1 Foundations of Computer Science (LCP)

This question has been translated from Standard ML to OCaml

(*a*) Write brief notes on polymorphism in OCaml, using lists and standard list functions such as `@` (append) and `List.map`. [4 marks]

(*b*) Explain the meaning of the following declaration and describe the corresponding data structure, including the role of polymorphism.

```
type 'a se = Void | Unit of 'a | Join of 'a se * 'a se
```

[4 marks]

(*c*) Show that OCaml lists can be represented using this variant type by writing the functions `encode_list` of type `'a list -> 'a se` and `decode_list` of type `'a se -> 'a list`, such that `decode_list (encode_list xs) = xs` for every list `xs`. [3 marks]

(*d*) Consider the following function declaration:

```
let rec cute p = function
| Void -> false
| Unit x -> p x
| Join (u, v) ->
    cute p u || cute p v
```

What does this function do, and what is its type? [4 marks]

(*e*) Consider the following expression:

```
fun p -> cute (cute p)
```

What does it mean, and what is its type? Justify your answer carefully.
[5 marks]