A single-player game requires you to guess the letters of a target word within a limited number of tries. Each try the player guesses a single letter $c$ and integer position $i$ (starting from 0). The game responds: Green if $c$ appears in the position $i$ in the target, Amber if $c$ is present elsewhere within the target, or Black if $c$ is not present in the target. For simplicity, repeated letters are not allowed in the target.

```plaintext
type word = char list
type guess = char * int
type guesses = guess list
val prune_guesses : guesses -> guesses
```

The word type is used to list the correct letters in order, guess is a single guess (the letter and its position within the target word) and guesses is used to list the guesses in order with the most recent first. You may assume that a prune_guesses function is available that removes all but the most recent guess for each character.

(a) Define two functions mapi and lookfor with the following types:

```plaintext
val mapi : (int -> 'a -> 'b) -> 'a list -> 'b list
val lookfor : 'a -> ('b * 'a) list -> 'b option
```

The call mapi f l maps over every element of l using a function f that accepts the position of the current list element along with the element. For example, mapi f ["a";"b";"c"] = [ f 0 "a"; f 1 "b"; f 2 "c" ].

The call lookfor y l searches l for a pair where the second component is equal to y, and returns the first component in the pair if found. [4 marks]

(b) Using the earlier definitions, or otherwise, write a function

```plaintext
val respond : word -> guesses -> responses
```

that takes a target word and a list of guesses and returns feedback about the game progress to the player. You should define a responses type by referring to the game rules. [8 marks]

(c) (i) Define a function create_game: word -> (guess -> responses) that returns a function $g$ which can be used to imperatively play an independent game: any such $g$ can be called for at most six tries after which it raises an Out_of_turns exception. [6 marks]

(ii) Illustrate with a brief example how you would use such a function $g$ to make some tries, showing the three possible response types. [2 marks]