbers in 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

SUM DW 0 ; Variable to store the sum of odd 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 to 0

LOOP\_START:

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

AND AL, 1 ; Perform bitwise AND with 1 to check if it's odd

JZ EVEN\_NUM ; Jump if it's even

ADD AX, SUM ; Add the number to the sum if it's odd

EVEN\_NUM:

INC SI ; Increment the source index

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

MOV SUM, AX ; Store the final sum of odd numbers in the SUM variable

MOV AH, 4CH ; Set the exit function number

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

CODE ENDS

END START