LAB # 09 - TASKS

TASK # 01

You are developing a simple grading system for a teacher. Write an Assembly Programming Language program using the Irvine Library that stores the marks of students in an array and counts how many students passed and how many failed. A student passes if the mark is greater than or equal to 50.

Use a procedure to perform the counting of passed and failed students, and then display the results in the main program.

Sample Input:
Marks = [45, 78, 90, 32, 60, 49, 85]

Sample Output:
Passed students = 4
Failed students = 3

TASK # 02

Write a program using the Irvine Library that multiplies two positive integers without using the MUL, IMUL, or repeated addition method.
Use bitwise shift and conditional logic to perform the multiplication (i.e., based on the binary representation of the multiplier).
Display the final product.

Sample Input:

First number = 6
Second number = 5

Sample Output:

Product = 30

TASK # 03

You are designing a simple encryption system for securing small 8-bit data values.
Write a program using the Irvine Library that takes an 8-bit binary value as input and “encrypts” it by performing the following operations:

1. Rotate the bits of the value to the left twice using the ROL instruction.
2. Shift the rotated result to the right once using the SHR instruction to simulate bit masking.
3. Display the final “encrypted” result.

Sample Input:

Original Value = 10110110b

Sample Output:

Encrypted Value = 01101101b

TASK # 04

You are developing a performance analyzer for a small embedded system that processes sensor readings.
Write a program using the Irvine Library that performs the following tasks:

1. Store 10 sensor readings in an array.
2. Use a procedure to calculate:

   - The sum of all readings.
   - The average of the readings (using DIV).

3. Use conditional jumps to count how many readings are above and below the average.
4. Perform a bitwise shift (SHL) on the average to simulate data scaling (multiply by 2).
5. Perform a bitwise rotate (ROR) on the scaled value to simulate data encryption.
6. Display:

   - The sum of readings
   - The average reading
   - The number of readings above and below average
   - The scaled average
   - The rotated (encrypted) result

Sample Input:

Sensor readings = [12, 25, 30, 40, 18, 22, 27, 35, 20, 15]

Sample Output:

Sum of readings = 244
Average reading = 24
Readings above average = 5
Readings below average = 5
Scaled average (SHL) = 48
Encrypted result (ROR) = 10011000b