**Q7:**

A function defined using fun as recursive, like

fun fact 0 = 1

| fact n = n \* fact (n - 1);

\* This definition uses two clause:-

->The first clause uses a constant, 0, as a pattern; this pattern is only matched when the argument is zero. When evaluating a function call, the first clause that matches is used.

->The second clause will be applied for all non-zero arguments.

\* The ML definition of factorial closely corresponds to the mathematical definition like:

!(0) = 1

!(n + 1) = (n + 1)×!(n)

notice the following difference: in mathematical notation we use n + 1 as a pattern signifying that this case applies to natural numbers greater than zero; in ML there is no such pattern — we rely on the order of the patterns to get the same effect.

\* The second clause of the ML function definition is only applied when the pattern of the first clause, in this case the constant 0, has failed to match the argument.

\* Recursion is used extensively in functional programs, taking the place of constructs such as while and for loops in an imperative language. For example, to compute the sum of the squares of the first n natural numbers

SS(n) = Σn

n1

i=0

i2