ASSIGNMENT 5
Programming Assignment

Name: K.Vishwanath
Course Title: Development of Real-Time System
Date: 05-07-2022

Assignment:

Create a task "matrixtask" containing the following functionality:

```
#define SIZE 10
#define ROW SIZE
#define COL SIZE
static void matrix_task()
{
    int i;
    double **a = (double **)pvPortMalloc(ROW * sizeof(double*));
    for (i = 0; i < ROW; i++) a[i] = (double *)pvPortMalloc(COL *
sizeof(double));
    double **b = (double **)pvPortMalloc(ROW * sizeof(double*));
    for (i = 0; i < ROW; i++) b[i] = (double *)pvPortMalloc(COL *
sizeof(double));
    double **c = (double **)pvPortMalloc(ROW * sizeof(double*));
    for (i = 0; i < ROW; i++) c[i] = (double *)pvPortMalloc(COL *
sizeof(double));

    double sum = 0.0;
    int j, k, l;

    for (i = 0; i < SIZE; i++) {
        for (j = 0; j < SIZE; j++) {
            a[i][j] = 1.5;
            b[i][j] = 2.6;
        }
    }

    while (1) {
        /*
        * In an embedded systems, matrix multiplication would block the
CPU for a long time
        * but since this is a PC simulator we must add one additional
dummy delay.
        */
        long simulationdelay;
        for (simulationdelay = 0; simulationdelay<1000000000;
simulationdelay++)
            ;
        for (i = 0; i < SIZE; i++) {
            for (j = 0; j < SIZE; j++) {
                c[i][j] = 0.0;
            }
        }

        for (i = 0; i < SIZE; i++) {
            for (j = 0; j < SIZE; j++) {
```

```
                    sum = 0.0;
                    for (k = 0; k < SIZE; k++) {
                        for (l = 0; l<10; l++) {
                            sum = sum + a[i][k] * b[k][j];
                        }
                    }
                    c[i][j] = sum;
                }
            }
            vTaskDelay(100);
        }
}
```

-Create a queue and send the content of (double **)c to the queue in matrix task with before the vTaskDelay() call (hint: place the c variable in a struct). (More information Here).

-Create a reader task which reads the content of the queue in case there is something in the queue.

-In case the queue has some content it should save the data in a local (double **) variable.

-Print out the content of the (double **)c variable in case the content is updated. The data transferred from c should be a 10x10 matrix with the value 390 in each slot.

The following should be provided in a written report:

- A screenshot of the execution