Report on ARM32 Assembly Program for Sales Data Analysis

Introduction

The task involves processing sales data over five days, where each day contains variable-length entries marked by a 0 to signal the end. The program computes:

- Total sales over all five days
- Average sales per day
- Maximum and minimum sales across all days

Approach

The sales data for each day is stored in arrays where 0 shows the end of the data. Five separate arrays are used, one for each day. The program processes these arrays sequentially, summing the sales values, counting the total number of sales, and determining the highest and lowest sales values.

Code Logic

Initialization: The program initializes registers to store the cumulative total sales, count the number of sales, and track the maximum and minimum sales. The day counter is also set to 5.

Processing Sales: The program iterates through the sales data using a loop. For each sale:

- It adds the value to the total sales.
- It increments the sales count.
- It compares the value with the current maximum and minimum sales and updates these as necessary.
- If the value 0 is encountered, it marks the end of the current day's sales, and the program moves on to the next day.

Handling End of Day: When 0 is encountered, indicating the end of a day's sales data, the program reduces the day counter. If all five days are processed, it exits the loop.

Storing Results: After processing all five days, the total sales are stored in memory. The average sales are calculated by dividing the total sales by 5 using a custom division function. The maximum and minimum sales are also stored in memory.

Division Subroutine: The division subroutine uses a loop to repeatedly subtract the divisor (5, for the number of days) from the total sales until the remaining amount is less than the divisor. This gives the quotient, which represents the average sales.

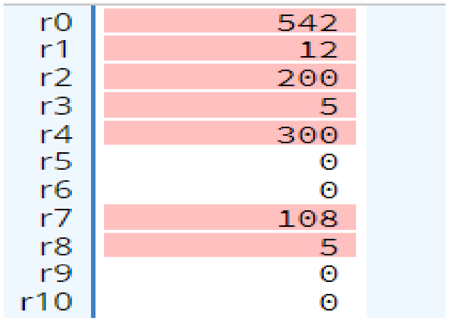
Testing and Verification

The program was tested in the ARM32 cpulator environment and produced correct results:

- **Total Sales**: The sum of all sales (542) was accurately computed.
- **Average Sales**: The program correctly divided the total sales by 5 (108).
- **Maximum and Minimum Sales**: The highest sale (200) and lowest sale (5) were identified.

Conclusion

This ARM32 assembly program efficiently processes variable-length sales data, computes total and average sales, and identifies maximum and minimum values. It avoids hardcoded assumptions, uses clear conditional logic, and performs division using an efficient custom subroutine. The solution was verified through testing.



```
 r0        542
 r1         12
 r2        200
 r3          5
 r4        300
 r5          0
 r6          0
 r7        108
 r8          5
 r9          0
r10          0
```