Project Assignment 1: Matrix Operations using Intel Assembly

Description of Assignment

A square matrix is represented by a 2-dimensional array of n rows and n columns. All matrix operations to be implemented in this project are supposed to work on square matrices of size n, which will be given in the first line of the input files containing matrices. All matrices are signed integer matrices.

As an example, a 5x5 matrix is stored in a text file as follows:

```
5
3     6     3     7     11
1     1     0     17    9
2     7     0     -6    4
0     4     2     9     13
21    71    65    32    11
```

A group of functions written in Intel assembly language will be called from a program in C. The main program will read two matrices, A and B from two files each containing one matrix. Then in a simple menu loop, the user will be given the choice to select the matrix operation. The result of each operation performed will be shown on screen. The program will end when the user selects the quit option from the menu.

Functions are not required to perform error checking on the parameters, but must follow C calling conventions and must fully conform to the given prototypes.

Following functions are to be implemented:

Function 1.

```
void add(int *matrix1, int *matrix2, int *result, int size);
```

This function will add the two matrices given, and return the result in result array.

Function 2.

```
int sum(int *matrix1, int size);
```

This function will add up the values in a matrix, and return the sum.

Function 3.

```
void mult(int *matrix1, int *matrix2, int *result, int size);
```

This function will multiply (matrix multiplication) the two matrices given, and return the result in result array.

Function 4.

```
void scale(int *matrix1, int number, int *result, int size);
```

This function will do a scalar multiplication of the matrix with the number given, and return the result in result array.

Function 5.

```
void square(int *matrix1, int *result, int size);
```

This function will take the square of the given matrix by calling the `mult()` function above, and return the result in result array.

Function 6.

```
void ITU(int *matrix1, int size);
```

This function will create a binary matrix, where the ITU letters will appear at the center as '1's clusters on a '0' background. Size of the letters will depend on the size of the matrix. Here size can be between 11 and 80. For other values, an error message should be printed out.

Submission Details

You are required to implement the given 6 functions in Intel assembly. The function implementations must fully conform to the provided prototypes since they are expected to be linked to the main program implemented in C.

You are required to submit the assembly language source code file(s) through the Ninova system as a zip file.Each member of the group must make a submission, even though the submitted files may be the same for all group members.

Group members will be graded individually based on their performance in the lab session and the submitted group project. The students who are not present during the lab session will not receive a grade for the project, even though they may have made a submission through the Ninova system.

Any form of cheating or plagiarism will not be tolerated. This includes actions such as, but not limited to, submitting the work of others as one's own (even if in part and even with modifications) and copy/pasting from other resources (even when attributed). Serious offenses will be reported to the administration for disciplinary measures.