

Simple RISC Program

<http://www.cse.iitd.ernet.in/~srsarangi/archbooksoft.html>

Simulators:

See the course webpage for detailed information about SimpleRISC simulators:

[http://10.1.0.78/mediawiki/index.php/CSL211 - Computer Architecture](http://10.1.0.78/mediawiki/index.php/CSL211_-_Computer_Architecture)

How to run these simulators:

1. Java based (emuSimpleRisc): goto dist directory. Read README file. Run "java -jar emuSimpleRisc.jar"
2. C based(interpreter): First compile the application using gcc: "gcc interpreter.c". Then run it using ". /a.out <file name>"

functioncall.s

```
int f(int a, int b, int c);
int g(int a, int b, int c);
int h(int a, int b, int c, int d);

int main() {
    int a = 10;
    int b = 12;
    int c = -1;
    int d;
    d = f(a, b, c);
    printf("%d", d);
}

int f(int a, int b, int c) {
    int i;
    for (i=0; i < 5; i++) {
        a = b + c;
        b = a - c;
        c = c + g(a, b, c);
    }
    return a + b + c;
}

int g(int a, int b, int c) {
    a = h(a, b, c, 0);
    b = h(a, b, c, 1);
    c = h(a, b, c, 2);
    return a + b + c;
}

int h(int a, int b, int c, int d) {
    switch(d) {
        case 0: return a + b - c;
        case 1: return a - b + c;
        case 2: return a + b + c;
    }
    return 0;
}
```

Matmul.s

You need to multiply a square $N \times N$ matrix A with another square matrix ($N \times N$) B and store the results in matrix C.

Assume the N as input argument of the program and it is stored in the register r0. To implement this, your first instruction of the program should be mov instruction initializing the r0.

The base address of A, B, and C is assumed to be 0x100, 0x200, and 0x300, respectively.

The matrix A and B would be initialized before multiplication with the following rules:

$$A[i][j] = i + j$$

$$B[i][j] = 2i + 3j$$

In the end of the program, print the matrix C using .print commands.

PerfectSquare.s

Find if the given number is a perfect square.

The program takes input arguments in register r0 and r1, and stores the result in register r0. In other words, as a first line of your program, you may initialize your arguments with mov instruction. And at the end, you may move the result to r0.