2 Foundations of Computer Science (MOM)

This question has been translated from Standard ML to OCaml

(*a*) Write brief notes on exceptions in OCaml and on the functions and control structures available for programming with them. [6 marks]

Parts (*b*) and (*c*) make use of the following OCaml exception:

```
exception Olive
```

(*b*) Code in OCaml a function called `cannot` which takes two arguments, a function `f` and a value `x`. Define the `cannot` function in such a way that it returns `true` if and only if evaluation of `f(x)` causes exception `Olive`. For all other inputs, it should return `false`. [Hint: evaluation of `f(x)` may cause exceptions other than `Olive`.] [4 marks]

(*c*) Consider the following OCaml type and functions `bun` and `cheese`.

```
type 'a tree = Leaf of 'a
             | Branch of 'a tree * 'a tree

let rec bun x = function
  | Leaf y          -> if x = y then raise Olive else Leaf y
  | Branch (t1, t2) -> Branch (bun x t1, bun x t2)

let cheese x t = if cannot (bun x) t then Leaf x else bun x t
```

(*i*) Write down the type of `cheese`. [3 marks]

(*ii*) Write a function that is equivalent to `cheese` but makes no use of exceptions. Briefly explain why your function is equivalent to `cheese`. [7 marks]