課題 1

以下では，必要ならば，補助関数を定義して利用してもよい.

1. 以下の関数 fib は多重再帰を含むため大変性能が悪い.
   $O(n)$ の手間でフィボナッチ数を計算する関数 fib を定義する.
   - **ヒント:** 3引数の補助関数を定義してみよう.

```
let rec fib n =
  if n<=1 then n
  else fib (n-1) + fib (n-2);;
```

【発展課題】$O(\log n)$ の手間でフィボナッチ数を計算する関数は作れるだろうか？

課題 1

2. リスト（個々のリストは互いに異なる要素からなるものとする）を２本もらって，どちらのリストにも現れる要素を集めたリストを返す関数 intersection を書く．
   - `intersection [1;3;6;10;14] [10;3]` の実行結果が `[10;3]` か `[3;10]` となればよい

3. リスト（個々のリストは互いに異なる要素からなるものとする）を２本もらって，最初のリストには現れるが２つめのリストには現れない要素のリストを返す関数 difference を書く．

課題 1

4. リストを１本もらって，重複する要素を取り除いたリストを返す関数 `unduplicate` を書く．
   - たとえば
     `unduplicate [1;4;6;1;6;6]` の実行結果が `[4;1;6]` または `[1;4;6]` などになればよい．

課題 1

5. 整数 n をもらって以下の数列の和を求める関数 exp を書く.

$$\exp(n) = 1 + \frac{1}{1!} + \frac{1}{2!} + \cdots + \frac{1}{n!}$$

- int と float の混合演算はできないので自分で変換
- 掛算の回数が $O(n^2)$ にならないように工夫しよう

課題 1（Cプログラミング，任意提出）

6. 以下のように，コマンドラインから与えた０個以上の１桁の整数の和を印刷するＣプログラムを書く．

```
$ ./a.out 3 1 4 1 5 9
23
```

ただし ...

```
#include <stdio.h>
int main(int argc, char *argv[]) {
```

- 変数への破壊的代入 (=, +=, ++ etc.)
- for, while, do_while, goto, longjmp
- (main 以外の) 関数定義

使用禁止

```
}
```

課題 1

以上の問について，

(a) 書いたプログラム（できたところまで）

(b) 実行結果

- OCaml 処理系の出力の切り貼り（画面キャプチャではなく出力テキストの切り貼り）を，個々の問題に対して複数個添付する

(c) OCamlプログラミングの感想

を CourseN@vi から提出してください.

なお，問１の【発展課題】と問６は任意提出で，良い答案は加点材料にします.