Fundamentals of Programming
Assignment 1, due Friday Feb 11th at 11:59 pm (cutoff enforced by IVLE)

This assignment is out of 10 marks.

- (6 marks) Please implement union-find as discussed in class. I have uploaded these files to IVLE:
    1. `unionfind.mli`, giving the interface I expect you to follow. Please implement a file `unionfind.ml` that implements (at least) the functions in that interface.
    2. Note that you need to define several different versions of union and find with various optimizations turned on and off for comparison purposes.
    3. `test_unionfind.ml`, suitable for testing your code. On Windows, compile with:
       ```
       ocamlfind ocamlc -package unix unionfind.mli
           unionfind.ml test_unionfind.ml -linkpkg -o test_uf
       ```
       *Can someone post the correct incantation for Mac on the IVLE Forum?*
    4. `test_unionfind_top.ml`, suitable for testing your code in interactive mode (e.g., in ocaml-top).
    5. `tinyUF.txt`, `mediumUF.txt`, and `largeUF.txt`, three test files (by Segwick & Wayne)
- (1 mark) Please document the runtimes you have for the various options, being sure to note any surprises and explain them if you can. Put this information in a `(* comment *)` in the bottom of your file.
- (3 marks in total) Create a new version of the implementation and interface that satisfies two additional properties:
    1. (2 marks) One of the disadvantages of the naïve array-based format I discussed is that you need to specify in advance how many Singletons you want (in the call to `make` and `make_full`). Please remove this restriction so that `make` has type `unit -> uf_t` instead of `int -> uf_t`. Your new version should enjoy the same amortized time bounds as the original. **Hint:** recall the expandable arrays from last semester.
    2. (1 mark) Add the function `components : uf_t -> (int list) list` which gives you a list of all of the components/sets in the structure. The first element of each list should be the principal element of the component. Note that the lists may not be unique.
    3. Put your solution in `unionfind_unbound.ml`, and redo the interface into `unionfind_unbound.mli`.

Your files should be put into a zipfile `yourname_unionfind_tests.zip`. *Please don't put the testing file `largeUF.txt` in your zip file.* It is really big and I already have a copy.