Quiz 2 - OCaml

Q1 OCaml Typing
2 Points

Write an OCaml expression of type int := 42;

Q2 OCaml Typing
2 Points

Write an OCaml expression of type int := 42;

Q3 OCaml Typing2
4 Points

Q3.1 OCaml Typing2
2 Points

Write the type of the following expression:

\[
\text{let f p x y = which s t w}
\]

Q3.2 OCaml Typing2
2 Points

Write the type of the following expression:

\[
\text{let f p x y = what (p x) y}
\]

Q4 OCaml Coding
6 Points

Write a function `f ` which returns a new list of multiplications between sum of elements of list and each member of list.

You can assume that `f` will not be empty.

You can use exactly the following definitions of `map` and `fold` (you cannot use the `List` module):

\[
\text{let m o p} \text{ x} = \text{ fold } (\text{ o } \text{ p } \text{ x}) \text{ x}
\]

Example:

\[
\text{map} \text{ add } \text{ [0;1;2;3]} = \text{ [2;3;4;5]}
\]

Notes: You are not allowed to use the `List` module.

```
let m o p x = fold (o p x) x
```

Blank 1:

```
let m o p x = fold (o p x) x
```

Blank 2:

```
let m o p x = fold (o p x) x
```

Example:

\[
\text{map} \text{ m add } \text{ [0;1;2;3]} = \text{ [2;3;4;5]}
\]