CS1021 Tutorial #4
Flow Control and Pseudo-Code Translation

Translating Pseudo-code into ARM Assembly Language

Translate each of the following pseudo-code programs into ARM Assembly Language, making use of the CMP instruction and the conditional branch instructions.

(a) Assume x is a signed value stored in R0.

```
if (x > 1)
{
    x = x + 5;
}
```

(b) Assume x is stored in R0.

```
if (x == 0)
{
    x = 1;
}
else
{
    x = x * 2;
}
```

(c) Assume x is a signed value stored in R0 and y is stored in R1.

```
while (x < 0)
{
    y = y * x;
    x = x + 1;
}
```

(d) Assume x is an unsigned value stored in R0 and y is stored in R1.

```
while (x > 5)
{
    y = y + (2 * x);
    x = x - 5;
}
```

(e) Assume `i` is an unsigned value stored in `R0` and `y` is stored in `R1`.

```
for (i = 0; i < 10; i = i + 1)
{
    y = y + (i * i);
}
```

(f) Assume `a`, `b` and `c` are unsigned values stored in `R4`, `R5` and `R6` respectively.

```
while (a + b < 100)
{
    a = a + 1;
    b = b + c;
}
```

(g) Assume `s` is an unsigned value stored in `R3`, `t` is an unsigned value stored in `R4` and `r` is an unsigned value stored in `R5`.

```
t = 0;
while (t < 5)
{
    s = 0;
    while (s < 10)
    {
        r = (t * 10) + s;
        s = s + 1
    }
    t = t + 1;
}
```

(h) Assume `ch` is an ASCII character code stored in `R1` and `v` is stored in `R0`.

```
if (ch >= '0' && ch <= '9')
{
    v = ch - '0';
}
else if (ch >= 'A' && ch <= 'F')
{
    v = ch - 'A' + 0xA;
}
else
{
    v = 0xFFFFFFFF;
}
```

(What does this pseudo-code do?)