COL216 Computer Architecture
Lab Assignment 2

This assignment involves writing ARM assembly language program to generate *Happy Numbers* and storing these in a file. Information on Happy Numbers can be found here. File input/output in ARMSim# is done using 'swi' instructions, as described in section 8 of ARMSim User Guide.

A program in C for generating Happy Numbers with up to 4 digits is given below. This exploits the fact that 1 and 7 are the only two Happy Numbers with a single digit. Here the numbers are stored in binary coded decimal (BCD) form - one digit per word. In assembly language, one byte could be used for storing one digit.

```
#include <stdio.h>
int zero [4] = {0, 0, 0, 0};
int one [4] = {1, 0, 0, 0};
void copy_BCD (int * x, int * s) {
        for (int i = 0; i < 4; i++)
                x[i] = s[i];
};
void square_digit (int * dd, int d) {
        copy_BCD (dd, zero);
        dd[0] = d * d;
        while (dd[0] > 9) {
                dd[0] -= 10;
                dd[1]++;
        };
};
void add_BCD (int * x, int * y, int * z) {
        int c = 0;
        for (int i = 0; i < 4; i++) {
                x[i] = y[i] + z[i] + c;
                c = 0;
                if (x[i] > 9) {
                        x[i] -= 10; c = 1;
                };
        };
};
void sum_square (int * s, int * x) {
        int dd[4];
        copy_BCD (s, zero);
        for (int i = 0; i < 4; i++) {
                square_digit (dd, x[i]);
                add_BCD (s, s, dd);
        };
};
```

```
int check_gt_1 (int * x) {
	return ((x[1]|x[2]|x[3]) > 0);
};
int check_happy (int x) {
	return (x == 1 || x == 7);
};

int main () {
	int j = 0;
	int x[4];
	int s[4];
	int y[4];

	copy_BCD (x, one);
	for (int i = 1; i < 9999; i++) {
		copy_BCD (y, x);
		while (check_gt_1 (y)) {
			sum_square (s, y);
			copy_BCD (y, s);
		};
		if (check_happy (y[0])) {
			j++;
			printf("number[%i] = %i%i%i%i \n", j, x[3], x[2], x[1], x[0]);
		};
		add_BCD (x, x, one);
	}
	return (0);
}
```