F28PL1 Programming Languages
Laboratory 7

Write SML functions to:

A)
i) multiply real `x` by integer `y` to give a real

```
- realIntMult 3.3 3;
> 9.9 : real
```

ii) check if string `s1` has more letters than string `s2`

```
- moreLetters "big" "small";
> false : bool
```

iii) check if character `c` represents a digit

```
- isDigit #"7";
> true : bool
```

iv) if character `c` represents a digit then return its integer equivalent; otherwise return -1

```
- digitValue #"a";
> ~1 : int
- digitValue #"7";
> 7 : int
```

v) convert a real `r` into a tuple of its value and integer equivalent

```
- conv 99.99;
> (99.99,99) : real * int
```

vi) implement the NAND function from the following table:

X	Y	X NAND Y
FALSE	FALSE	TRUE
FALSE	TRUE	TRUE
TRUE	FALSE	TRUE
TRUE	TRUE	FALSE

```
- NAND false true;
> true : bool
```

B)
i) find the integer logarithm to the base 2 of an integer `n` by repeatedly dividing by 2 and counting:

```
log2 1 = 0
log2 n = 1+log2 (n div 2)

- log2 8;
> 3 : int
```

ii) find the square root of an integer `x` by repeatedly decrementing an initial guess `s`:

```
sqrt x s = s if s*s <= x
sqrt x s = sqrt x (s-1) if s*s > x

- sqrt 10 5;
> 3 : int
```

iii) find the sum of the squares of the first `n` integers:

$n^2+(n-1)^2+(n-2)^2+...1^2$

- the sum of the squares of the first 0 integers is 0
- the sum of the squares of the first `n` integers is n squared plus the sum of the squares of the first `n-1` integers

```
- sumSq 3;
> 14 : int
```

iv) find the sum of the halves of the first `n` integers:

n/2+(n-1)/2...+1/2

```
- sumHalf 6;
> 9 : int
```

v) find the sum of applying function `f` to the first `n` integers

- the sum of applying f to the first 0 integers is 0
- the sum of applying `f` to the first `n` integers is `f` applied to `n` plus the sum of applying `f` to the first `n-1` integers

```
- fun inc x = x+1;
> val inc = fn : inet -> int
- sumF inc 3;
> 9 : int
```
i.e. `inc 3+inc 2+ inc 1+ 0`

vi) write `sumSq` using `sumF`

vii) write `sumHalf` using `sumF`