

COE 301 – Computer Organization

MIPS Programming Assignment 3, Term 221

Matrix Vector Multiplication

Matrix-vector multiplication can be described by the following function:

```
float* MVM (int n, float A[n][n], float X[n]) {
    float* V = new float[n];          // allocate an array of n floats
    int i, j;
    for (i=0; i<n; i++) {
        float sum = 0;
        for (j=0; j<n; j++) { sum = sum + A[i][j] * X[j]; }
        V[i] = sum;
    }
    return V;                          // return a pointer to vector V
}
```

In addition, you need the following support functions:

```
float* read_matrix (int n) {
    // allocate a matrix of n*n floats
    // ask the user to input n*n floats and read them into allocated matrix
    // return address of matrix
}

float* read_vector (int n) {
    // allocate a vector of n floats
    // ask the user to input n floats and read them into allocated vector
    // return address of vector
}

void print_vector (int n, float V[n]) {
    // Display the n elements of vector V
}
```

- a) (15 points) Write a **main** function that asks the user to input **n**. Call functions **read_matrix** and **read_vector** from **main** to read a matrix and a vector. Call function **MVM** to do matrix-vector multiplication. Then call **print_vector** to print the result vector.
- b) (20 points) Write a function **read_matrix** that allocates an array of **n×n** floats dynamically on the heap, asks the user to input all **n²** elements, stores the values in the matrix (starting at row 0, then row 1, etc.), and returns a pointer (address) of the dynamically allocated matrix.

- c) (15 points) Write a function **read_vector** that allocates an array of **n** floats dynamically on the heap, asks the user to input all **n** elements, stores the values in the array, and returns a pointer (address) of the dynamically allocated array.
- d) (30 points) Write a function **MVM** that multiplies a matrix by a vector and stores the result in a vector, which is dynamically allocated on the heap. The result vector should have **n** floats, where **n** is passed as the first argument to the function. The function should return the address of the dynamically allocated vector.
- e) (10 points) Write a function **print_vector** that prints the **n** float elements of the vector, whose address is passed as an argument to the function.
- f) (10 points) For passing arguments properly, returning results, program readability, and comments.

The late penalty is -1 point for two days late. A program is not accepted if it more than two days late.

Submission Guidelines:

This assignment can be solved individually or in groups of two students only. No group should have more than two students. Both students should contribute to the solution. **At the beginning of your program, write the names of the student(s) who worked on the program.**

All submissions should be done through Blackboard. Submit the source code of the program. Make sure that your program is well written and documented.