



INTRODUCTION TO COMPUTERS II

LAB 4: C AND ASSEMBLY LANGUAGE

1. Implement the following C program to determine which of two vectors is farther from the origin. Build and debug this program **in C** (use the **pr4.c** file of the **pr4** project in the Workspace):

```
#define N 5

//the two N-component vectors
int U[N] = {5, 2, -3, 7, 6};
int V[N] = {6, -1, 1, 0, 3};
//variable to store the result: 1 if U is greater, 0 otherwise
char greater_u;

/**
 * Function that saves a value in the provided pointer
 */
void save(char value, char * location) {
    *location = value;
}

/**
 * Simple function to multiply two numbers in an iterative way
 */
int mul(int a, int b) {
    int res = 0, sign = 0;
    if (a < 0) {
        sign = 1;
        a = -a;
    }
    while (a--) res += b;
    if (sign)
        return -res;
    else
        return res;
}

/**
 * Simple function that calculates the square root in an iterative way
 */
int i_sqrt(int a) {
    int root = 0;
    while (mul(root, root) < a) {
        root++;
    }
    return root;
}
```

```

/**
 * The euclidean distance is calculated. All the squares are added and
 finally
 * the (integer) square root is applied.
 */
int eucl_dist(int w [], int size) {
    int acc = 0;
    for (int i = 0; i < size; i++) {
        acc += mul(w[i], w[i]);
    }
    return i_sqrt(acc);
}

/**
 * Entry point to the program
 */
void main() {
    //the euclidean distance to the origin is calculated
    int d_u = eucl_dist(U, N);
    int d_v = eucl_dist(V, N);
    //we check if U was farther
    char greater = d_u > d_v;
    save(greater, &greater_u);

    while(1); //the program remains in an infinite loop
}

```

2. In order to produce a hybrid C / Assembly program, translate the **eucl_dist** and **save** functions into the RISC-V assembly language. To do so, use the **fun_asm.asm** file of the **pr4** project in the Workspace:

```

//fill with the .extern and .global directives
//with the appropriate functions

//int eucl_dist(int * w, int size);
eucl_dist:
    //the address of W is received in a0, and size N in a1
    //perform the calculations
    //the result is returned in a0

//int save(char value, char * location);
save:
    //the value is received in a0, and the destination address in a1
    //make sure that only ONE BYTE is saved!!

```

Remember to declare the **eucl_dist** and **save** functions as **extern** in the **pr4.c** file (and remove their original declaration) and export them with **.global** in the **fun_asm.asm** file. Also, you need to declare the **mul** and **i_sqrt** functions as **.extern** in the **fun_asm.asm** file (since they are defined in the **pr4.c** file).